To all our customers

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

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

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

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

Cautions

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

Hitachi 16-Bit Single-Chip Microcomputer

H8S/2556 Series,
H8S/2552 Series
H8S/2506 Series

H8S/2556
HD64F2556
H8S/2552
HD64F2552
H8S/2506
HD64F2506

Hardware Manual

ADE-602-311

Rev 1.0
03/12/03
Hitachi, Ltd.

Cautions

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

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 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.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

     The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

Preface

This LSI is a high-performance microcomputer made up of the H8S/2000 CPU with an internal 32-bit configuration as its core, and the peripheral functions required to configure a system.

A single-power flash memory (F-ZTAT™)*[1] version is available for this LSI's ROM. The F-ZTAT version provides flexibility as it can be reprogrammed in no time to cope with all situations from the early stages of mass production to full-scale mass production. This is particularly applicable to application devices with specifications that will most probably change. The on-chip peripheral functions for each of the series are shown below.

List of on-chip peripheral functions:

Series Name	**H8S/2556 Series**	**H8S/2552 Series**	**H8S/2506 Series**
Product Name	**H8S/2556**	**H8S/2552**	**H8S/2506**
Bus controller	O (16 bits)	O (16 bits)	O (16 bits)
Data transfer controller (DTC)	O	O	O
PC break controller (PBC)	O	O	O
16-bit timer pulse unit (TPU)	X 6	X 6	X 6
8-bit timer (TMR)	X 4	X 4	X 4
Watch dog timer (WDT)	X 2	X 2	X 2
Serial communication interface (SCI)	X 5	X 5	X 5
I^2C bus interface 2 (IIC2)	X 2	X 2	X 2
IEBus™*[2] controller (IEB)	—	X 1	—
Hitachi controller area network (HCAN)	X 1	—	—
D/A converter	X 2	X 2	X 2
A/D converter	X 16	X 16	X 16

Notes: 1. F-ZTAT™ is a trademark of Hitachi, Ltd.
2. IEBus is a trademark of NEC Electronics Corporation.

Target Users: This manual was written for users who will be using the H8S/2556 Series, H8S/2552 Series, and H8S/2506 Series in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2556 Series, H8S/2552 Series, and H8S/2506 Series, H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8S/2600 Series, H8S/2000 Series Programming Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 23, List of Registers.

  Examples:
  - Register name: The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
  - Bit order: The MSB is on the left and the LSB is on the right.
  - Number notation: Binary is B′xxxx, hexadecimal is H′xxxx, decimal is xxxx.
  - Signal notation: An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require.
http://www.hitachisemiconductor.com/

H8S/2556 Series, H8S/2552 Series, and H8S/2506 Series manuals:

Manual Title	ADE No.
H8S/2556 Series, H8S/2552 Series, H8S/2506 Series, Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	ADE-602-083

User's manuals for development tools:

Manual Title	ADE No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimized Linkage Editor User's Manual	ADE-702-247
H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-282
H8S, H8/300 Series Hitachi Embedded Workshop, Hitachi Debugging Interface Tutorial	ADE-702-231
Hitachi Embedded Workshop User's Manual	ADE-702-201

Contents

Figures

Section 9 I/O Ports

Section 10 16-Bit Timer Pulse Unit (TPU)

Section 11 8-Bit Timers (TMR)

Section 12 Watchdog Timer (WDT)

Section 13 Serial Communication Interface (SCI)

Section 14 I^2C Bus Interface 2 (IIC2)

Section 15 A/D Converter

Section 16 D/A Converter

Section 17 IEBus™ Controller (IEB) [H8S/2552 Series]

Section 18 Hitachi Controller Area Network (HCAN) [H8S/2556 Series]

Section 20 Flash Memory (0.18-µm F-ZTAT Version)

Section 21 Clock Pulse Generator

Section 22 Power-Down Modes

Section 24 Electrical Characteristics

Appendix

Tables

Section 1 Overview

1.1 Features

- High-speed H8S/2000 central processing unit with an internal 16-bit architecture
  - Upward-compatible with H8/300 and H8/300H CPUs on an object level
  - Sixteen 16-bit general registers
  - 65 basic instructions
- Various peripheral functions
  - PC break controller
  - Data transfer controller (DTC)
  - 16-bit timer-pulse unit (TPU)
  - 8-bit timer (TMR)
  - Watchdog timer (WDT)
  - Serial communication interface (SCI)
  - I^2C bus interface 2 (IIC2)
  - 10-bit A/D converter
  - 8-bit D/A converter
  - IEBus™ controller (IEB) (H8S/2552)
  - Hitachi controller area network (HCAN) (H8S/2556)
- On-chip memory

ROM	Model	ROM	RAM	Remarks
Flash memory version	HD64F2556	512 kbytes	32 kbytes	
	HD64F2552	512 kbytes	32 kbytes	
	HD64F2506	512 kbytes	32 kbytes	

- General I/O ports

  Support two types of the port with different power supply sources
  - H8S/2556 Series

    I/O pins: 102

    HCAN pins: 2 (one input and one output)

    Input pins: 16
  - H8S/2552 Series and H8S/2506 Series

    I/O pins: 104

    Input port: 16
- Supports various power-down modes

- Compact package

Package	(Code)	Body Size	Pin Pitch
QFP-144	FP-144J	20.0 × 20.0 mm	0.5 mm

1.2 Internal Block Diagram

Figures 1.1 to 1.3 show the internal block diagrams.

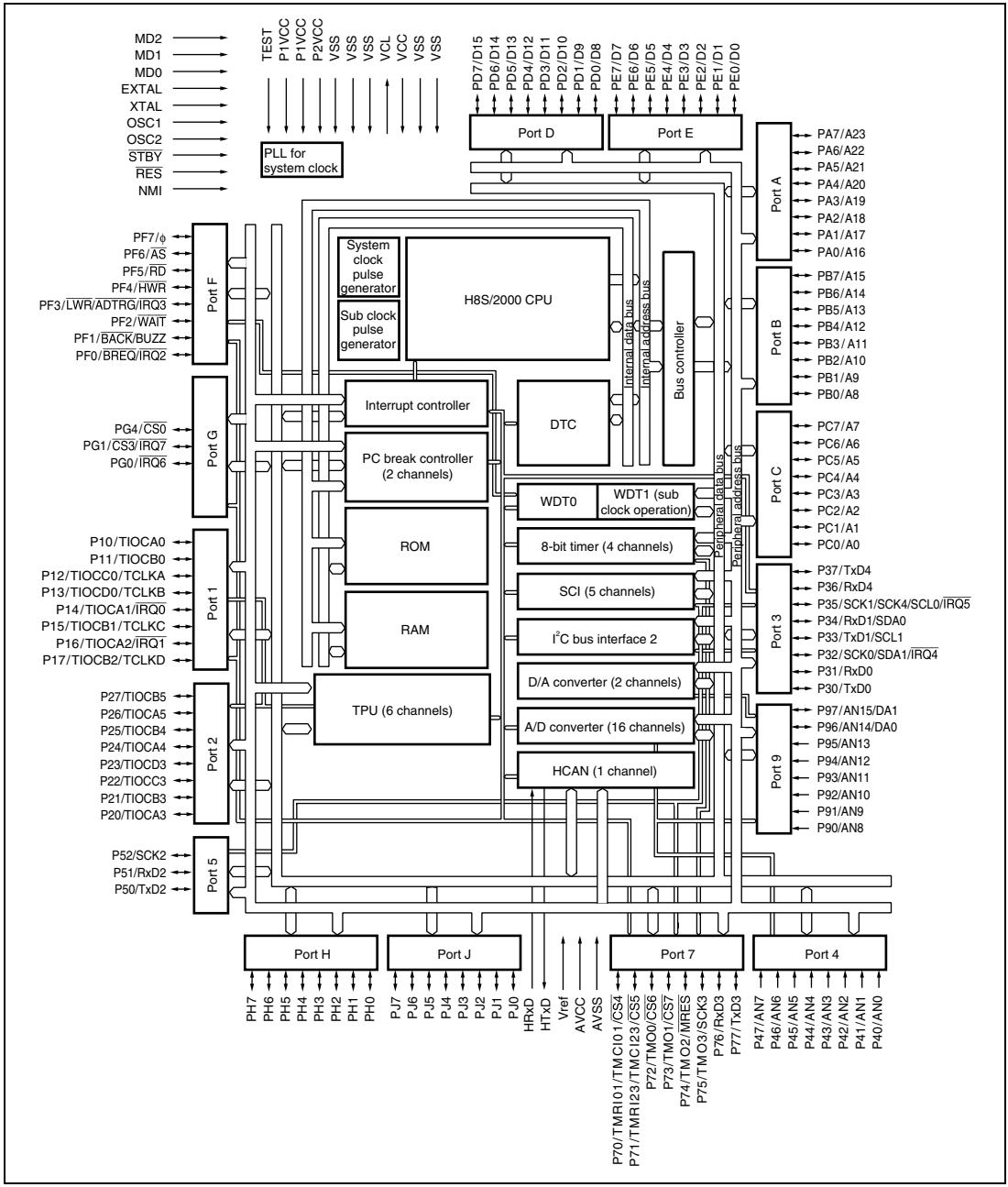


Figure 1.1 Internal Block Diagram of H8S/2556 Series

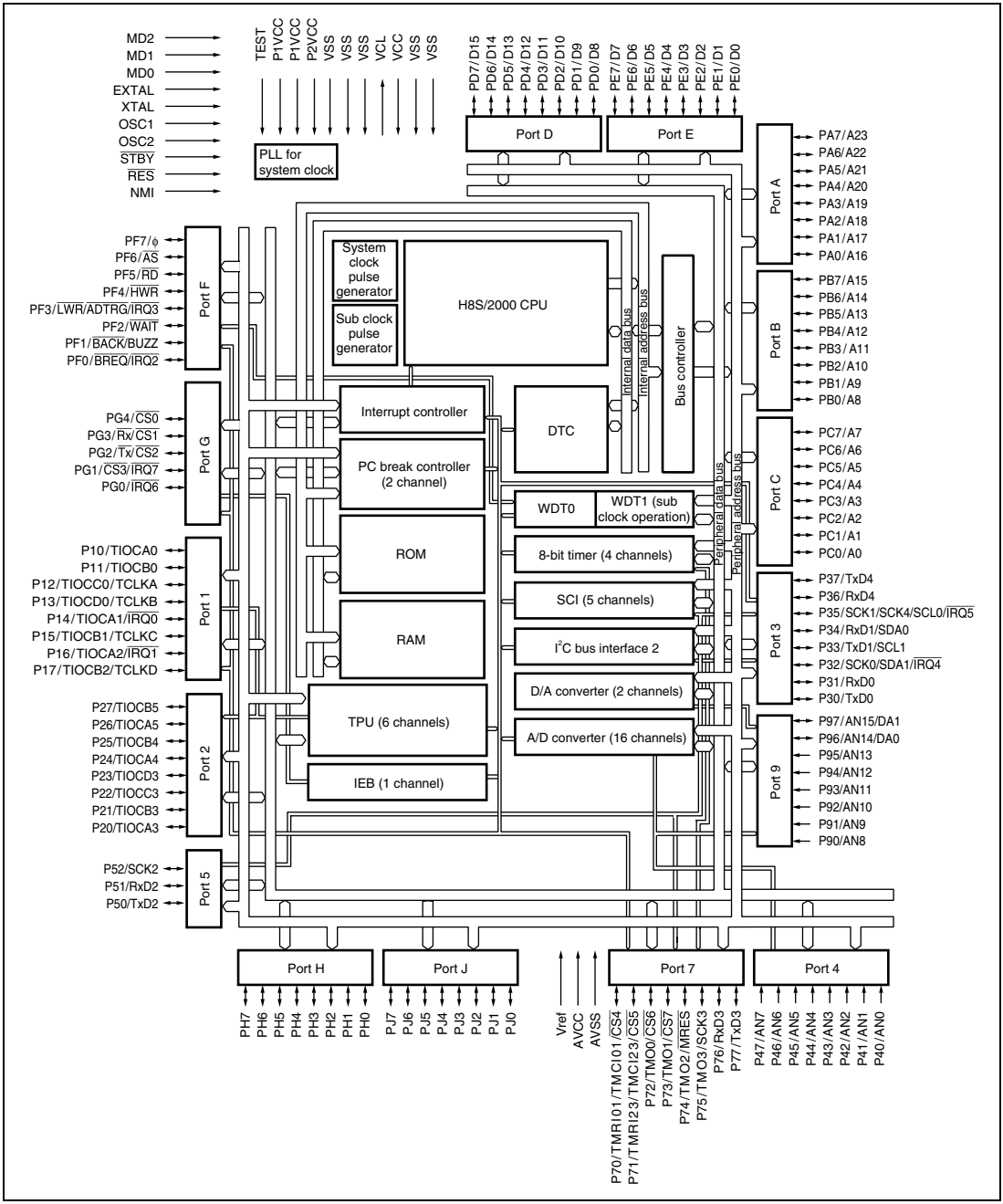


Figure 1.2 Internal Block Diagram of H8S/2552 Series

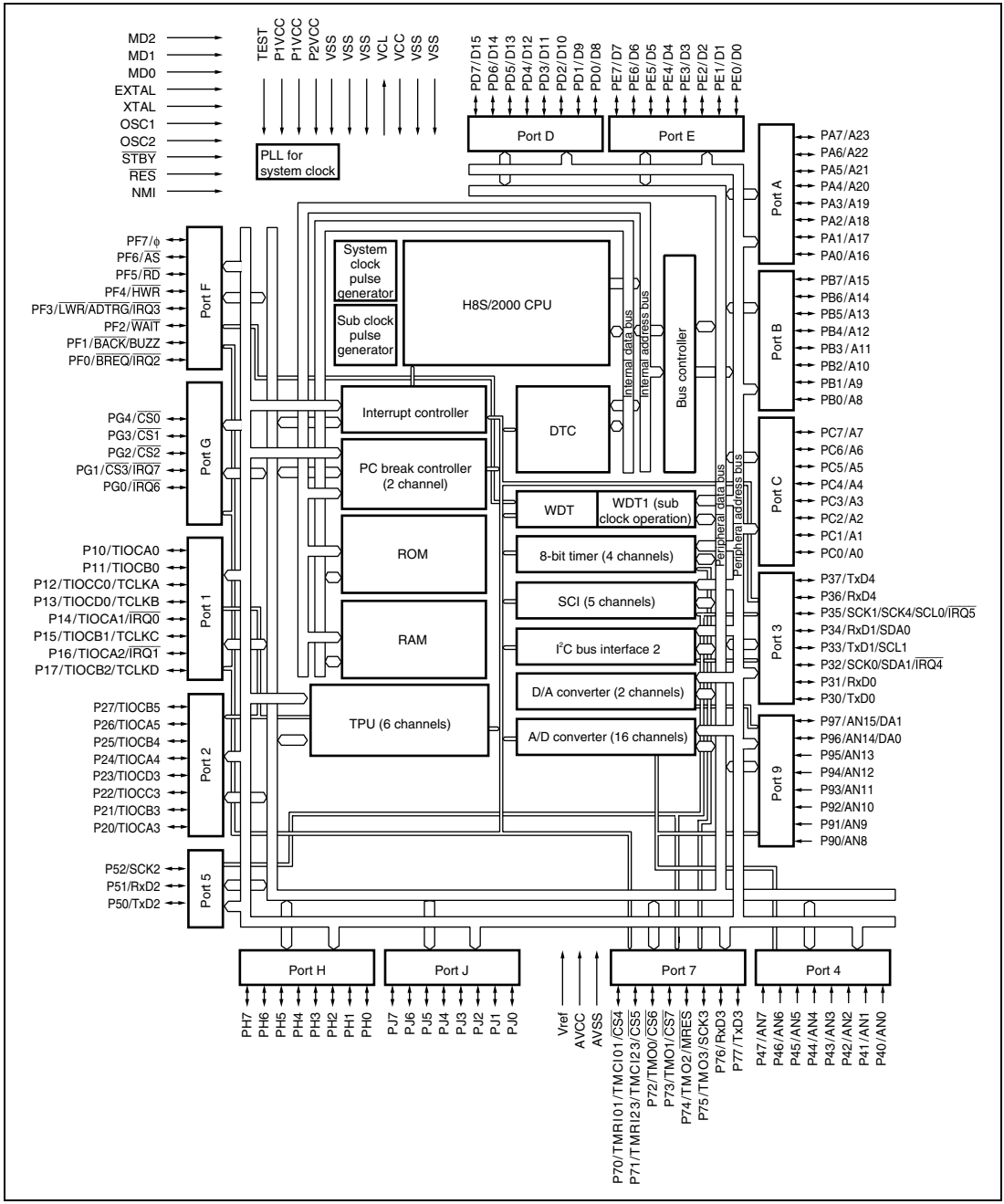


Figure 1.3 Internal Block Diagram of H8S/2506 Series

1.3 Pin Arrangement

1.3.1 Pin Arrangement

Figures 1.4 to 1.6 show the pin arrangement.

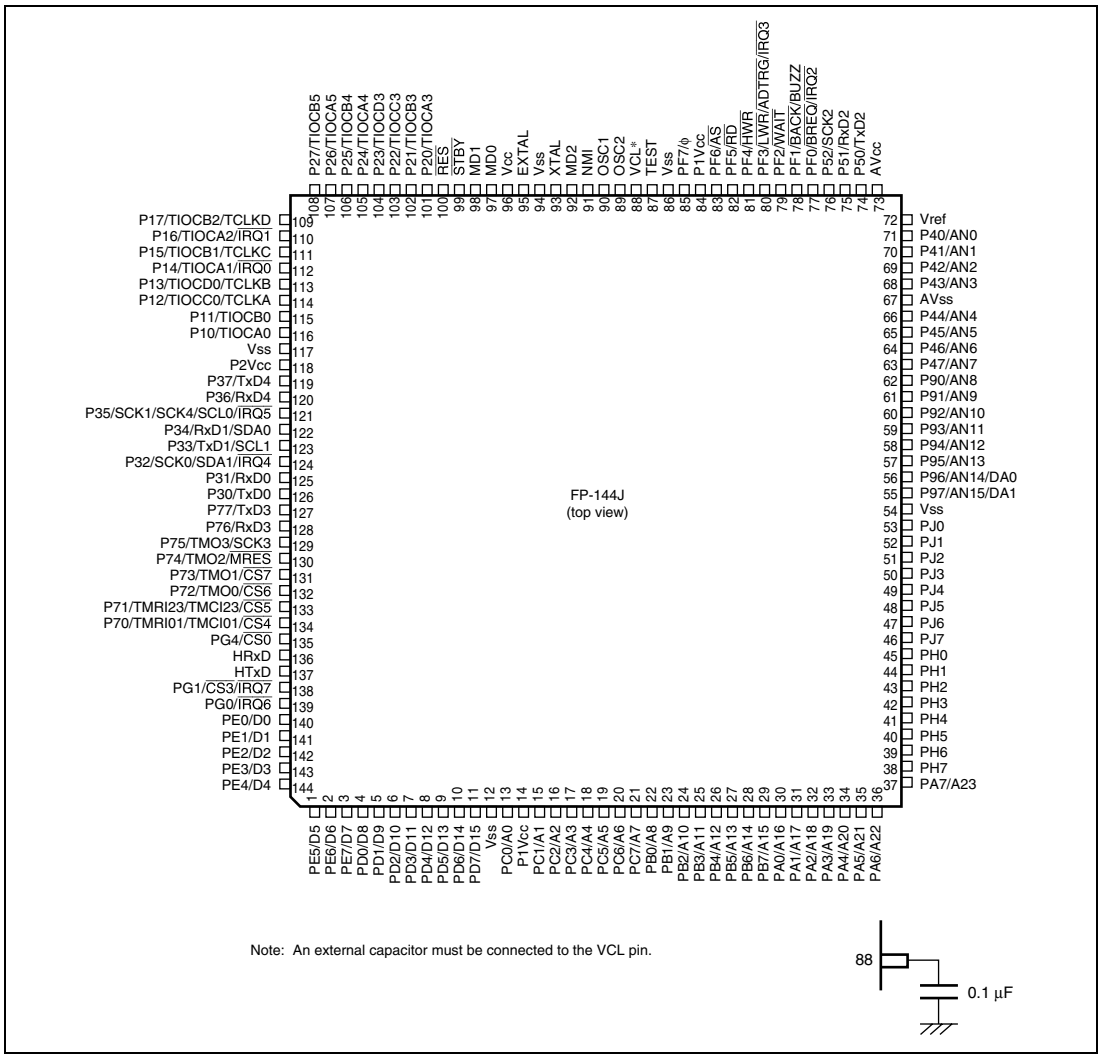


Figure 1.4 Pin Arrangement of H8S/2556 Series (FP-144J)

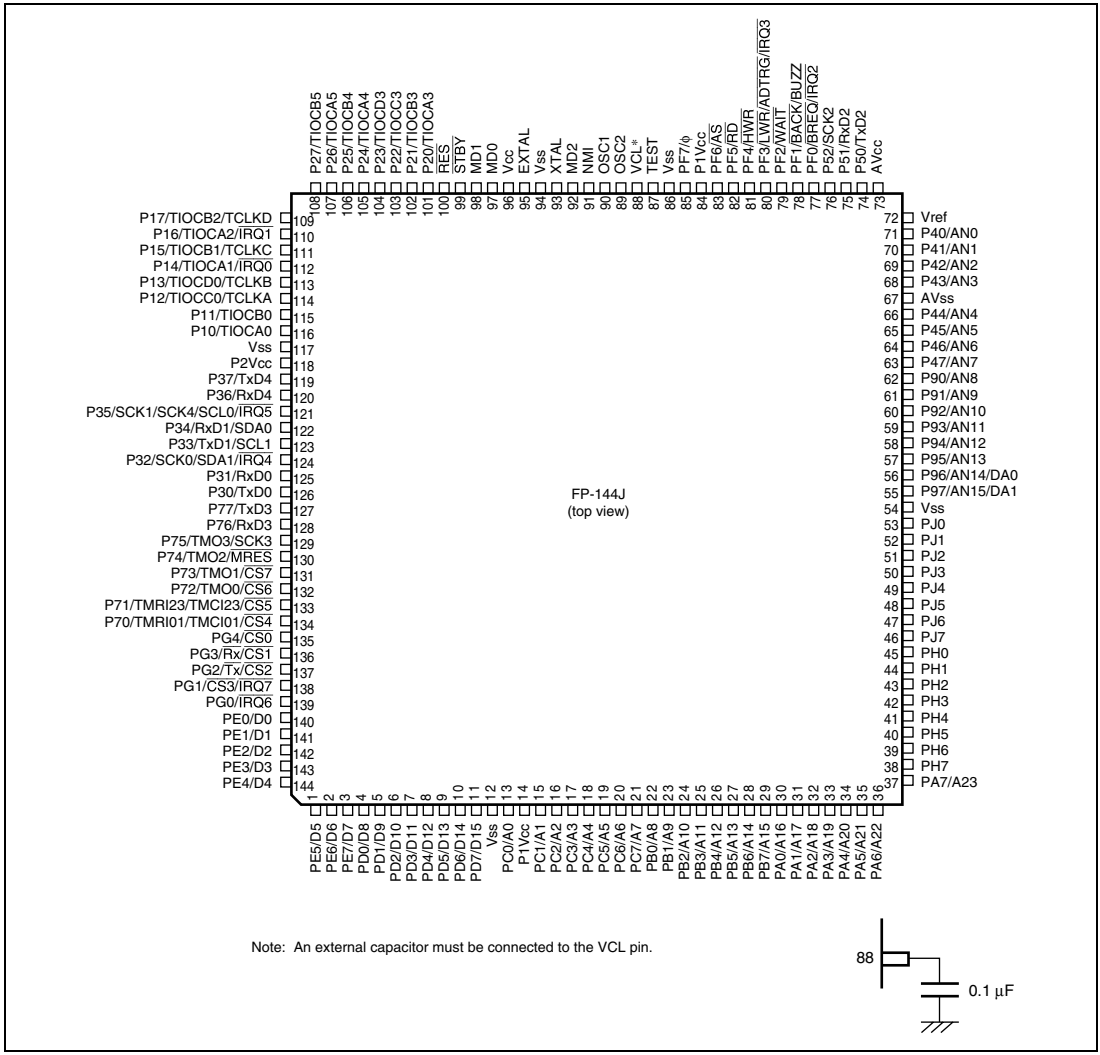


Figure 1.5 Pin Arrangement of H8S/2552 Series (FP-144J)

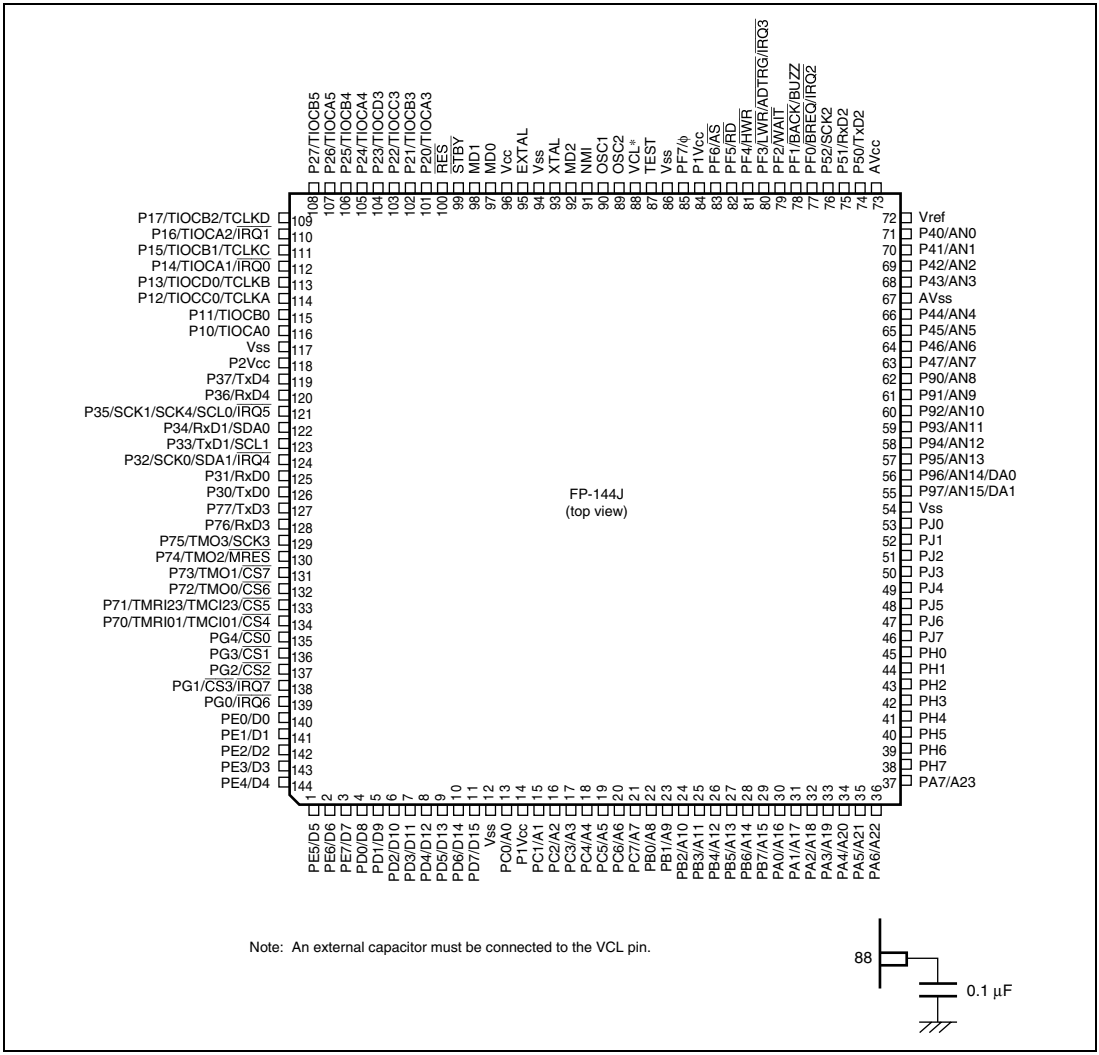


Figure 1.6 Pin Arrangement of H8S/2506 Series (FP-144J)

1.3.2 Pin Arrangement in Each mode

Pin arrangement in each mode are shown below.

Table 1.1 Pin Arrangement in Operating Mode

	Pin Name			
Pin No.	Mode 6	Mode 7	Flash Memory Programmer Mode	Power Supply Source
1	PE5/D5	PE5	$\overline{OE}$	P1Vcc
2	PE6/D6	PE6	$\overline{WE}$	P1Vcc
3	PE7/D7	PE7	$\overline{CE}$	P1Vcc
4	D8	PD0	D0	P1Vcc
5	D9	PD1	D1	P1Vcc
6	D10	PD2	D2	P1Vcc
7	D11	PD3	D3	P1Vcc
8	D12	PD4	D4	P1Vcc
9	D13	PD5	D5	P1Vcc
10	D14	PD6	D6	P1Vcc
11	D15	PD7	D7	P1Vcc
12	Vss	Vss	Vss	Vss
13	PC0/A0	PC0	A0	P1Vcc
14	P1Vcc	P1Vcc	Vcc	P1Vcc
15	PC1/A1	PC1	A1	P1Vcc
16	PC2/A2	PC2	A2	P1Vcc
17	PC3/A3	PC3	A3	P1Vcc
18	PC4/A4	PC4	A4	P1Vcc
19	PC5/A5	PC5	A5	P1Vcc
20	PC6/A6	PC6	A6	P1Vcc
21	PC7/A7	PC7	A7	P1Vcc
22	PB0/A8	PB0	A8	P1Vcc
23	PB1/A9	PB1	A9	P1Vcc
24	PB2/A10	PB2	A10	P1Vcc
25	PB3/A11	PB3	A11	P1Vcc
26	PB4/A12	PB4	A12	P1Vcc
27	PB5/A13	PB5	A13	P1Vcc

Pin No.	Pin Name		Flash Memory Programmer Mode	Power Supply Source
	Mode 6	Mode 7		
28	PB6/A14	PB6	A14	P1Vcc
29	PB7/A15	PB7	A15	P1Vcc
30	PA0/A16	PA0	A16	P1Vcc
31	PA1/A17	PA1	A17	P1Vcc
32	PA2/A18	PA2	A18	P1Vcc
33	PA3/A19	PA3	NC	P1Vcc
34	PA4/A20	PA4	NC	P1Vcc
35	PA5/A21	PA5	NC	P1Vcc
36	PA6/A22	PA6	NC	P1Vcc
37	PA7/A23	PA7	NC	P1Vcc
38	PH7	PH7	NC	P1Vcc
39	PH6	PH6	NC	P1Vcc
40	PH5	PH5	NC	P1Vcc
41	PH4	PH4	NC	P1Vcc
42	PH3	PH3	NC	P1Vcc
43	PH2	PH2	NC	P1Vcc
44	PH1	PH1	NC	P1Vcc
45	PH0	PH0	NC	P1Vcc
46	PJ7	PJ7	NC	P1Vcc
47	PJ6	PJ6	NC	P1Vcc
48	PJ5	PJ5	NC	P1Vcc
49	PJ4	PJ4	NC	P1Vcc
50	PJ3	PJ3	NC	P1Vcc
51	PJ2	PJ2	NC	P1Vcc
52	PJ1	PJ1	NC	P1Vcc
53	PJ0	PJ0	NC	P1Vcc
54	Vss	Vss	Vss	Vss
55	P97/AN15/DA1	P97/AN15/DA1	NC	AVcc
56	P96/AN14/DA0	P96/AN14/DA0	NC	AVcc
57	P95/AN13	P95/AN13	NC	AVcc
58	P94/AN12	P94/AN12	NC	AVcc

Pin No.	Pin Name Mode 6	Mode 7	Flash Memory Programmer Mode	Power Supply Source
59	P93/AN11	P93/AN11	NC	AVcc
60	P92/AN10	P92/AN10	NC	AVcc
61	P91/AN9	P91/AN9	NC	AVcc
62	P90/AN8	P90/AN8	NC	AVcc
63	P47/AN7	P47/AN7	NC	AVcc
64	P46/AN6	P46/AN6	NC	AVcc
65	P45/AN5	P45/AN5	NC	AVcc
66	P44/AN4	P44/AN4	NC	AVcc
67	AVss	AVss	Vss	AVss
68	P43/AN3	P43/AN3	NC	AVcc
69	P42/AN2	P42/AN2	NC	AVcc
70	P41/AN1	P41/AN1	NC	AVcc
71	P40/AN0	P40/AN0	NC	AVcc
72	Vref	Vref	Vcc	Vref
73	AVcc	AVcc	Vcc	AVcc
74	P50/TxD2	P50/TxD2	NC	P1Vcc
75	P51/RxD2	P51/RxD2	NC	P1Vcc
76	P52/SCK2	P52/SCK2	NC	P1Vcc
77	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	Vcc	P1Vcc
78	PF1/$\overline{\text{BACK}}$/BUZZ	PF1/BUZZ	NC	P1Vcc
79	PF2/$\overline{\text{WAIT}}$	PF2	NC	P1Vcc
80	PF3/$\overline{\text{LWR}}$/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	Vcc	P1Vcc
81	$\overline{\text{HWR}}$	PF4	NC	P1Vcc
82	$\overline{\text{RD}}$	PF5	NC	P1Vcc
83	$\overline{\text{AS}}$	PF6	NC	P1Vcc
84	P1Vcc	P1Vcc	Vcc	P1Vcc
85	PF7/ϕ	PF7/ϕ	NC	P1Vcc
86	Vss	Vss	Vss	Vss
87	TEST	TEST	Vss	Vcc
88	VCL	VCL	VCL	VCL
89	OSC2	OSC2	NC	—

Pin No.	Pin Name		Flash Memory Programmer Mode	Power Supply Source
	Mode 6	Mode 7		
90	OSC1	OSC1	Vss	—
91	NMI	NMI	Vcc	Vcc
92	MD2	MD2	Vss	Vcc
93	XTAL	XTAL	XTAL	—
94	Vss	Vss	Vss	Vss
95	EXTAL	EXTAL	EXTAL	Vcc
96	Vcc	Vcc	Vcc	Vcc
97	MD0	MD0	Vss	Vcc
98	MD1	MD1	Vss	Vcc
99	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	Vcc	Vcc
100	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	Vcc
101	P20/TIOCA3	P20/TIOCA3	NC	P2Vcc
102	P21/TIOCB3	P21/TIOCB3	NC	P2Vcc
103	P22/TIOCC3	P22/TIOCC3	NC	P2Vcc
104	P23/TIOCD3	P23/TIOCD3	NC	P2Vcc
105	P24/TIOCA4	P24/TIOCA4	NC	P2Vcc
106	P25/TIOCB4	P25/TIOCB4	NC	P2Vcc
107	P26/TIOCA5	P26/TIOCA5	NC	P2Vcc
108	P27/TIOCB5	P27/TIOCB5	NC	P2Vcc
109	P17/TIOCB2/TCLKD	P17/TIOCB2/TCLKD	NC	P2Vcc
110	P16/TIOCA2/$\overline{\text{IRQ1}}$	P16/TIOCA2/$\overline{\text{IRQ1}}$	Vss	P2Vcc
111	P15/TIOCB1/TCLKC	P15/TIOCB1/TCLKC	NC	P2Vcc
112	P14/TIOCA1/$\overline{\text{IRQ0}}$	P14/TIOCA1/$\overline{\text{IRQ0}}$	Vss	P2Vcc
113	P13/TIOCD0/TCLKB	P13/TIOCD0/TCLKB	NC	P2Vcc
114	P12/TIOCC0/TCLKA	P12/TIOCC0/TCLKA	NC	P2Vcc
115	P11/TIOCB0	P11/TIOCB0	NC	P2Vcc
116	P10/TIOCA0	P10/TIOCA0	NC	P2Vcc
117	Vss	Vss	Vss	Vss
118	P2Vcc	P2Vcc	Vcc	P2Vcc
119	P37/TxD4	P37/TxD4	NC	P2Vcc
120	P36/RxD4	P36/RxD4	NC	P2Vcc

Pin No.	Pin Name Mode 6	Mode 7	Flash Memory Programmer Mode	Power Supply Source
121	P35/SCK1/SCK4/SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCK4/SCL0/$\overline{\text{IRQ5}}$	NC	P2Vcc
122	P34/RxD1/SDA0	P34/RxD1/SDA0	NC	P2Vcc
123	P33/TxD1/SCL1	P33/TxD1/SCL1	NC	P2Vcc
124	P32/SCK0/SDA1/$\overline{\text{IRQ4}}$	P32/SCK0/SDA1/$\overline{\text{IRQ4}}$	NC	P2Vcc
125	P31/RxD0	P31/RxD0	NC	P2Vcc
126	P30/TxD0	P30/TxD0	NC	P2Vcc
127	P77/TxD3	P77/TxD3	NC	P2Vcc
128	P76/RxD3	P76/RxD3	NC	P2Vcc
129	P75/TMO3/SCK3	P75/TMO3/SCK3	NC	P2Vcc
130	P74/TMO2/$\overline{\text{MRES}}$	P74/TMO2/$\overline{\text{MRES}}$	NC	P2Vcc
131	P73/TMO1/$\overline{\text{CS7}}$	P73/TMO1	NC	P2Vcc
132	P72/TMO0/$\overline{\text{CS6}}$	P72/TMO0	NC	P2Vcc
133	P71/TMRI23/TMCI23/$\overline{\text{CS5}}$	P71/TMRI23/TMCI23	NC	P2Vcc
134	P70/TMRI01/TMCI01/$\overline{\text{CS4}}$	P70/TMRI01/TMCI01	NC	P2Vcc
135	PG4/$\overline{\text{CS0}}$	PG4	NC	P1Vcc
136	HRxD*[1]	HRxD*[1]	NC	P1Vcc
	PG3/$\overline{\text{Rx}}$/$\overline{\text{CS1}}$*[2]	PG3/$\overline{\text{Rx}}$*[2]		
	PG3/$\overline{\text{CS1}}$*[3]	PG3*[3]		
137	HTxD*[1]	HTxD*[1]	NC	P1Vcc
	PG2/$\overline{\text{Tx}}$/$\overline{\text{CS2}}$*[2]	PG2/$\overline{\text{Tx}}$*[2]		
	PG2/$\overline{\text{CS2}}$*[3]	PG2*[3]		
138	PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{IRQ7}}$	NC	P1Vcc
139	PG0/$\overline{\text{IRQ6}}$	PG0/$\overline{\text{IRQ6}}$	NC	P1Vcc
140	PE0/D0	PE0	NC	P1Vcc
141	PE1/D1	PE1	NC	P1Vcc
142	PE2/D2	PE2	NC	P1Vcc
143	PE3/D3	PE3	Vcc	P1Vcc
144	PE4/D4	PE4	Vss	P1Vcc

Notes: 1. Symbol name for the H8S/2556 Series
2. Symbol name for the H8S/2552 Series
3. Symbol name for the H8S/2506 Series

1.3.3 Pin Functions

Table 1.2 lists the pins functions in each mode.

Table 1.2 Pin Functions

Type	Symbol	Pin No.	I/O	Function
Power supply	Vcc	96	I	Power supply pin. Connect this pin to the system power supply.
	P1Vcc	14, 84	I	Power supply pin for ports indicated that its power is supplied by P1Vcc (see table 1.1).
	P2Vcc	118	I	Power supply pin for ports indicated that its power is supplied by P2Vcc (see table 1.1).
	VCL	88	O	Must not be directly connected to a power supply. A capacitor of 0.1 μF must be connected between this pin and Vss. (Place close to the pin.)
	VSS	12, 54, 86, 94, 117	I	Ground pins. Connect this pin to the system power supply (0V).
Clock	XTAL	93	I	For connection to a crystal resonator. For examples of connecting crystal resonator and external clock input, see section 21, Clock Pulse Generator.
	EXTAL	95	I	For connection to a crystal resonator or a ceramic resonator. This pin can be also used for external clock input. For examples of connecting crystal resonator and external clock input, see section 21, Clock Pulse Generator.
	OSC1	90	I	Connects to a 32.768 kHz crystal resonator. For examples of connecting crystal resonator, see section 21, Clock Pulse Generator.
	OSC2	89	I	Connects to a 32.768 kHz crystal resonator. For examples of connecting crystal resonator, see section 21, Clock Pulse Generator.
	ϕ	85	O	Supplies clock pulses to external devices.
Operating mode control	MD2 MD1 MD0	92 98 97	I	Sets the operating mode. Inputs at these pins should not be changed during operation. Be sure to fix the levels of the mode pins (MD2 to MD0) by pull-down or pull-up, except for mode changing.
System control	$\overline{\text{RES}}$	100	I	Reset input pin. When this pin is low, this LSI enters the power-on reset state.
	$\overline{\text{MRES}}$	130	I	Reset input pin. When this pin is low, this LSI enters the manual reset state.
	$\overline{\text{STBY}}$	99	I	When this pin is low, a transition is made to hardware standby mode.

Type	Symbol	Pin No.	I/O	Function
System control	$\overline{\text{BREQ}}$	77	I	Indicates that an external bus master is requesting bus mastership.
	$\overline{\text{BACK}}$	78	O	Indicates that the bus is released to an external bus master.
	TEST	87	I	Test pin. Connect to a Vss.
Interrupts	NMI	91	I	Nonmaskable interrupt pin. If this pin is not used, it should be fixed-high.
	$\overline{\text{IRQ7}}$ $\overline{\text{IRQ6}}$ $\overline{\text{IRQ5}}$ $\overline{\text{IRQ4}}$ $\overline{\text{IRQ3}}$ $\overline{\text{IRQ2}}$ $\overline{\text{IRQ1}}$ $\overline{\text{IRQ0}}$	138 139 121 124 80 7 110 112	I	These pins request maskable interrupts.
Address bus	A23 to A0	37 to 15, 13	O	Outputs addresses.
Data bus	D15 to D0	11 to 1, 144 to 140	I/O	Bi-directional bus.
Bus control	$\overline{\text{CS7}}$ $\overline{\text{CS6}}$ $\overline{\text{CS5}}$ $\overline{\text{CS4}}$ $\overline{\text{CS3}}$ $\overline{\text{CS2}}$ $\overline{\text{CS1}}$ $\overline{\text{CS0}}$	131 132 133 134 138 137 136 135	O	Chip select signals for areas 7 to 0. Pins $\overline{\text{CS2}}$ and $\overline{\text{CS1}}$ are not supported by the H8S/2556 Series.
	$\overline{\text{AS}}$	83	O	Indicates that data output on the address bus is valid when this pin is a low level.
	$\overline{\text{RD}}$	82	O	Indicates that an access to the external address space is in progress when this pin is a low level.
	$\overline{\text{HWR}}$	81	O	Strobe signal. Indicates that data on the upper bits (D15 to D8) of the data bus is valid during a write access.
	$\overline{\text{LWR}}$	80	O	Strobe signal. Indicates that data on the upper bits (D7 to D0) of the data bus is valid during a write access.
	$\overline{\text{WAIT}}$	79	I	Requests insertion of wait cycles in a bus cycle when the access is made to the external address space.
16-bit timer-pulse unit (TPU)	TCLKD TCLKC TCLKB TCLKA	109 111 113 114	I	These pins input an external clock.

Type	Symbol	Pin No.	I/O	Function
16-bit timer-pulse unit (TPU)	TIOCA0 TIOCB0 TIOCC0 TIOCD0	116 115 114 113	I/O	Pins for the TGRA_0 to TGRD_0 input capture input, output compare output, or PWM output.
	TIOCA1 TIOCB1	112 111	I/O	Pins for the TGRA_1 and TGRB_1 input capture input, output compare output, or PWM output.
	TIOCA2 TIOCB2	110 109	I/O	Pins for the TGRA_2 and TGRB_2 input capture input, output compare output, or PWM output.
	TIOCA3 TIOCB3 TIOCC3 TIOCD3	101 102 103 104	I/O	Pins for the TGRA_3 and TGRD_3 input capture input, output compare output, or PWM output.
	TIOCA4 TIOCB4	105 106	I/O	Pins for the TGRA_4 and TGRB_4 input capture input, output compare output, or PWM output.
	TIOCA5 TIOCB5	107 108	I/O	Pins for the TGRA_5 and TGRB_5 input capture input, output compare output, or PWM output.
8-bit timer	TMO3 TMO2 TMO1 TMO0	129 130 131 132	O	Compare-match output pins
	TMCI23 TMCI01	133 134	I	Pins for external clock input to the counter
	TMRI23 TMRI01	133 134	I	Counter reset input pins.
Watch dog timer (WDT)	BUZZ	78	O	Outputs pulse signal divided by the watch dog timer.
Serial communi-cation interface (SCI)/ smart card interface	TxD4 TxD3 TxD2 TxD1 TxD0	119 127 74 123 126	O	Data output pins
	RxD4 RxD3 RxD2 RxD1 RxD0	120 128 75 122 125	I	Data input pins
	SCK4 SCK3 SCK2 SCK1 SCK0	121 129 76 121 124	I/O	Clock input/output pins

Type	Symbol	Pin No.	I/O	Function
I^2C bus interface 2 (IIC2)	SCL1 SCL0	123 121	I/O	I^2C clock input/output pins. These pins are capable of driving bus. Pin SCL0 is an NMOS open-drain outputs.
	SDA1 SDA0	124 122	I/O	I^2C data input/output pins. These pins are capable of driving bus. Pin SDA0 is an NMOS open-drain outputs.
A/D converter	AN15 to AN0	55 to 66, 68 to 71	I	Analog input pins for A/D converter.
	$\overline{\text{ADTRG}}$	80	I	Pin for input of an external trigger to start A/D conversion
D/A converter	DA1 DA0	55 56	O	Analog output pins for the D/A converter.
A/D converter, D/A converter	AVcc	73	I	Power supply pin for the A/D and D/A converters. If both converters are not used, connect this pin to the system power supply.
	AVss	67	I	Ground pin for the A/D and D/A converters. Connect this pin to the system power supply (0 V).
	Vref	72	I	Reference voltage input pin for the A/D and D/A converters. If both converters are not used, connect this pin to the system power supply.
IEBus™ controller (IEB)	$\overline{\text{Tx}}$	137	O	Transmit data output pin for the IEB. (Supported only by the H8S/2552 Series.)
	$\overline{\text{Rx}}$	136	I	Receive data input pin for the IEB. (Supported only by the H8S/2552 Series.)
Hitachi controller area network (HCAN)	HTxD	137	O	Pin for CAN bus transmission. (Supported only by the H8S/2556 Series.)
	HRxD	136	I	Pin for CAN bus reception. (Supported only by the H8S/2556 Series.)
I/O ports	P17 to P10	109 to 116	I/O	8-bit I/O pins
	P27 to P20	108 to 101	I/O	8-bit I/O pins
	P37 to P30	119 to 126	I/O	8-bit I/O pins. Pins P34 and P35 are NMOS push-pull outputs.
	P47 to P40	63 to 66, 68 to 71	I	8-bit input pins
	P52 to P50	76 to 74	I/O	3-bit I/O pins
	P77 to P70	127 to 134	I/O	8-bit I/O pins

Type	Symbol	Pin No.	I/O	Function
I/O ports	P97 to P90	55 to 62	I	8-bit input pins
	PA7 to PA0	37 to 30	I/O	8-bit I/O pins
	PB7 to PB0	29 to 22	I/O	8-bit I/O pins
	PC7 to PC0	21 to 15, 13	I/O	8-bit I/O pins
	PD7 to PD0	11 to 4	I/O	8-bit I/O pins
	PE7 to PE0	3 to 1, 144 to 140	I/O	8-bit I/O pins
	PF7 to PF0	85, 83 to 77	I/O	8-bit I/O pins
	PG4 to PG0	135 to 139	I/O	5-bit I/O pins Pins PG3 and PG2 are not supported by the H8S/2556 Series.
	PH7 to PH0	38 to 45	I/O	8-bit I/O pins
	PJ7 to PJ0	46 to 53	I/O	8-bit I/O pins

Section 2 CPU

The H8S/2000 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2000 CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control. This section describes the H8S/2000 CPU. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

2.1 Features

- Upward-compatible with H8/300 and H8/300H CPU
  - Can execute H8/300 and H8/300H CPU object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- 65 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:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes
- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract : 1 state
  - 8 × 8-bit register-register multiply : 12 states
  - 16 ÷ 8-bit register-register divide : 12 states
  - 16 × 16-bit register-register multiply : 20 states
  - 32 ÷ 16-bit register-register divide : 20 states

- Two CPU operating modes
  - Normal mode*
  - Advanced mode
- Power-down state
  - Transition to power-down state by a SLEEP instruction
  - CPU clock speed selection

Note: * Normal mode is not available in this LSI.

2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are shown below.

- Register configuration
  - The MAC register is supported by the H8S/2600 CPU only.
- Basic instructions
  - The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported by the H8S/2600 CPU only.
- The number of execution states of the MULXU and MULXS instructions;

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, and power-down modes, etc., depending on the model.

2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2000 CPU has the following enhancements:

- More general registers and control registers
  - Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2000 CPU has the following enhancements:

- Additional control register
  - One 8-bit control registers have been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space. The mode is selected by the mode pins.

2.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU.

- Address Space

  Linear access is provided to a maximum address space of 64 kbytes.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.
- Instruction Set

  All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.
- Exception Vector Table and Memory Indirect Branch Addresses

  In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. Figure 2.1 shows the structure of the exception vector table in normal mode. For details of the exception vector table, see section 4, Exception Handling.

  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.
- Stack Structure

  When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR) and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: * Normal mode is not available in this LSI.

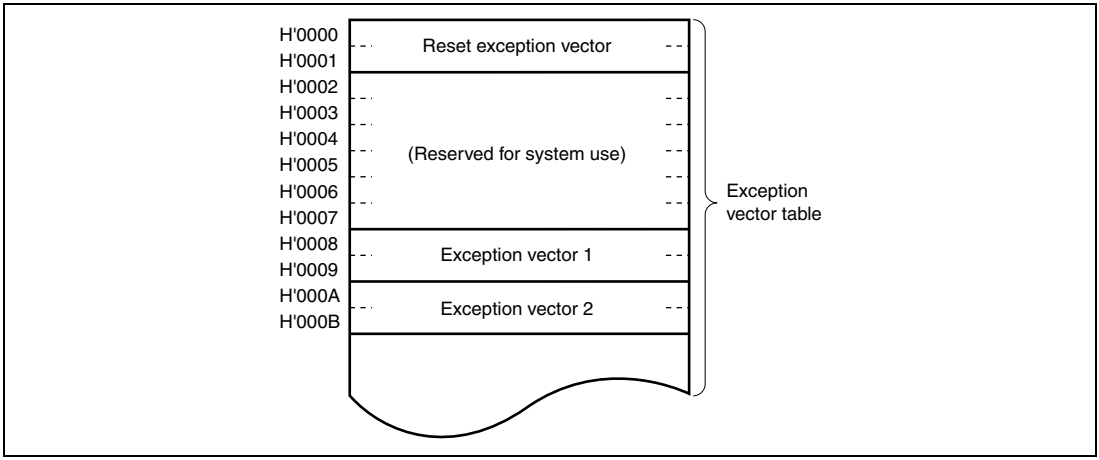


Figure 2.1 Exception Vector Table (Normal Mode)

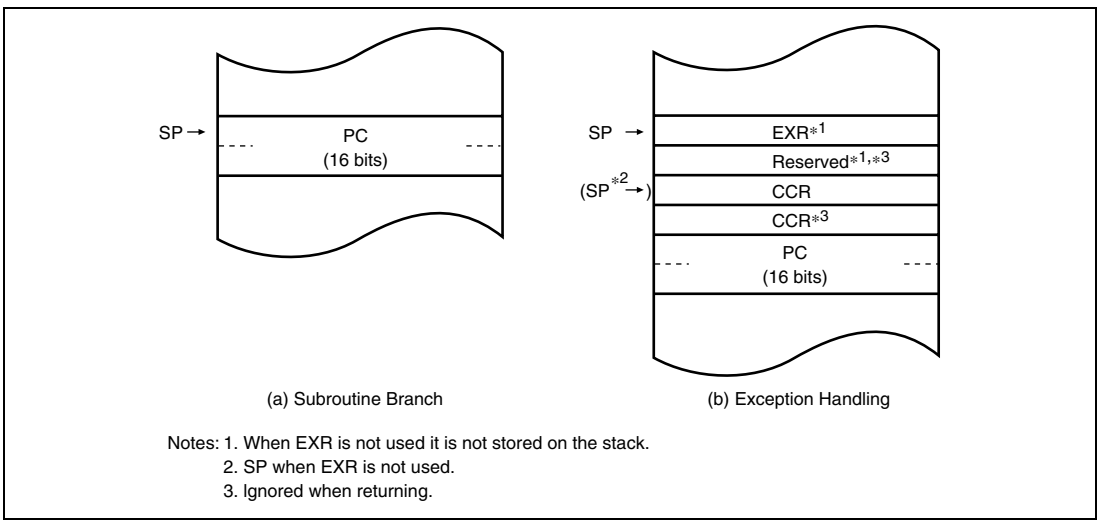


Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access is provided to a maximum 16-Mbyte address space.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers or address registers.
- Instruction Set

  All instructions and addressing modes can be used.

- Exception Vector Table and Memory Indirect Branch Addresses

  In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

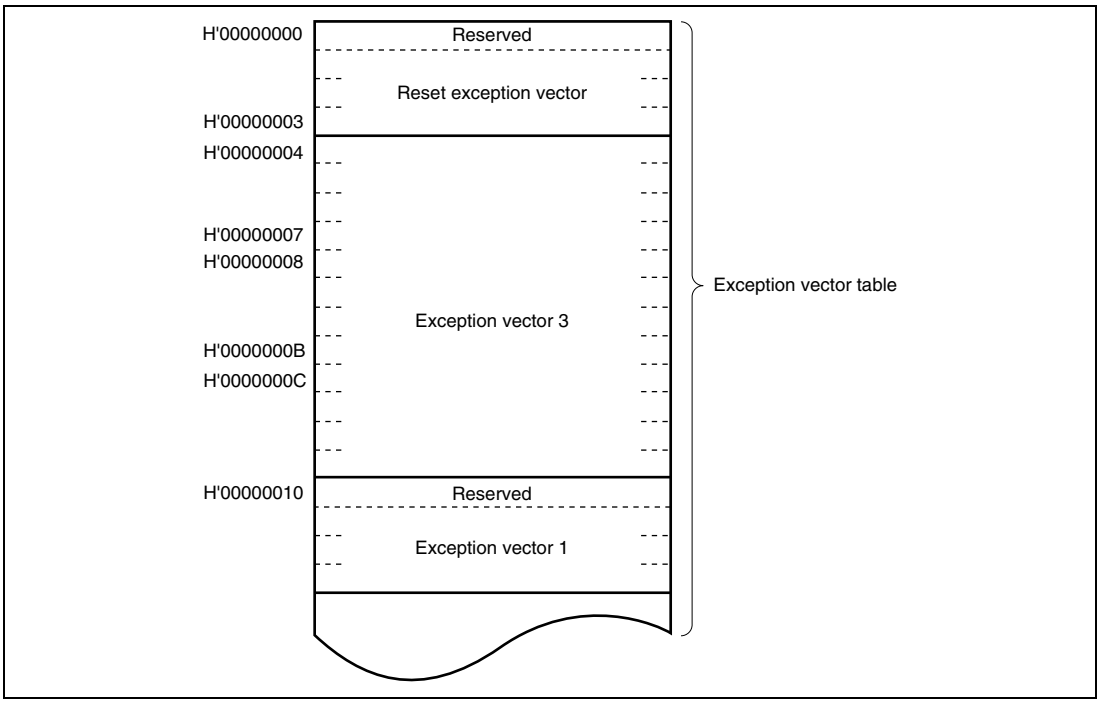
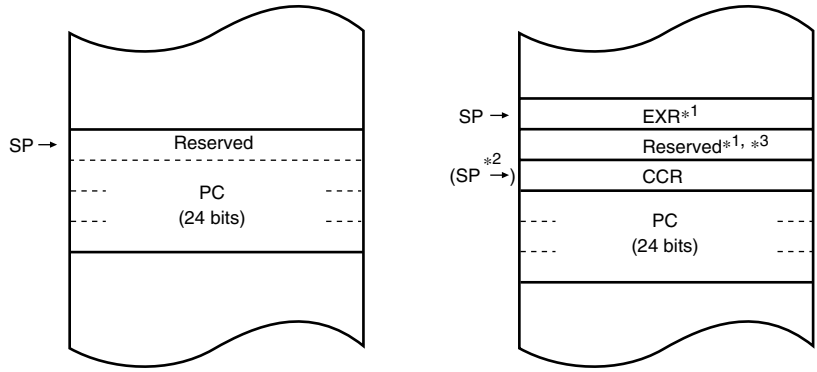


Figure 2.3 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode, the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits is a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the first part of this range is also the exception vector table.

- Stack Structure

  In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



Notes: 1. When EXR is not used it is not stored on the stack.
2. SP when EXR is not used (The H8S/2264 Series SP always points here).
3. Ignored when returning.

Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.5 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

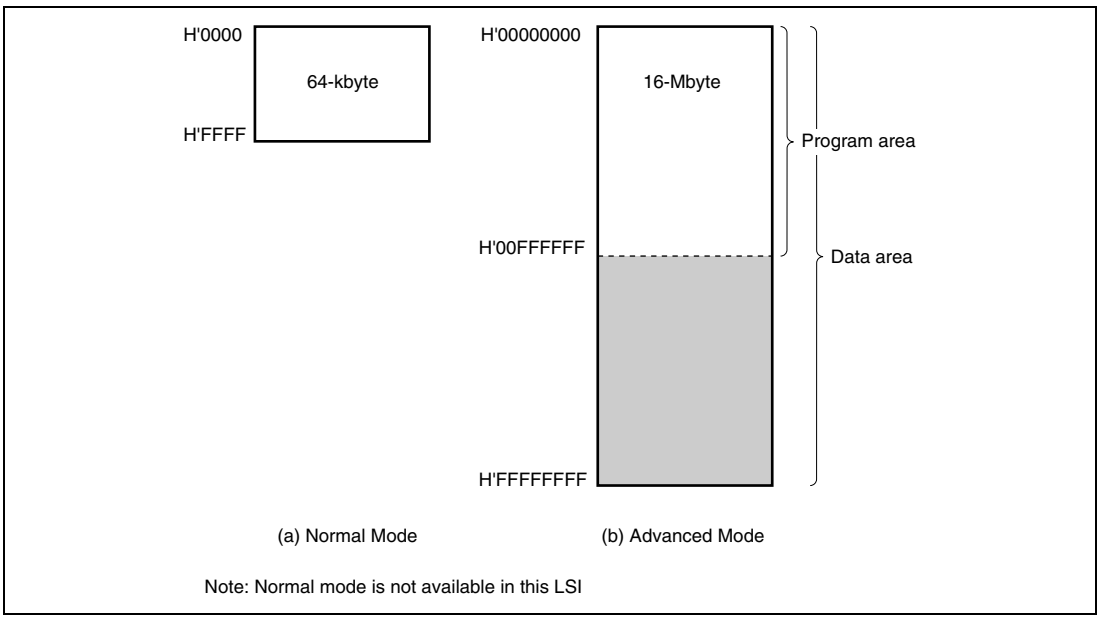


Figure 2.5 Memory Map

2.4 Register Configuration

The H8S/2000 CPU has the internal registers shown in figure 2.6. There are two types of registers: general registers and control registers. Control registers are a 24-bit program counter (PC), an 8-bit extended control register (EXR), and an 8-bit condition code register (CCR).

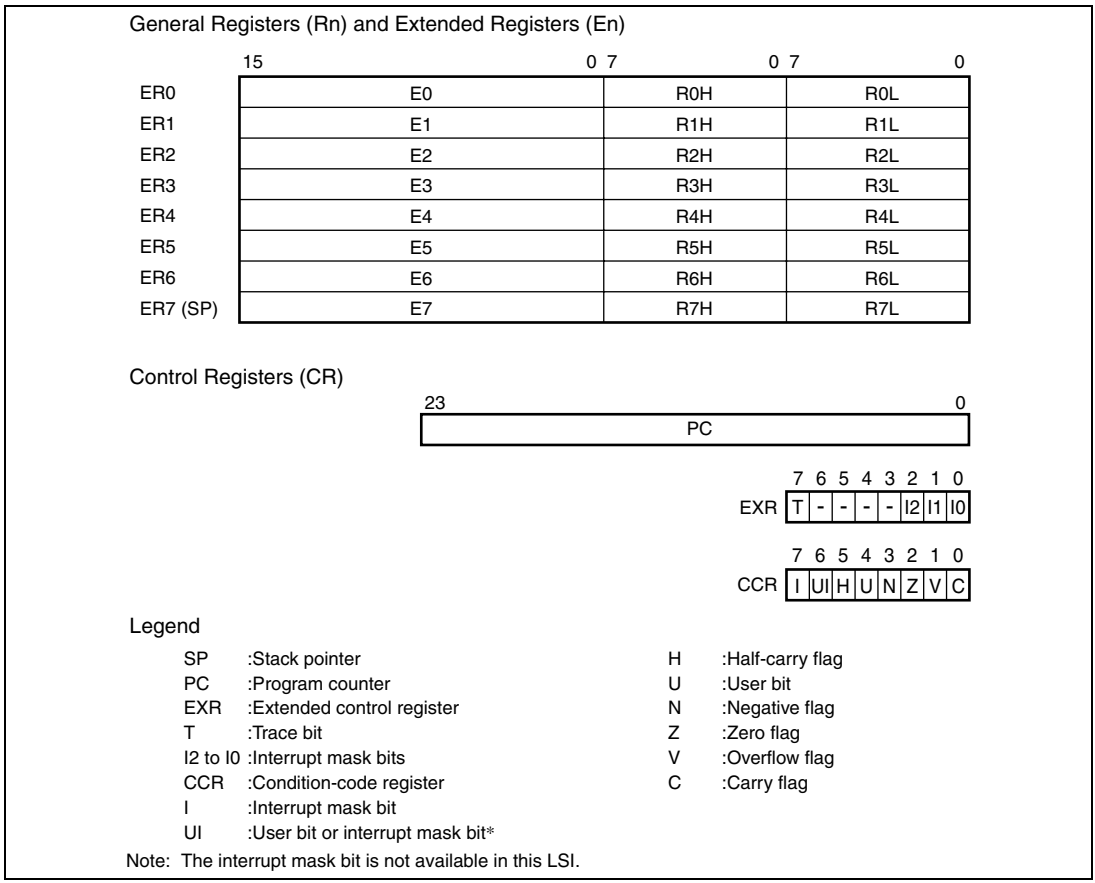


Figure 2.6 CPU Registers

2.4.1 General Registers

The H8S/2000 CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data 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. Figure 2.7 illustrates the usage of the general registers.

When the general registers are used as 32-bit registers or 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 2egisters.

The usage of each register can be selected independently.

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 2.8 shows the stack.

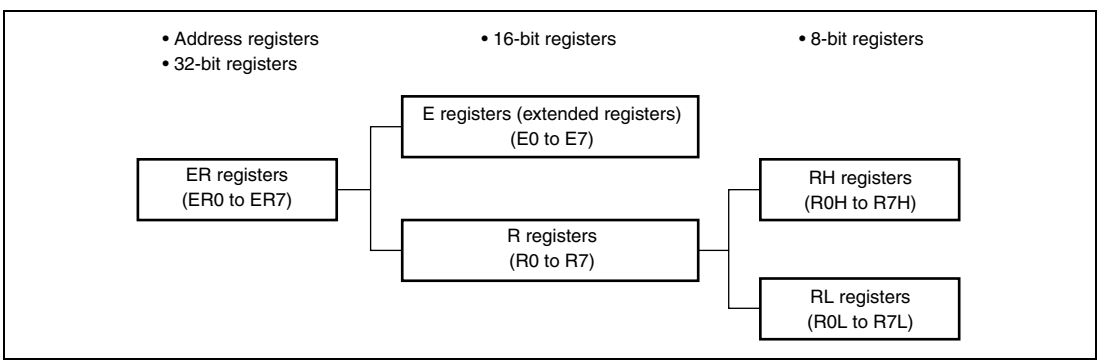


Figure 2.7 Usage of General Registers

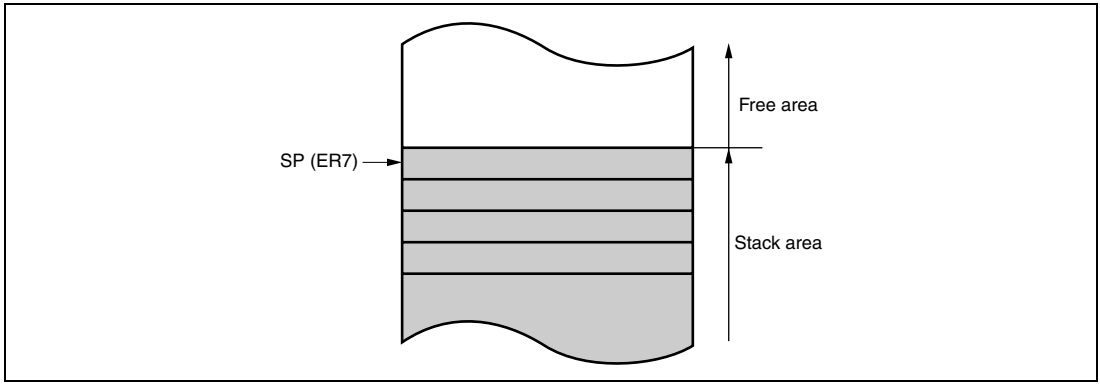


Figure 2.8 Stack Status

2.4.2 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.)

2.4.3 Extended Control Register (EXR)

EXR is an 8-bit register that manipulates the LDC, STC, ANDC, ORC, and XORC instructions. When these instructions except for the STC instruction is executed, all interrupts including NMI will be masked for three states after execution is completed.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.
1	I1	1	R/W	
0	I0	1	R/W	

2.4.4 Condition-Code Register (CCR)

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

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit 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 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit cannot be used as an interrupt mask bit in this LSI.
5	H	Undefined	R/W	Half-Carry Flag 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.
4	U	Undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.

Bit	Bit Name	Initial Value	R/W	Description
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.4.5 Initial Values of CPU Registers

Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace bit in EXR to 0, and sets the interrupt mask bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (ER7) is not initialized. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

2.5 Data Formats

The H8S/2000 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

Figure 2.9 shows the data formats in general registers.

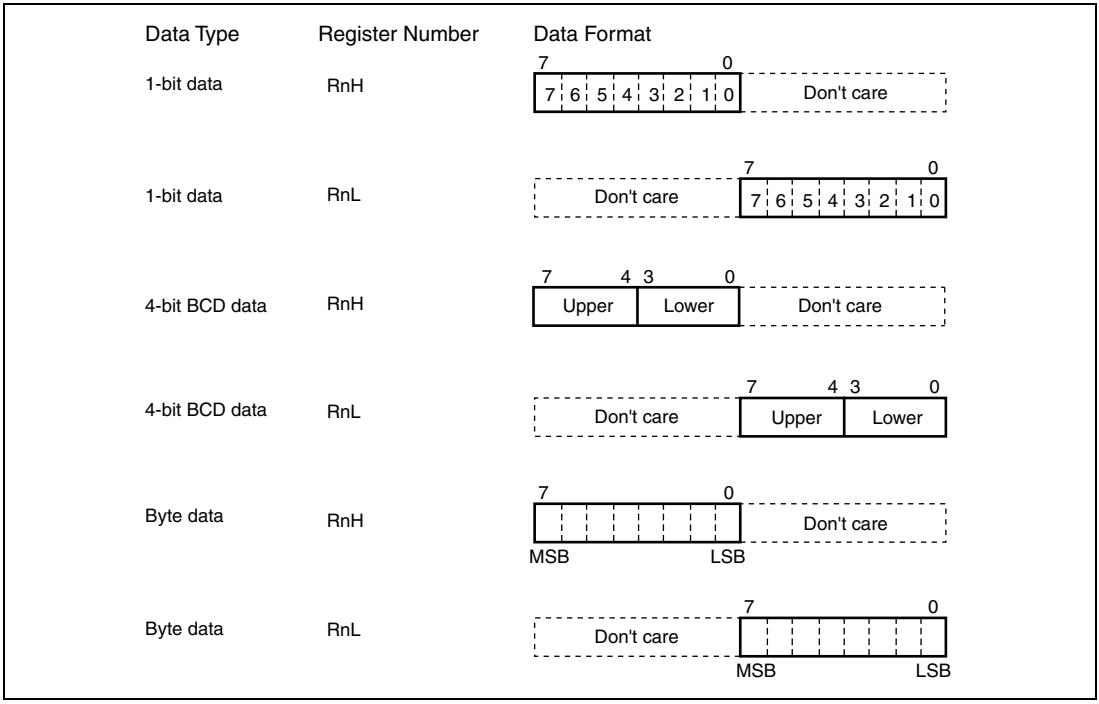


Figure 2.9 General Register Data Formats (1)

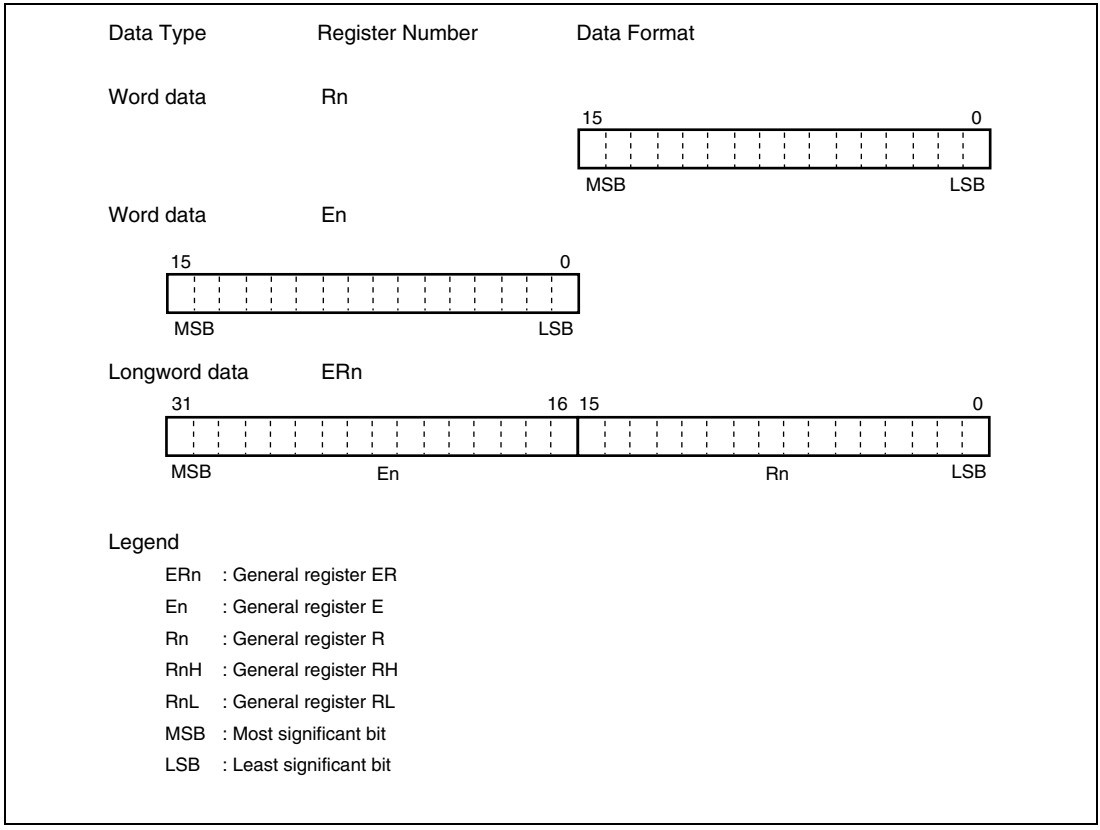


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2000 CPU can access word data and longword data in 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.

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

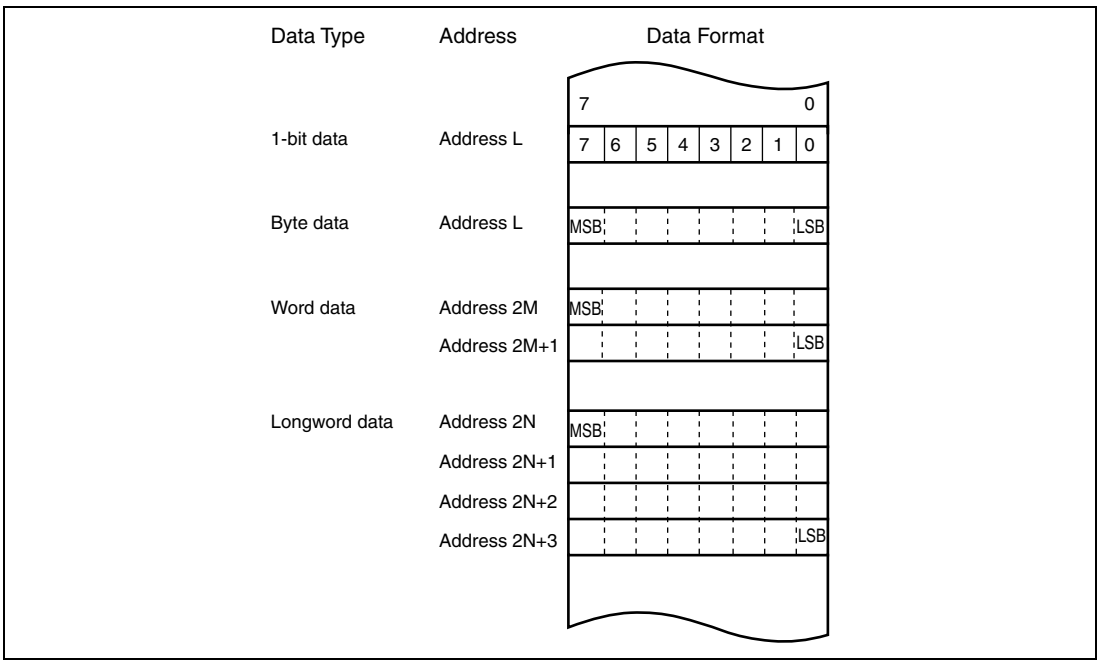


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function in table 2.1.

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP*[1], PUSH*[1]	W/L	
	LDM, STM	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	19
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS*[4]	B	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1
			Total: 65

Notes: B-byte; W-word; L-longword.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

2.6.1 Table of Instructions Classified by Function

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

Table 2.2 Operation Notation

Symbol	Description
Rd	General register (destination) *
Rs	General register (source) *
Rn	General register*
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR
→	Move
~	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

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

Table 2.3 Data Transfer Instructions

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

Note: * Refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.4 Arithmetic Operations Instructions (1)

Instruction	Size*	Function
ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → 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 byte data in a general register. Use the SUBX or ADD instruction.)
ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry on byte data in two general registers, or on immediate data and data in a general register.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.

Note: * Refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.4 Arithmetic Operations Instructions (2)

Instruction	Size*[1]	Function
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 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 bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7) to 1.

Notes: 1. Refers to the operand size.
B: Byte
W: Word
L: Longword
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Table 2.5 Logic Operations Instructions

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

Note: * Refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.6 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shifts are possible.
SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 1-bit or 2-bit shifts are possible.
ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotations are possible.
ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotations are possible.

Note: * Refers to the operand size.
B: Byte
W: Word
L: Longword

Table 2.7 Bit Manipulation Instructions (1)

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

Note: * Refers to the operand size.
B: Byte

Table 2.7 Bit Manipulation Instructions (2)

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

Note: * Refers to the operand size.
B: Byte

Table 2.8 Branch Instructions

Instruction	Size	Function
Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA(BT)	Always (true)	Always
BRN(BF)	Never (false)	Never
BHI	High	$C \vee Z = 0$
BLS	Low or same	$C \vee Z = 1$
BCC(BHS)	Carry clear (high or same)	$C = 0$
BCS(BLO)	Carry set (low)	$C = 1$
BNE	Not equal	$Z = 0$
BEQ	Equal	$Z = 1$
BVC	Overflow clear	$V = 0$
BVS	Overflow set	$V = 1$
BPL	Plus	$N = 0$
BMI	Minus	$N = 1$
BGE	Greater or equal	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

JMP	—	Branches unconditionally to a specified address.
BSR	—	Branches to a subroutine at a specified address.
JSR	—	Branches to a subroutine at a specified address.
RTS	—	Returns from a subroutine

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 a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the source operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically XORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: * Refers to the operand size.
B: Byte
W: Word

Table 2.10 Block Data Transfer Instructions

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L–1 → R4L Until R4L = 0 else next;
EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ • @ER6+ R4–1 • R4 Until R4 = 0 else next; Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6. Execution of the next instruction begins as soon as the transfer is completed.

2.6.2 Basic Instruction Formats

This LSI 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).

Figure 2.11 shows examples of instruction formats.

- 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 four bits of the instruction. Some instructions have two operation fields.
- Register Field

  Specifies a general register. Address registers are specified by 3 bits, and 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.
- Condition Field

  Specifies the branching condition of Bcc instructions.

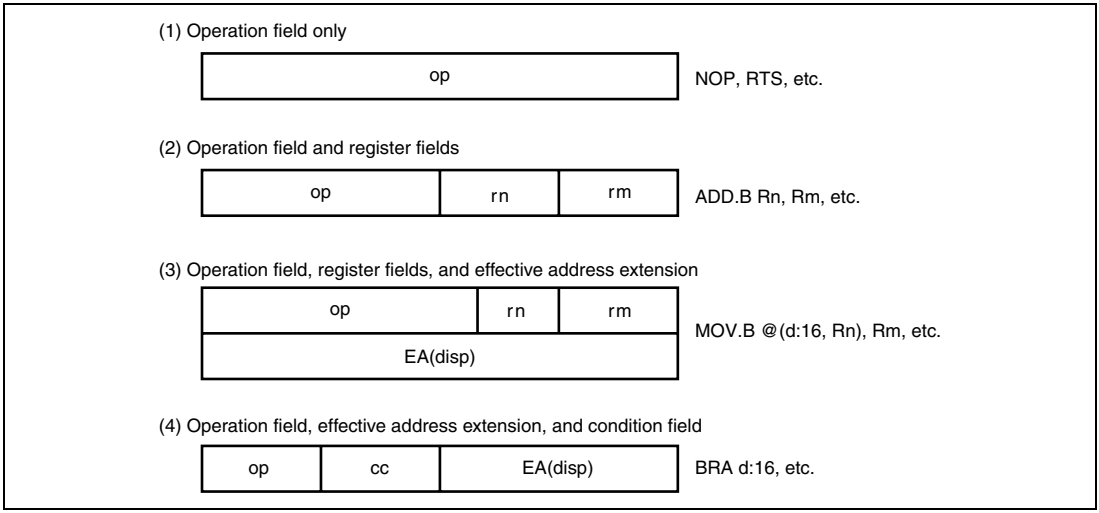


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2000 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 the absolute 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:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general 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.

2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)

A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

2.7.4 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) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For the word or longword transfer instructions, 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 result is 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 transfer instruction, or 4 for longword transfer instruction. For the word or longword transfer instructions, the register value should be even.

2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

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), 24 bits long (@aa:24), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute address can access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

Table 2.12 Absolute Address Access Ranges

Absolute Address		Normal Mode*	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

Note: * Normal mode is not available in this LSI.

2.7.6 Immediate—#xx:8, #xx:16, or #xx:32

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

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

2.7.7 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 is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). 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.

2.7.8 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 upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode*, H'000000 to H'0000FF in advanced mode). In normal mode, the memory operand is a word operand and the branch address is 16 bits long. In advanced mode, the memory operand is a longword operand, the first byte of which is assumed to be 0 (H'00).

Note that the first part of the address range is also the exception vector area. For further details, refer to section 4, Exception Handling.

If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

Note: * Normal mode is not available in this LSI.

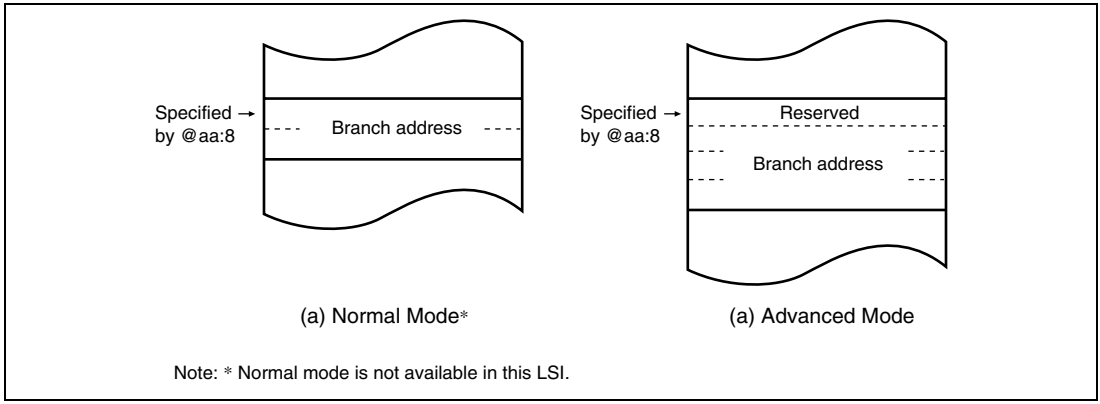


Figure 2.12 Branch Address Specification in Memory Indirect Mode

2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode. In normal mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Table 2.13 Effective Address Calculation (1)

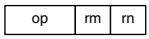
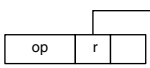
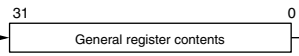
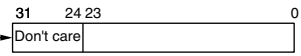
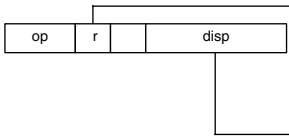
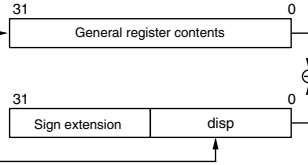
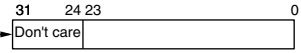
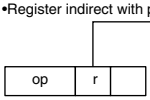
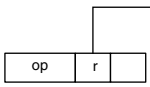
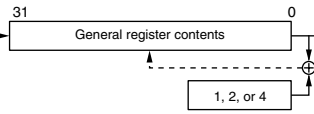
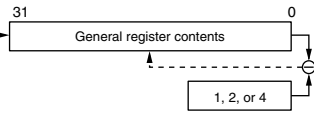
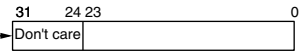
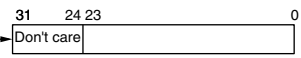
No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct(Rn) 		Operand is general register contents.
2	Register indirect(@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 		
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+  •Register indirect with pre-decrement @-ERn 	  Operand Size / Offset Byte 1 Word 2 Longword 4	 

Table 2.13 Effective Address Calculation (2)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8 op abs @aa:16 op abs @aa:24 op abs @aa:32 op abs		31 24 23 8 7 0 Don't care H'FFFF 31 24 23 16 15 0 Don't care Sign extension 31 24 23 0 Don't care 31 24 23 0 Don't care
6	Immediate #xx:8/#xx:16/#xx:32 op IMM		Operand is immediate data.
7	Program-counter relative @(d:8,PC) @(d:16,PC) op disp	23 0 PC contents 23 0 Sign extension disp	31 24 23 0 Don't care
8	Memory indirect @@aa:8 • Normal mode* op abs ^ • Advanced mode op abs	31 8 7 0 H'000000 abs 15 0 Memory contents 31 8 7 0 H'000000 abs 31 0 Memory contents	31 24 23 16 15 0 Don't care H'00 31 24 23 0 Don't care

Note: * Normal mode is not available in this LSI.

2.8 Processing States

The H8S/2000 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and power-down state. Figure 2.13 indicates the state transitions.

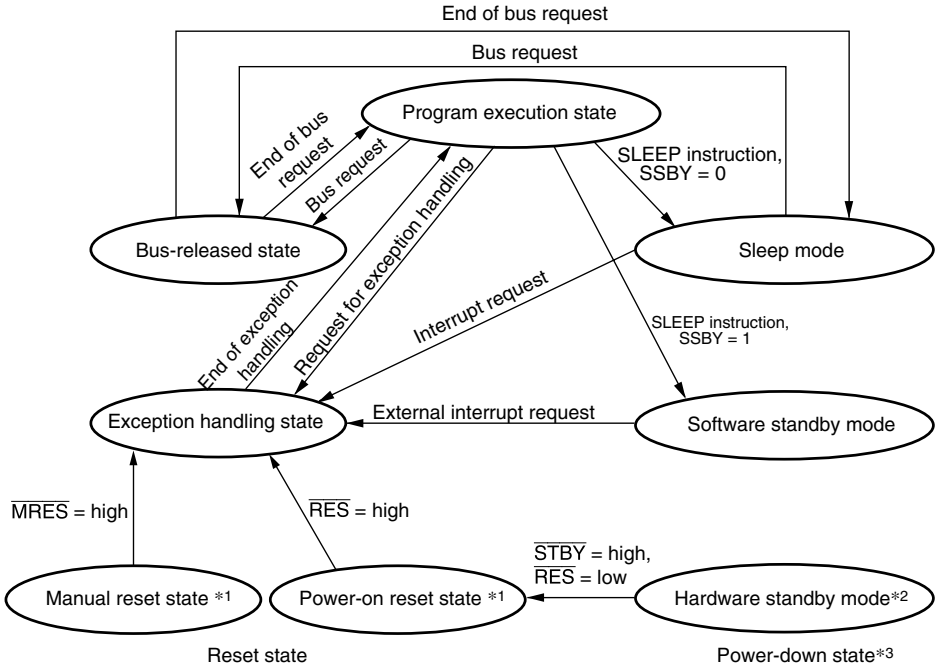
- Reset State
  In this state, the CPU and all on-chip peripheral modules are initialized and not operating. When the $\overline{\text{RES}}$ input goes low, all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high. For details, refer to section 4, Exception Handling.

  The reset state can also be entered by a watchdog timer overflow.

- Exception-Handling State
  The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further details, refer to section 4, Exception Handling.
- Program Execution State
  In this state, the CPU executes program instructions in sequence.
- Bus-Released State
  In a product which has a bus master other than the CPU, such as a data transfer controller (DTC), the bus-released state occurs when the bus has been released in response to a bus request from a bus master other than the CPU.

  While the bus is released, the CPU halts operations.
- Power-down State

  This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For further details, refer to section 22, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.
3. Apart from these states, there is also watch mode. For details, see section 22, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Notes

2.9.1 TAS Instruction

Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction. The TAS instruction is not generated by the Hitachi H8S and H8/300 Series C/C++ compilers. If the TAS instruction is used as a user-defined intrinsic function, ensure that only register ER0, ER1, ER4, or ER5 is used.

2.9.2 STM/LDM Instruction

With the STM or LDM instruction, the ER7 register is used as the stack pointer, and thus cannot be used as a register that allows save (STM) or restore (LDM) operation.

With a single STM or LDM instruction, two to four registers can be saved or restored. The available registers are as follows:

For two registers: ER0 and ER1, ER2 and ER3, or ER4 and ER5

For three registers: ER0 to ER2, or ER4 to ER6

For four registers: ER0 to ER3

For the Hitachi H8S or H8/300 Series C/C++ Compiler, the STM/LDM instruction including ER7 is not created.

2.9.3 Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions are used to read data in byte-wise, operate the data in bit-wise, and write the result of the bit-wise operation in bit-wise again. Therefore, special care is necessary to use these instructions for the registers and the ports that include write-only bit.

The BCLR instruction can be used to clear to 0 the flags in the internal I/O registers. In this time, if it is obvious that the flag has been set to 1 in the interrupt handler, there is no need to read the flag beforehand.

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI supports two types of operating mode (modes 6 and 7). Pin functions are changed according to each operating mode. The operation mode is determined by the setting of mode pins (MD2 to MD0). Mode 6 is the external expansion mode, which allows external memory and peripheral device to be accessed. In external expansion mode, the bus controller sets address space of 8 bits or 16 bits for each area after the program is started to execute. Making one of the areas a 16-bit address space leads to 16-bit bus mode and making all the area 8-bit access space leads to 8-bit bus mode.

In mode 7, the external address space cannot be used. Mode pins should not be changed during operations.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
							Initial Width	Max. Width
6	1	1	0	Advanced mode	On-chip ROM valid expansion mode	Enabled	8 bits	16 bits
7	1	1	1	Advanced mode	Single-chip mode	Enabled	—	—

3.2 Register Descriptions

The following registers are related to the operating mode.

- Mode control register (MDCR)
- System control register (SYSCR)

3.2.1 Mode Control Register (MDCR)

MDCR monitors the current operating mode.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	—	1	R/W	Reserved This bit is always read as 1 and cannot be modified.
6 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
2	MDS2	—*	R	Mode Select 2 to 0 These bits indicate the input levels at mode pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD2 to MD0, respectively. MDS2 to MDS0 are read-only bits and cannot be modified. The input levels at mode pins MD2 to MD0 are latched into these bits when MDCR is read. These latches are canceled by a power-on reset, but retained by a manual reset.
1	MDS1	—*	R	
0	MDS0	—*	R	

Note: * Determined by the setting of pins MD2 to MD0.

3.2.2 System Control Register (SYSCR)

SYSCR performs the selection of interrupt control mode, the selection of NMI detection edge, the selection of enable/disable of $\overline{\text{MRES}}$ pin input, and the selection of valid/invalid of on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	—	0	R/W	Reserved The write value should always be 0.
6	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
5 4	INTM1 INTM0	0 0	R/W R/W	Select interrupt control mode of the interrupt controller. For interrupt control mode, refer to section 5.5.1, Interrupt Control Modes and Interrupt Operation. 00: Interrupt control mode 0 01: Setting prohibited 10: Interrupt control mode 2 11: Setting prohibited
3	NMIEG	0	R/W	NMI Edge Select Performs input edge selection of the NMI pin. 0: Interrupt request is generated at the falling edge of NMI input. 1: Interrupt request is generated at the rising edge of NMI input.
2	MRESE	0	R/W	Manual Reset Selection Bit Selects enable/disable of the $\overline{\text{MRES}}$ pin input. 0: Disables manual reset. 1: Enables manual reset. The $\overline{\text{MRES}}$ pin input is enabled.
1	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
0	RAME	1	R/W	RAM Enable Selects valid/invalid of the on-chip RAM. The RAME bit is initialized when a reset is canceled. 0: The on-chip RAM is disabled. 1: The on-chip RAM is enabled.

3.3 Operating Mode

3.3.1 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is valid. Immediately after a reset, ports A, B, and C become input ports. The AE3 to AEO bits in PFCR allow enable/disable setting of the address (A23 to A8) output, regardless of the corresponding DDR value. The pin which is disabled of the address output at ports A and B becomes an output port when the corresponding DDR is set to 1.

The address (A7 to A0) is output when the corresponding DDR is set to 1 at port C.

Ports D and E are data buses, and a part of the port F is the bus control signal.

Immediately after a reset, 8-bits bus mode is set and all the areas become 8-bit access space. However, when any of the areas is set to 16-bit access space by the bus controller, 16-bit bus mode is set and port E becomes the data bus.

3.3.2 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is valid and the external address space cannot be accessed.

All the I/O port can be used as an input/output port.

3.3.3 Pin Functions

Table 3.2 shows the pin functions in modes 6 and 7.

Table.3.2 Pin Function in Each Operating Mode

Port		Mode 6	Mode 7
Port A		P*/A	P
Port B		P*/A	P
Port C		P*/A	P
Port D		D	P
Port E		P*/D	P
Port F	PF7	P/C*	P*/C
	PF6 to PF4	C	P
	PF3	P*/C	
	PF2 to PF0	P*/C	

Legend

P: Input/output port

A: Address bus output

D: Data bus Input/output

C: Control signal, clock Input/output

*: Immediately after a reset

3.4 Address Map in Each Operating Mode

Figures 3.1 shows the address map in each operating mode.

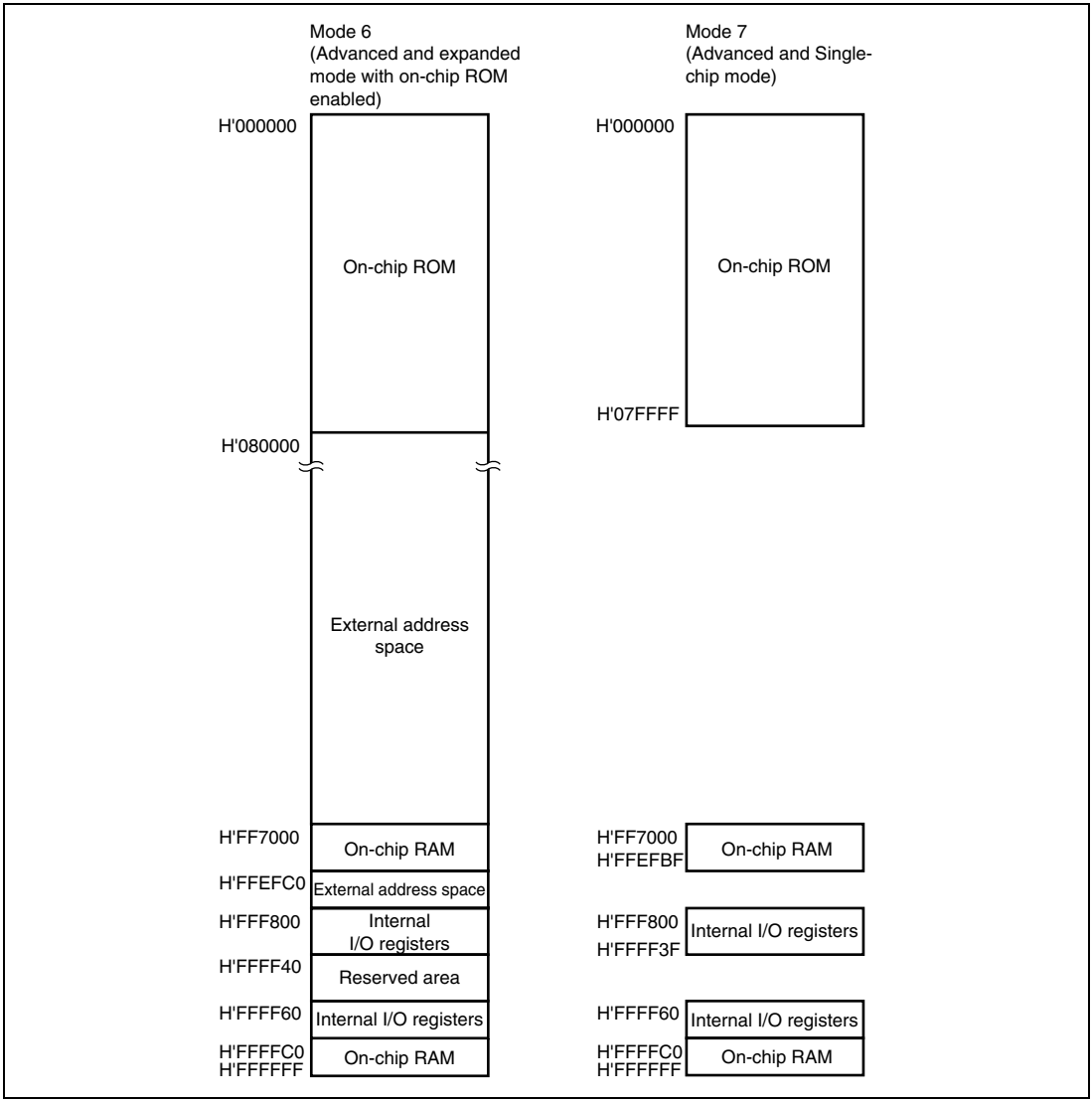


Figure 3.1 Address Map in Each Operating Mode

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trace, 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 order of priority. Trap instruction exception handling requests are accepted at all times in program execution state.

The exception source, the stack structure, and the operation of the CPU vary depending on the interrupt control mode set by the INTM0 and INTM1 bits in SYSCR.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ↑ Low	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ and $\overline{\text{MRES}}$ pins, or when the watchdog timer overflows. The CPU enters the power-on reset state when the $\overline{\text{RES}}$ pin is low. The CPU enters the manual reset state when the $\overline{\text{MRES}}$ pin is low.
	Trace	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit is set to 1. Trace is enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA). Trap instruction exception handling requests are accepted at all times in program execution state.

4.2 Exception Sources and Exception Vector Table

Different vector address is assigned to each exception source. Table 4.2 lists the exception sources and their vector addresses.

Table 4.2 Exception Handling Vector Table

Exception Source		Vector Number	Vector Address*[1] Advanced Mode
Power-on reset		0	H'0000 to H'0003
Manual reset		1	H'0004 to H'0007
Reserved for system use		2	H'0008 to H'000B
		3	H'000C to H'000F
		4	H'0010 to H'0013
Trace		5	H'0014 to H'0017
Direct transitions*[3]		6	H'0018 to H'001B
External interrupt (NMI)		7	H'001C to H'001F
Trap instruction (four sources)		8	H'0020 to H'0023
		9	H'0024 to H'0027
		10	H'0028 to H'002B
		11	H'002C to H'002F
Reserved for system use		12	H'0030 to H'0033
		13	H'0034 to H'0037
		14	H'0038 to H'003B
		15	H'003C to H'003F
External interrupt	IRQ0	16	H'0040 to H'0043
	IRQ1	17	H'0044 to H'0047
	IRQ2	18	H'0048 to H'004B
	IRQ3	19	H'004C to H'004F
	IRQ4	20	H'0050 to H'0053
	IRQ5	21	H'0054 to H'0057
	IRQ6	22	H'0058 to H'005B
	IRQ7	23	H'005C to H'005F
Internal interrupt*[2]		24 | 127	H'0060 to H'0063 | H'01FC to H'01FF

Notes: 1. Indicates lower 16 bits of the address.
2. For details on the internal interrupt vector table, see section 5.4.3, Interrupt Exception Handling Vector Table.
3. Direct transitions are not supported in this LSI.

4.3 Reset

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin goes low, all processing halts and this LSI enters the reset state. A reset initializes the internal state of the CPU and the registers of the on-chip peripheral modules. This LSI enters interrupt control mode 0 immediately after a reset.

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin goes high from the low state, this LSI starts reset exception handling.

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

4.3.1 Types of Reset

The LSI supports two types of resets: power-on reset and manual reset.

Table 4.3 shows the types of reset. Set to power-on reset when the power is tuned on.

The CPU internal status is initialized both by the power-on reset and manual reset. By power-on reset, all registers of the on-chip peripheral modules are initialized; by manual reset, registers of the on-chip peripheral modules, except for the bus controller and I/O ports, are initialized. The status of the bus controller and I/O ports is maintained.

By manual reset, on-chip peripheral modules are initialized and thus ports used as input/output pins of the on-chip peripheral modules are switched to input/output ports controlled by DDR and DR.

Table 4.3 Types of Reset

	Reset Shift Conditions		Internal State	
Types	**$\overline{\text{MRES}}$**	**$\overline{\text{RES}}$**	**CPU**	**On-Chip Peripheral Modules**
Power-on reset	*	Low	Initialized	Initialized
Manual reset	Low	High	Initialized	Initialized except for bus controller and I/O ports

Note: * Don't care

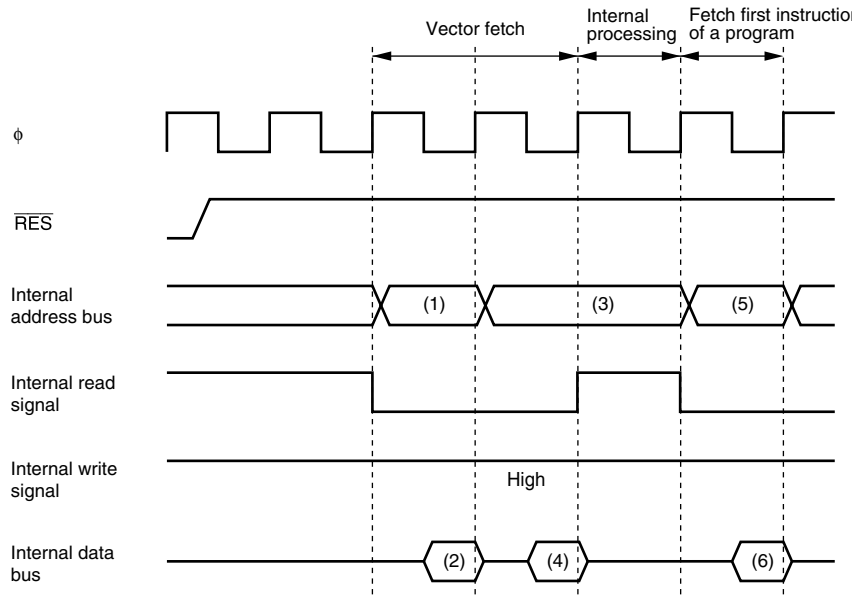
4.3.2 Reset Exception Handling

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin goes low, this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low for at least 20 ms at power-up. To reset the chip during operation, hold the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low for at least 20 states.

When the $\overline{RES}$ or $\overline{MRES}$ pin goes high after being held low for the necessary time, this LSI starts reset exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, the T bit in EXR is cleared to 0, and the I bits in EXR and CCR are set to 1.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figures 4.1 shows an example of the reset sequence.



(1)(3) Reset exception handling vector address (At a reset, (1) = H'000000; (3) = H'000002)
(2)(4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First instruction of a program

Figure 4.1 Reset Sequence (Advanced Mode with On-Chip ROM Enabled)

4.3.3 Interrupts after Reset

If an interrupt is accepted after a reset and before the stack pointer (SP) is initialized, the 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. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: SP).

4.3.4 State of On-Chip Peripheral Modules after Reset Release

After reset release, MSTPCRA is initialized to H'3F, MSTPCRB and MSTPCRC are initialized to H'FF, and all modules except the DTC enter module stop mode. Consequently, on-chip peripheral module registers cannot be read or written to. Register reading and writing is enabled when the module stop mode is exited.

4.4 Trace Exception Handling

Trace is enabled in interrupt control mode 2. Trace mode is not entered in interrupt control mode 0, irrespective of the state of the T bit. For details on the interrupt control mode, see section 5, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is entered. In trace mode, a trace exception handling occurs on completion of each instruction. After execution of trace exception handling, the T bit in EXR is cleared to 0 and trace mode is canceled. Trace mode is not affected by interrupt masking. Table 4.4 shows the state of CCR and EXR after execution of trace exception handling. Interrupts are accepted even within the trace exception handling routine.

The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes. Trace exception handling is not carried out after execution of the RTE instruction.

Table 4.4 State of CCR and EXR after Trace Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	Trace exception handling cannot be used.			
2	1	—	—	0

Legend

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution

4.5 Interrupt Exception Handling

Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control. For details, refer to section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution begins from that address.

4.6 Trap Instruction Exception Handling

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

Trap instruction exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.5 shows the state of CCR and EXR after execution of trap instruction exception handling.

Table 4.5 State of CCR and EXR after Trap Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

Legend

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution

4.7 Stack State after Exception Handling

Figures 4.2 shows the stack state after completion of trap instruction exception handling and interrupt exception handling.

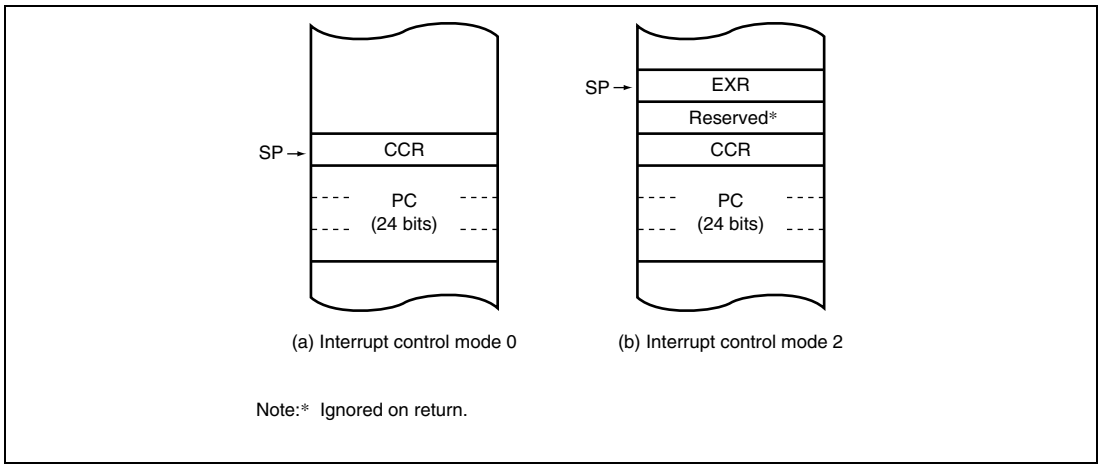


Figure 4.2 Stack State after Exception Handling (Advanced Mode)

4.8 Usage Note

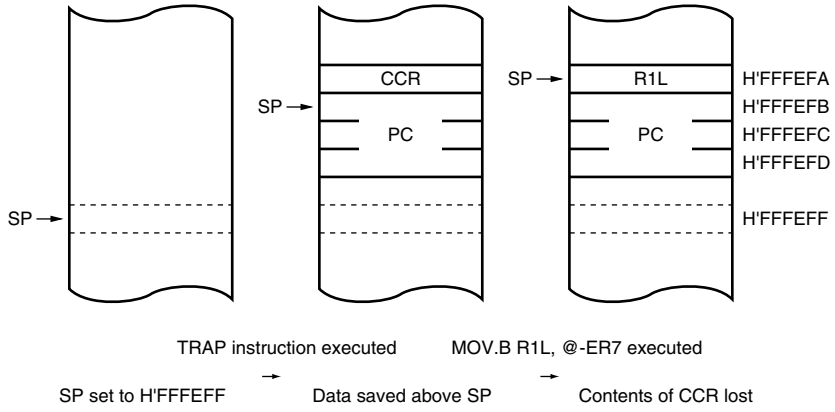
When accessing word data or longword data, this LSI assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP: ER7) should always be kept even. Use the following instructions to save 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 the SP to an odd value may lead to a malfunction. Figure 4.3 shows an example of what happens when the SP value is odd.



Legend
CCR: Condition code register
PC: Program counter
R1L: General register R1L
SP: Stack pointer

Note: This diagram illustrates an example in interrupt control mode 0 and advanced mode.

Figure 4.3 Operation when SP Value Is Odd

Section 5 Interrupt Controller

5.1 Features

This LSI controls interrupts with the interrupt controller. The interrupt controller has the following features:

- Two interrupt control modes
  - Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- Priorities settable with IPR
  - An interrupt priority register (IPR) is provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI. NMI is assigned the highest priority level of 8, and can be accepted at all times.
- Independent vector addresses
  - All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Nine external interrupt pins
  - NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI.
  - Falling edge, rising edge, both edges, or level sensing can be independently selected for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- DTC control
  - The DTC can be activated by an interrupt request.

A block diagram of the interrupt controller is shown in figure 5.1.

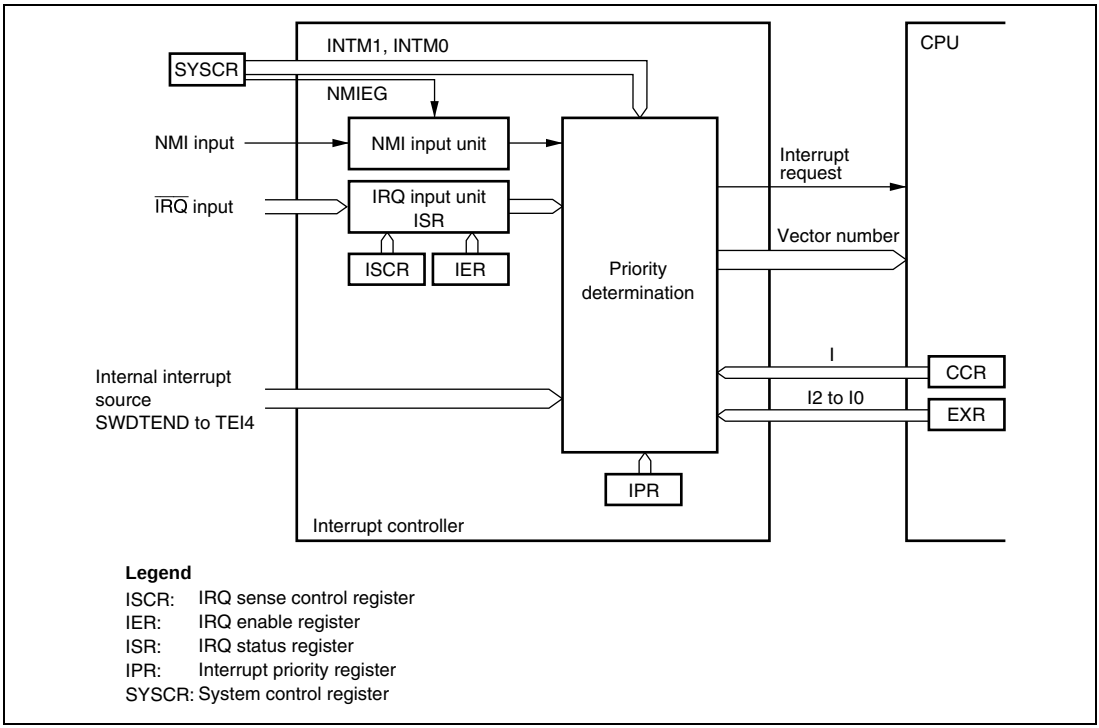


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Pin Configuration

Name	I/O	Function
NMI	Input	Nonmaskable external interrupt Rising edge or falling edge can be selected.
$\overline{\text{IRQ7}}$	Input	Maskable external interrupt
$\overline{\text{IRQ6}}$	Input	Rising edge, falling edge, both edges, or level sensing can be selected.
$\overline{\text{IRQ5}}$	Input	
$\overline{\text{IRQ4}}$	Input	
$\overline{\text{IRQ3}}$	Input	
$\overline{\text{IRQ2}}$	Input	
$\overline{\text{IRQ1}}$	Input	
$\overline{\text{IRQ0}}$	Input	

5.3 Register Descriptions

The interrupt controller has the following registers. For details on the system control register, refer to section 3.2.2, System Control Register (SYSCR).

- System control register (SYSCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)

- Interrupt priority register L (IPRL)
- Interrupt priority register M (IPRM)
- Interrupt priority register O (IPRO)

5.3.1 Interrupt Priority Registers A to M, and O (IPRA to IPRM, IPRO)

The IPR registers are fourteen 8-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI. The correspondence between interrupt sources and IPR settings is shown in section 5.4.3, Interrupt Exception Handling Vector Table. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 6 to 4 and 2 to 0 sets the priority of the corresponding interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
6	IPR6	1	R/W	These bits set the priority of the corresponding interrupt source. 000: Priority level 0 (lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (highest)
5	IPR5	1	R/W	
4	IPR4	1	R/W	
3	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
2	IPR2	1	R/W	These bits set the priority of the corresponding interrupt source. 000: Priority level 0 (lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (highest)
1	IPR1	1	R/W	
0	IPR0	1	R/W	

5.3.2 IRQ Enable Register (IER)

IER controls the enabling and disabling of interrupt requests $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when this bit is 1.
6	IRQ6E	0	R/W	IRQ6 Enable The IRQ6 interrupt request is enabled when this bit is 1.
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this bit is 1.
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this bit is 1.
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this bit is 1.
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this bit is 1.
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this bit is 1.
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this bit is 1.

5.3.3 IRQ Sense Control Registers H and L (ISCRH and ISCRL)

The ISCR registers select the source that generates an interrupt request at $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ pins. Specifiable sources are the falling edge, rising edge, both edges, and level sensing.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7SCB	0	R/W	IRQ7 Sense Control B
14	IRQ7SCA	0	R/W	IRQ7 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ7}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ7}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ7}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ7}}$ input

Bit	Bit Name	Initial Value	R/W	Description
13	IRQ6SCB	0	R/W	IRQ6 Sense Control B
12	IRQ6SCA	0	R/W	IRQ6 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ6}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ6}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ6}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ6}}$ input
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ5}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ5}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ5}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ4}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ4}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ4}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ4}}$ input
7	IRQ3SCB	0	R/W	IRQ3 Sense Control B
6	IRQ3SCA	0	R/W	IRQ3 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ3}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ3}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ3}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ2}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ2}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ2}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input
3	IRQ1SCB	0	R/W	IRQ1 Sense Control B
2	IRQ1SCA	0	R/W	IRQ1 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ1}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ1}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ1}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ1}}$ input
1	IRQ0SCB	0	R/W	IRQ0 Sense Control B
0	IRQ0SCA	0	R/W	IRQ0 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ0}}$ input level low 01: Interrupt request is generated at falling edge of $\overline{\text{IRQ0}}$ input 10: Interrupt request is generated at rising edge of $\overline{\text{IRQ0}}$ input 11: Interrupt request is generated at both falling and rising edges of $\overline{\text{IRQ0}}$ input

5.3.4 IRQ Status Register (ISR)

ISR indicates the status of IRQ7 to IRQ 0 interrupt requests.

Bit	Bit Name	Initial Value	R/W*	Description
7	IRQ7F	0	R/W	[Setting condition] • When the interrupt source selected by the ISCR registers occurs [Clearing conditions] • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed while low-level detection is set and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed while detection of falling edge, rising edge, or both edges is set • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
6	IRQ6F	0	R/W	
5	IRQ5F	0	R/W	
4	IRQ4F	0	R/W	
3	IRQ3F	0	R/W	
2	IRQ2F	0	R/W	
1	IRQ1F	0	R/W	
0	IRQ0F	0	R/W	

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

5.4 Interrupt Sources

5.4.1 External Interrupts

There are 9 external interrupts: NMI and IRQ7 to IRQ0. These interrupts can be used to restore this LSI from software standby mode.

NMI Interrupt: NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

IRQ7 to IRQ0 Interrupts: IRQ7 to IRQ0 interrupts are requested by an input signal at the $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ pins. IRQ7 to IRQ0 interrupts have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at the $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ pins.
- Enabling or disabling of IRQ7 to IRQ0 interrupt requests can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of IRQ7 to IRQ0 interrupt requests is indicated in ISR. ISR flags can be cleared to 0 by software.

A block diagram of IRQ7 to IRQ0 interrupts is shown in figure 5.2.

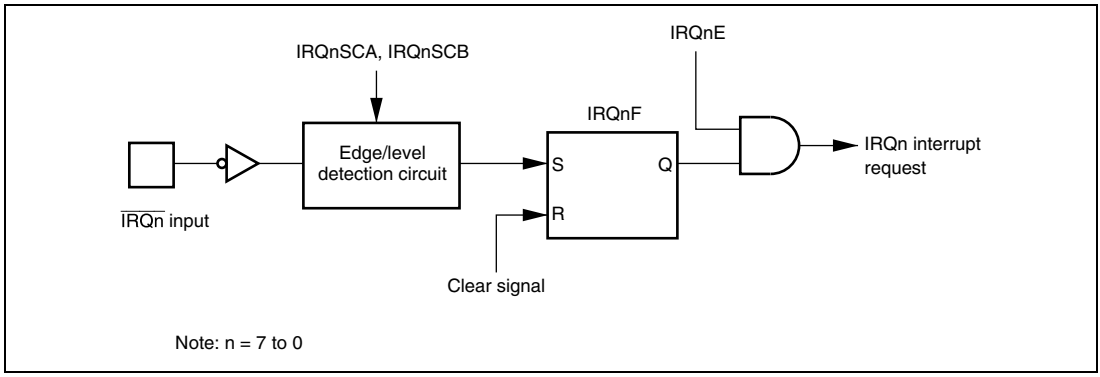


Figure 5.2 Block Diagram of IRQ7 to IRQ0 Interrupts

The set timing for IRQ7F to IRQ0F is shown in figure 5.3.

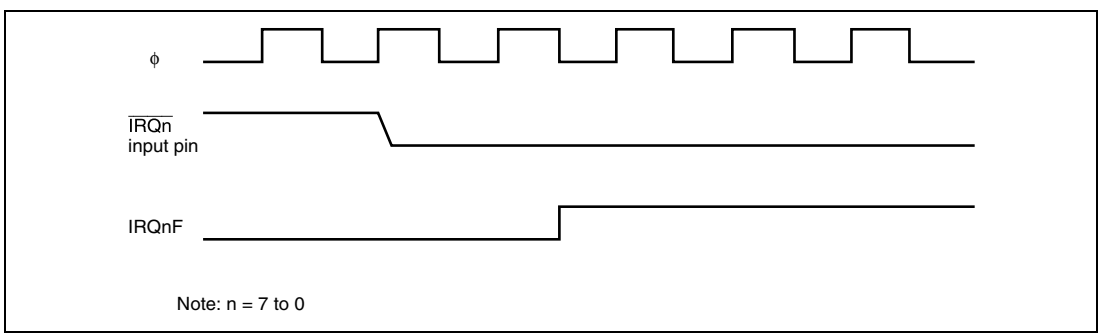


Figure 5.3 Set Timing for IRQ7F to IRQ0F

The detection of IRQ7 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0 to use the pin as an I/O pin for another function. The IRQ7F to IRQ0F interrupt request flags can be set to 1 when the setting condition is satisfied, regardless of IER settings. Accordingly, refer to only necessary flags.

5.4.2 Internal Interrupts

For each on-chip peripheral module, there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If both of these are set to 1 for a particular interrupt source, an interrupt request is sent to the interrupt controller.

5.4.3 Interrupt Exception Handling Vector Table

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Origin of Interrupt Source	Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
External pin	NMI	7	H'001C		High
	IRQ0	16	H'0040	IPRA6 to IPRA4	
	IRQ1	17	H'0044	IPRA2 to IPRA0	
	IRQ2	18	H'0048	IPRB6 to IPRB4	
	IRQ3	19	H'004C		
	IRQ4	20	H'0050	IPRB2 to IPRB0	
	IRQ5	21	H'0054		
	IRQ6	22	H'0058	IPRC6 to IPRC4	
	IRQ7	23	H'005C		
DTC	SWDTEND (completion of software initiation data transfer)	24	H'0060	IPRC2 to IPRC0	
Watchdog timer 0	WOVI0 (interval timer 0)	25	H'0064	IPRD6 to IPRD4	
PC break	PC break	27	H'006C	IPRE6 to IPRE4	
A/D	ADI (completion of A/D conversion)	28	H'0070	IPRE2 to IPRE0	
Watchdog timer 1	WOVI1 (interval timer 1)	29	H'0074		
—	Reserved	30 31	H'0078 H'007C		
TPU channel 0	TGI0A (TGR0A input capture/compare-match)	32	H'0080	IPRF6 to IPRF4	
	TGI0B (TGR0B input capture/compare-match)	33	H'0084		
	TGI0C (TGR0C input capture/compare-match)	34	H'0088		
	TGI0D (TGR0D input capture/compare- match)	35	H'008C		
	TCI0V (overflow 0)	36	H'0090		
—	Reserved	37 38 39	H'0094 H'0098 H'009C		Low

Origin of Interrupt Source	Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
TPU channel 1	TGI1A (TGR1A input capture/compare-match)	40	H'00A0	IPRF2 to IPRF0	High
	TGI1B (TGR1B input capture/compare-match)	41	H'00A4		
	TCI1V (overflow 1)	42	H'00A8		
	TCI1U (underflow 1)	43	H'00AC		
TPU channel 2	TGI2A (TGR2A input capture/compare-match)	44	H'00B0	IPRG6 to IPRG4	
	TGI2B (TGR2B input capture/compare-match)	45	H'00B4		
	TCI2V (overflow 2)	46	H'00B8		
	TCI2U (underflow 2)	47	H'00BC		
TPU channel 3	TGI3A (TGR3A input capture/compare-match)	48	H'00C0	IPRG2 to IPRG0	
	TGI3B (TGR3B input capture/compare-match)	49	H'00C4		
	TGI3C (TGR3C input capture/compare-match)	50	H'00C8		
	TGI3D (TGR3D input capture/compare-match)	51	H'00CC		
	TCI3V (overflow 3)	52	H'00D0		
—	Reserved	53 54 55	H'00D4 H'00D8 H'00DC		
TPU channel 4	TGI4A (TGR4A input capture/compare-match)	56	H'00E0	IPRH6 to IPRH4	
	TGI4B (TGR4B input capture/compare-match)	57	H'00E4		
	TCI4V (overflow 4)	58	H'00E8		
	TCI4U (underflow 4)	59	H'00EC		
TPU channel 5	TGI5A (TGR5A input capture/compare-match)	60	H'00F0	IPRH2 to IPRH0	
	TGI5B (TGR5B input capture/compare-match)	61	H'00F4		
	TCI5V (overflow 5)	62	H'00F8		
	TCI5U (underflow 5)	63	H'00FC		Low

Origin of Interrupt Source	Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
8-bit timer channel 0	CMIA0 (compare-match A0)	64	H'0100	IPRI6 to IPRI4	High
	CMIB0 (compare-match B0)	65	H'0104		
	OVI0 (overflow 0)	66	H'0108		
—	Reserved	67	H'010C		
8-bit timer channel 1	CMIA1 (compare-match A1)	68	H'0110	IPRI2 to IPRI0	
	CMIB1 (compare-match B1)	69	H'0114		
	OVI1 (overflow 1)	70	H'0118		
—	Reserved	71	H'011C		
SCI channel 0	ERI0 (receive error 0)	80	H'0140	IPRJ2 to IPRJ0	
	RXI0 (receive completion 0)	81	H'0144		
	TXI0 (transmit data empty 0)	82	H'0148		
	TEI0 (transmit end 0)	83	H'014C		
SCI channel 1	ERI1 (receive error 1)	84	H'0150	IPRK6 to IPRK4	
	RXI1 (receive completion 1)	85	H'0154		
	TXI1 (transmit data empty 1)	86	H'0158		
	TEI1 (transmit end 1)	87	H'015C		
SCI channel 2	ERI2 (receive error 2)	88	H'0160	IPRK2 to IPRK0	
	RXI2 (receive completion 2)	89	H'0164		
	TXI2 (transmit data empty 2)	90	H'0168		
	TEI2 (transmit end 2)	91	H'016C		
8-bit timer channel 2	CMIA2 (compare-match A2)	92	H'0170	IPRL6 to IPRL4	
	CMIB2 (compare-match B2)	93	H'0174		
	OVI2 (overflow 2)	94	H'0178		
—	Reserved	95	H'017C		
8-bit timer channel 3	CMIA3 (compare-match A3)	96	H'0180		
	CMIB3 (compare-match B3)	97	H'0184		
	OVI3 (overflow 3)	98	H'0188		
—	Reserved	99	H'018C		
IEB (H8S/2552 Series only)	IERSI (reception status)	104	H'01A0	IPRM6 to IPRM4	
	IERxI (RxRDY)	105	H'01A4		
	IETxI (TxRDY)	106	H'01A8		
	IETSI (transmission status)	107	H'01AC		Low

Origin of Interrupt Source	Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
HCAN (H8S/2556 Series only)	ERS0, OVR0, RM1, SLE0	108	H'01B0	IPRM2 to IPRM0	High
	RM0	109	H'01B4		
IIC2 channel 0	IICI0 (1-byte transmission/ reception completion)	110	H'01B8		
IIC2 channel 1	IICI1 (1-byte transmission/ reception completion)	111	H'01BC		
SCI channel 3	ERI3 (receive error 3)	120	H'01E0	IPRO6 to IPRO4	
	RXI3 (receive completion 3)	121	H'01E4		
	TXI3 (transmit data empty 3)	122	H'01E8		
	TEI3 (transmit end 3)	123	H'01EC		
SCI channel 4	ERI4 (receive error 4)	124	H'01F0	IPRO2 to IPRO0	
	RXI4 (receive completion 4)	125	H'01F4		
	TXI4 (transmit data empty 4)	126	H'01F8		
	TEI4 (transmit end 4)	127	H'01FC		Low

Note: * Indicates lower 16 bits of the start address.

5.5 Operation

5.5.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in this LSI differ depending on the interrupt control mode.

NMI interrupts are accepted at all times except in the reset state and the hardware standby state. In the case of IRQ interrupts and on-chip peripheral module interrupts, an enable bit is provided for each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. Interrupt sources for which the enable bits are set to 1 are controlled by the interrupt controller.

Table 5.3 shows the interrupt control modes.

The interrupt controller performs interrupt control according to the interrupt control mode set by the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state indicated by the I bit in the CPU's CCR, and bits I2 to I0 in EXR.

Table 5.3 Interrupt Control Modes

Interrupt Control Mode	SYSCR INTM1	SYSCR INTM0	Priority Setting Register	Interrupt Mask Bits	Description
0	0	0	—	I	Interrupt mask control is performed by the I bit.
—		1	—	—	Setting prohibited
2	1	0	IPR	I2 to I0	8-level interrupt mask control is performed by bits I2 to I0. 8 priority levels can be set with IPR.
—		1	—	—	Setting prohibited

Figures 5.4 shows a block diagram of the priority decision circuit.

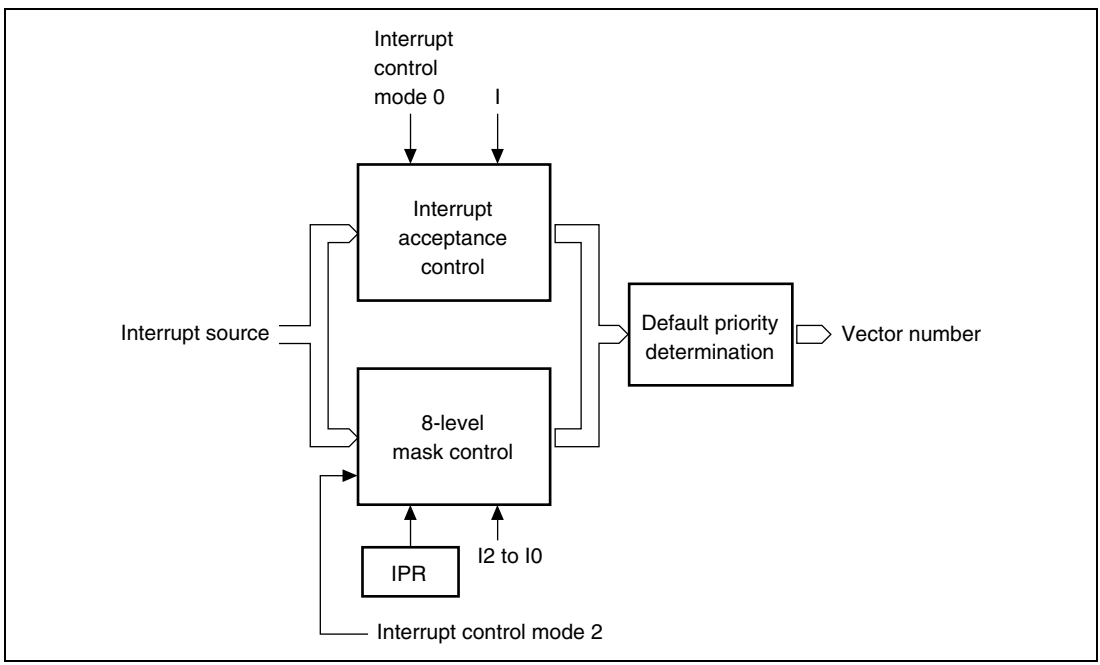


Figure 5.4 Block Diagram of Interrupt Control Operation

Interrupt Acceptance Control: In interrupt control mode 0, interrupt acceptance is controlled by the I bit in CCR.

Table 5.4 shows the interrupts selected in each interrupt control mode.

Table 5.4 Interrupts Selected in Each Interrupt Control Mode (1)

	Interrupt Mask Bits	
Interrupt Control Mode	**I**	**Selected Interrupts**
0	0	All interrupts
	1	NMI interrupt
2	X	All interrupts

Legend
X: Don't care

8-Level Control: In interrupt control mode 2, 8-level mask level determination is performed for the selected interrupts in interrupt acceptance control according to the interrupt priority level (IPR).

The interrupt source selected is the interrupt with the highest priority level, and whose priority level set in IPR is higher than the mask level.

Table 5.5 Interrupts Selected in Each Interrupt Control Mode (2)

Interrupt Control Mode	Selected Interrupts
0	All interrupts
2	Highest priority-level (IPR) interrupt whose priority level is greater than the mask level (IPR > I2 to I0)

Default Priority Determination: When an interrupt is selected by 8-level control, its priority is determined and a vector number is generated.

If the same value is set for IPR, acceptance of multiple interrupts is enabled, and so only the interrupt source with the highest priority according to the preset default priorities is selected and has a vector number generated. Interrupt sources with a lower priority than the accepted interrupt source are held pending.

Table 5.6 shows operations and control signal functions in each interrupt control mode.

Table 5.6 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting		Interrupt Acceptance Control		8-Level Control			Default Priority Determination	T (Trace)
	INTM1	INTM0		I		I2 to I0	IPR		
0	0	0	O	IM	X	—	—*2	O	—
2	1	0	X	—*1	O	IM	PR	O	T

Legend

O: Interrupt operation control performed

X: No operation. (All interrupts enabled)

IM: Used as interrupt mask bit

PR: Sets priority

—: Not used

Notes: 1. Set to 1 when an interrupt is accepted.

2. Keep the initial setting.

5.5.2 Interrupt Control Mode 0

Enabling and disabling of IRQ interrupts and on-chip peripheral module interrupts can be set by means of the I bit in the CPU's CCR. Interrupts are enabled when the I bit is cleared to 0, and disabled when set to 1.

Figure 5.5 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. The I bit is referred to. If the I bit is cleared to 0, an interrupt request is accepted. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.
3. Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address.

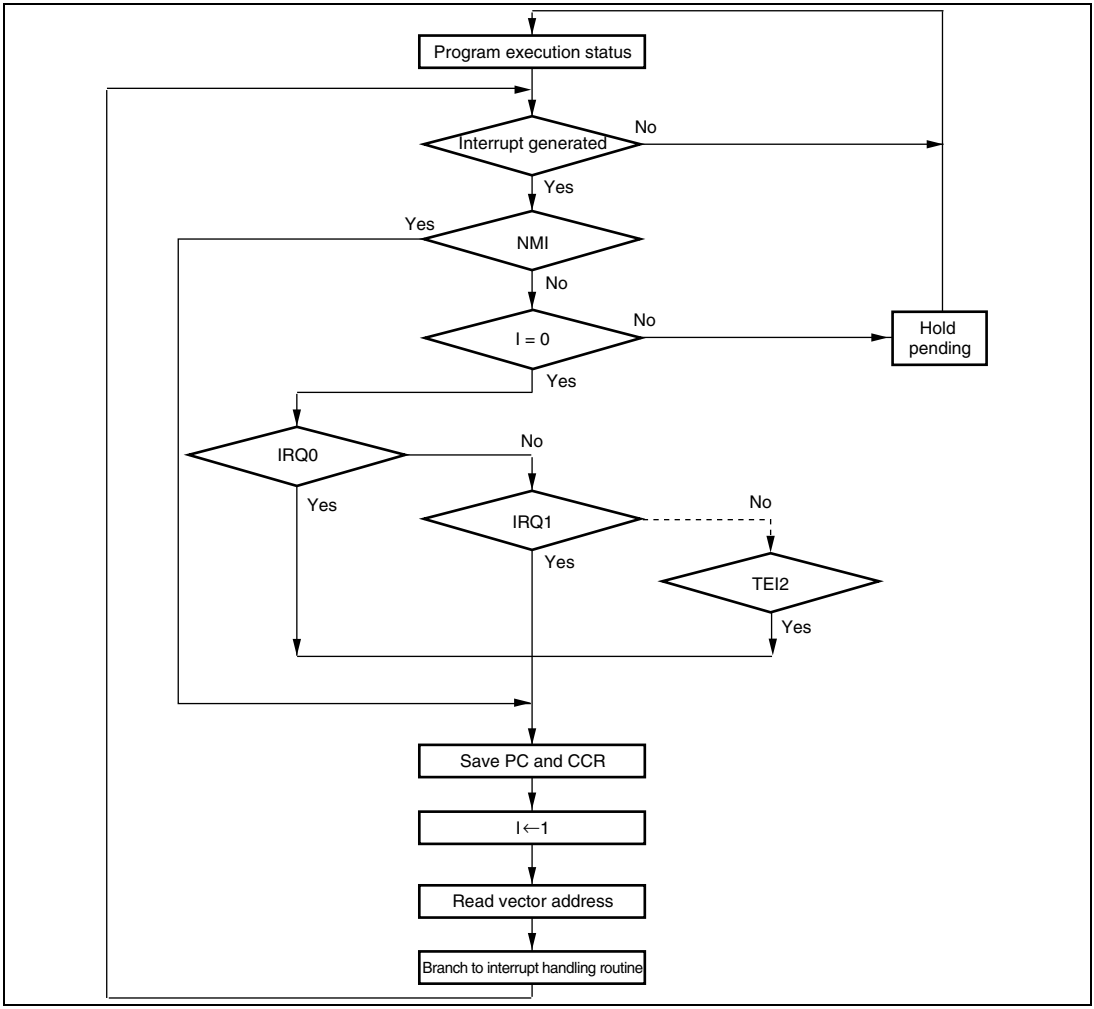


Figure 5.5 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0

5.5.3 Interrupt Control Mode 2

Eight-level masking is implemented for IRQ interrupts and on-chip peripheral module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Figure 5.6 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.2 is selected.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.
   If the accepted interrupt is NMI, the interrupt mask level is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address.

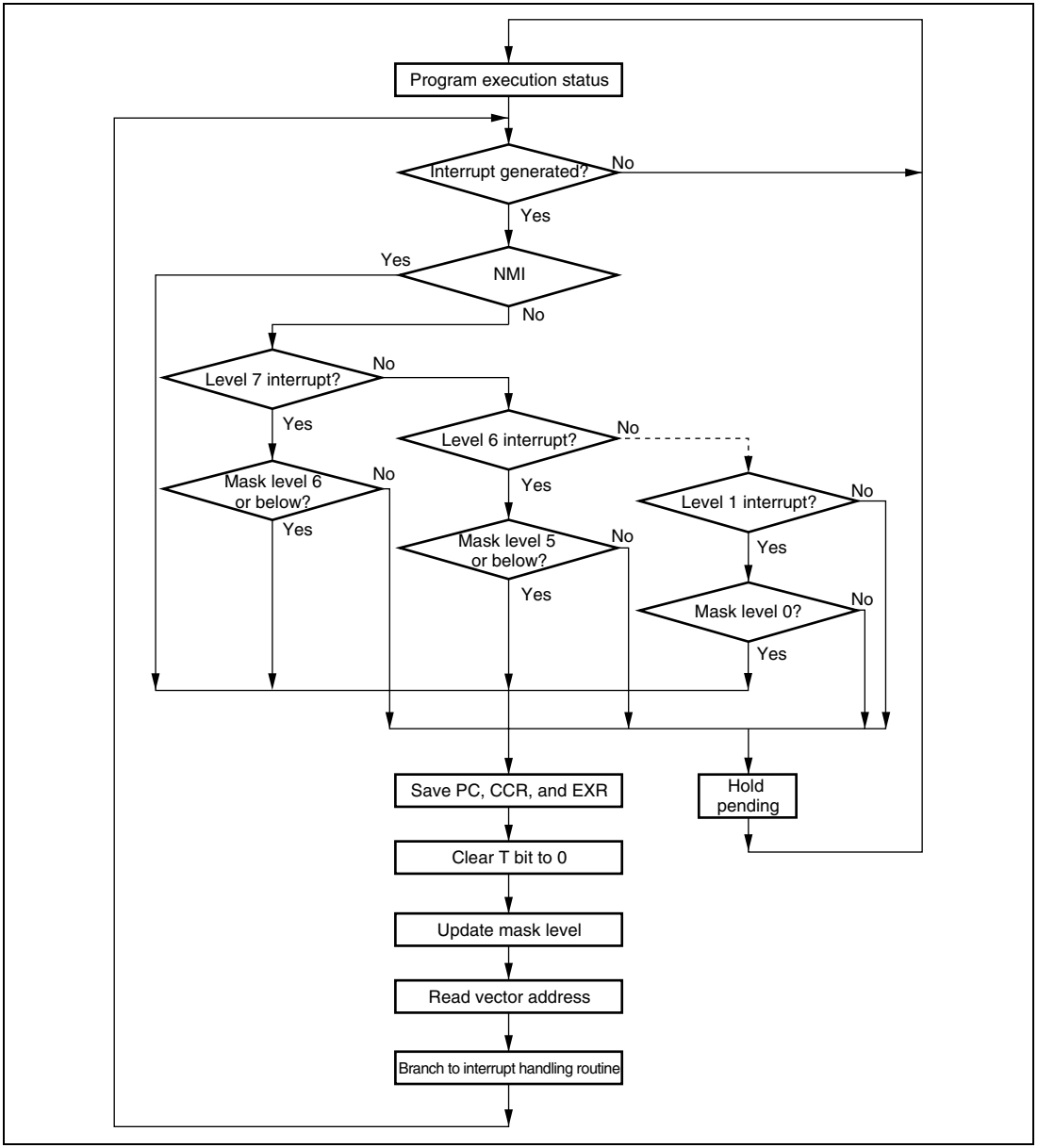


Figure 5.6 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 2

5.5.4 Interrupt Exception Handling Sequence

Figure 5.7 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

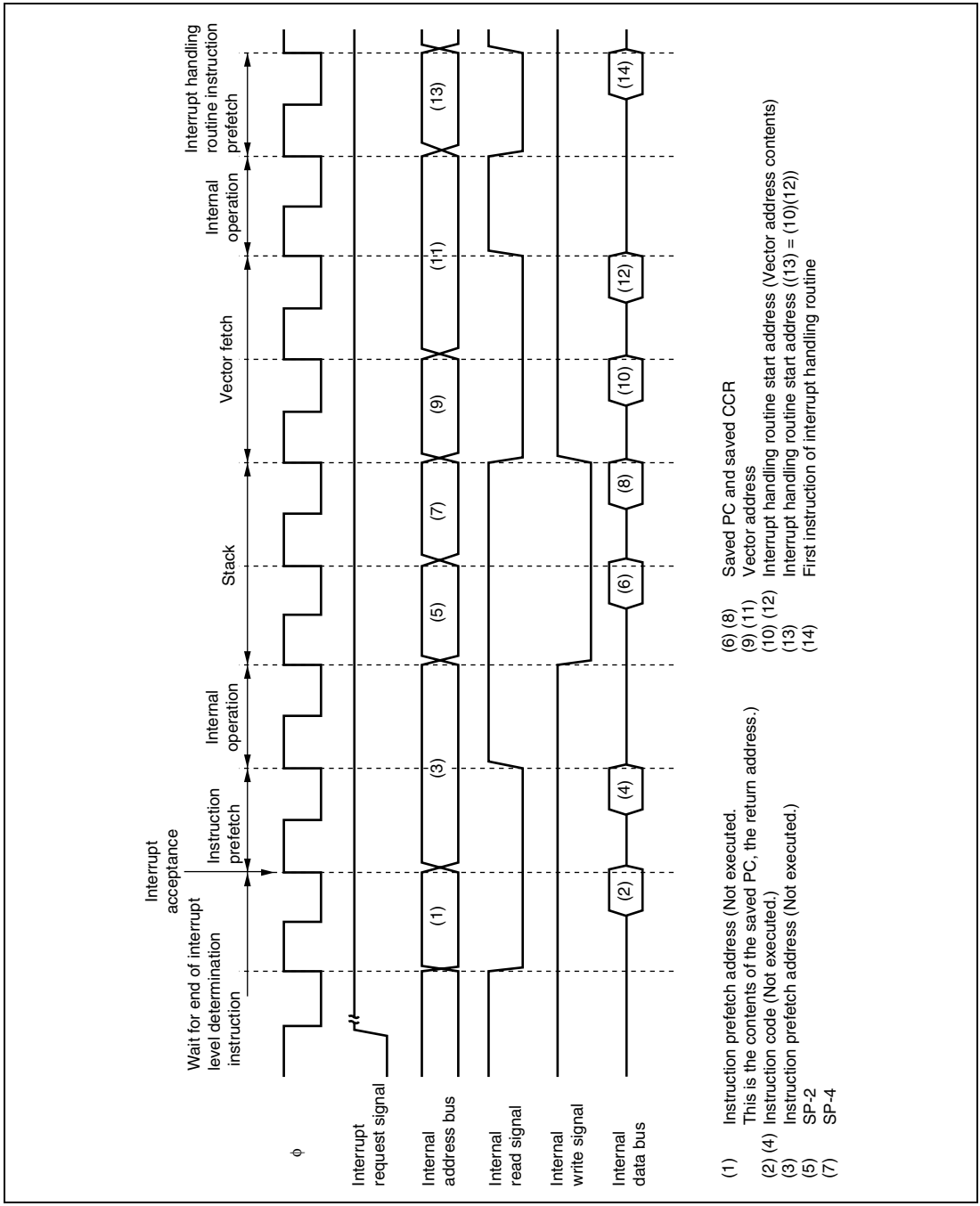


Figure 5.7 Interrupt Exception Handling

5.5.5 Interrupt Response Times

This LSI is capable of fast word transfer to on-chip memory, has the program area in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5.7 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5.7 are explained in table 5.8.

Table 5.7 Interrupt Response Times (States)

		Normal Mode*[5]		Advanced Mode	
No.	**Execution Status**	**INTM1 = 0**	**INTM1 = 1**	**INTM1 = 0**	**INTM1 = 1**
1	Interrupt priority determination*[1]	3	3	3	3
2	Number of wait states until executing instruction ends*[2]	1 to 19 + 2·S_I	1 to 19 + 2·S_I	1 to 19 + 2·S_I	1 to 19 + 2·S_I
3	PC, CCR, EXR stack save	2·S_K	3·S_K	2·S_K	3·S_K
4	Vector fetch	S_I	S_I	2·S_I	2·S_I
5	Instruction fetch*[3]	2·S_I	2·S_I	2·S_I	2·S_I
6	Internal processing*[4]	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13 to 33

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
4. Internal processing after interrupt acceptance and internal processing after vector fetch.
5. Not available in this LSI.

Table 5.8 Number of States in Interrupt Handling Routine Execution Status

		Object of Access				
			External Device			
			8-Bit Bus		16-Bit Bus	
Symbol		**On-Chip Memory**	**2-State Access**	**3-State Access**	**2-State Access**	**3-State Access**
Instruction fetch	S_I	1	4	6 + 2m	2	3 + m
Branch address read	S_J					
Stack manipulation	S_K					

Legend
m: Number of wait states in an external device access.

5.5.6 DTC Activation by Interrupt

The DTC can be activated by an interrupt. In this case, the following selections can be made.

1. Interrupt request to CPU
2. Activation request to DTC
3. Multiple selection of 1 and 2 above.

For details on interrupt request, which enables DTC activation, refer to section 8, Data Transfer Controller (DTC). Figure 5.8 shows a block diagram of DTC and interrupt controller.

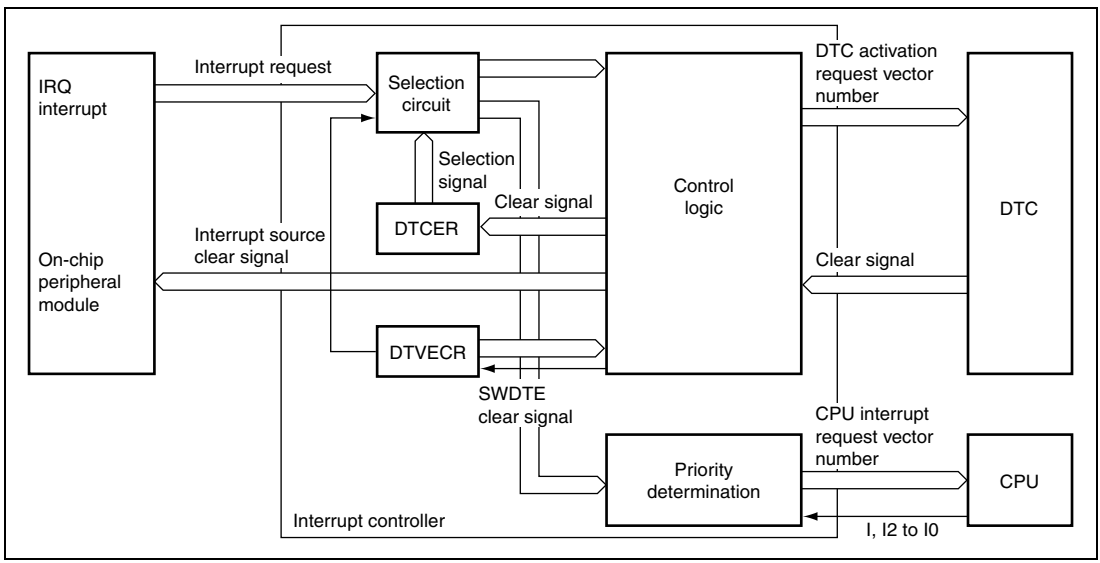


Figure 5.8 DTC and Interrupt Controller

Interrupt controller of DTC control has the following three main functions.

Interrupt source selection: For interruption source, select DTC activation request or CPU interruption request by the DTCE bits in DTCERA to DTCERG, and DTCERI of the DTC. After DTC data transfer, the DTCE bit is cleared to 0, and an interrupt request to the CPU can be made by the setting of the DISEL bit in MRB of the DTC. When DTC performs data transfer for prescribed number of times and transfer counter becomes 0, the DTCE bit should be cleared to 0 and an interrupt request to the CPU is made after DTC data transfer.

Priority determination: DTC activation source is selected according to priority of default setting. Mask level and priority level do not affect the selection. For details, refer to section 8.4, Location of Register Information and DTC Vector Table.

Operation order: When the same interrupts are selected as DTC activation source and CPU interruption source, DTC data is transferred, and then CPU interrupt exception processing is made.

Table 5.9 shows interrupt source selection and interrupt source clear control by the setting of the DTCE bit in DTCERA to DTCERG, and DTCERI of the DTC and the setting of the DISEL bit in MRB of the DTC.

Table 5.9 Interrupt Source Selection and Clear Control

Settings		Interrupt Source Selection and Clear Control	
DTC			
DTCE	DESEL	DTC	CPU
0	*	X	#
1	0	#	X
	1	O	#

Legend

#: Corresponding interrupt is used. Interrupt source is cleared.
(The CPU should clear the source flag in the interrupt processing routine.)

O: Corresponding interrupt is used. Interrupt source is not cleared.

X: Corresponding interrupt cannot be used.

*: Don't care

Usage note: Interrupt sources of the SCI and A/D converter are cleared when the DTC reads or writes prescribed register, and they do not depend on the DTCE or DISEL bit.

5.6 Usage Notes

5.6.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction.

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, and if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared to 0.

The above contention will not occur, if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

Figure 5.9 shows an example in which the CMIEA bit in the TCR register of the 8-bit timer is cleared to 0.

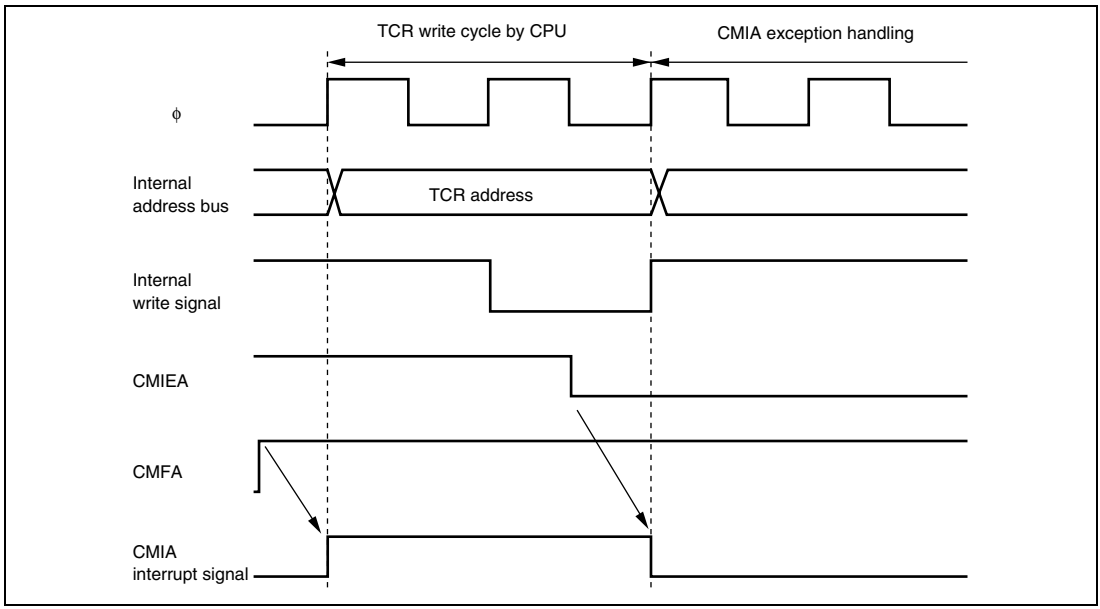


Figure 5.9 Contention between Interrupt Generation and Disabling

5.6.2 Instructions that Disable Interrupts

The instructions that disable interrupt requests directly after execution are LDC, ANDC, ORC, and XORC. After any of these instructions are executed, all interrupts including NMI are disabled and the next instruction is always executed.

When the I bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

5.6.3 When Interrupts Are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.6.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the transfer is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

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

Section 6 PC Break Controller (PBC)

The PC break controller (PBC) provides functions that simplify program debugging. Using these functions, it is easy to create a self-monitoring debugger, enabling programs to be debugged with the chip alone, without using an in-circuit emulator. A block diagram of the PC break controller is shown in figure 6.1.

6.1 Features

- Two break channels (A and B)
- 24-bit break address
  - Bit masking possible
- Four types of break compare conditions
  - Instruction fetch
  - Data read
  - Data write
  - Data read/write
- Bus master
  - Either CPU or CPU/DTC can be selected
- The timing of PC break exception handling after the occurrence of a break condition is as follows:
  - Immediately before execution of the instruction fetched at the set address (instruction fetch)
  - Immediately after execution of the instruction that accesses data at the set address (data access)
- Module stop mode can be set

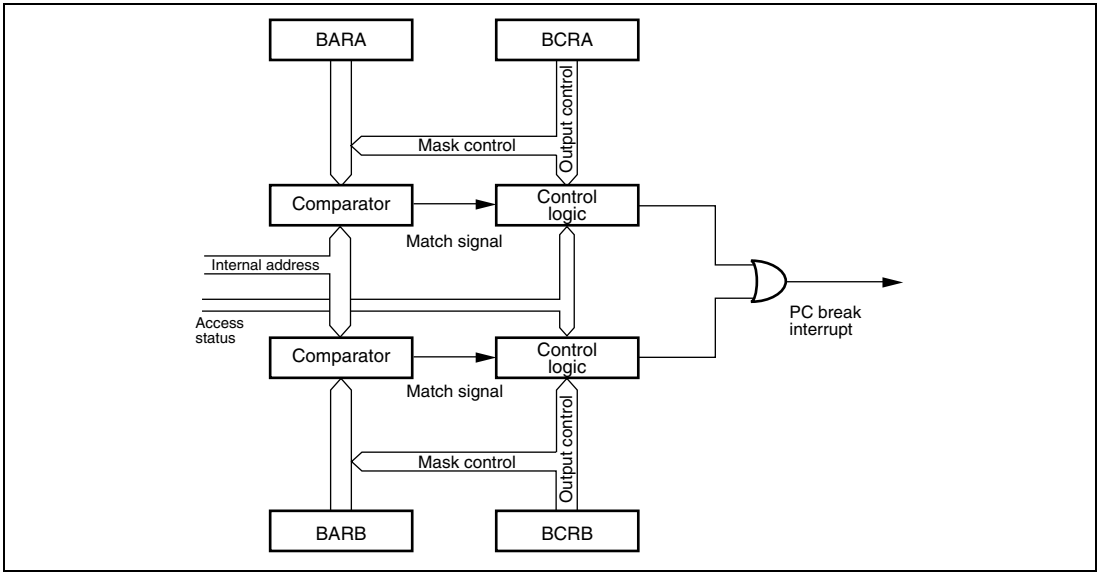


Figure 6.1 Block Diagram of PC Break Controller

6.2 Register Descriptions

The PC break controller has the following registers.

- Break address register A (BARA)
- Break address register B (BARB)
- Break control register A (BCRA)
- Break control register B (BCRB)

6.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register that specifies the channel A break address.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	Undefined	—	Reserved These bits are read as an undefined value and cannot be modified.
23 to 0	BAA23 to BAA0	H'000000	R/W	These bits set the PC break address of channel A.

6.2.2 Break Address Register B (BARB)

BARB is the channel B break address register. The bit configuration is the same as for BARA.

6.2.3 Break Control Register A (BCRA)

BCRA controls channel A PC breaks.

Bit	Bit Name	Initial Value	R/W	Description
7	CMFA	0	R/(W)*[1]	Condition Match Flag A [Setting condition] When a condition set for channel A is satisfied [Clearing condition] When 0 is written to CMFA after reading*[2] CMFA = 1
6	CDA	0	R/W	CPU Cycle/DTC Cycle Select A Selects the channel A break condition bus master. 0: CPU 1: CPU or DTC
5	BAMRA2	0	R/W	Break Address Mask Register A2 to A0
4	BAMRA1	0	R/W	These bits specify which bits of the break address set in BARA are to be unmasked.
3	BAMRA0	0	R/W	000: BAA23 to BAA0 (All bits are unmasked) 001: BAA23 to BAA1 (Lowest bit is masked) 010: BAA23 to BAA2 (Lower 2 bits are masked) 011: BAA23 to BAA3 (Lower 3 bits are masked) 100: BAA23 to BAA4 (Lower 4 bits are masked) 101: BAA23 to BAA8 (Lower 8 bits are masked) 110: BAA23 to BAA12 (Lower 12 bits are masked) 111: BAA23 to BAA16 (Lower 16 bits are masked)
2	CSELA1	0	R/W	Break Condition Select
1	CSELA0	0	R/W	These bits select the break condition of channel A. 00: Instruction fetch is used as the break condition. 01: Data read cycle is used as the break condition. 10: Data write cycle is used as the break condition. 11: Data read/write cycle is used as the break condition.

Bit	Bit Name	Initial Value	R/W	Description
0	BIEA	0	R/W	Break Interrupt Enable When this bit is set to 1, the PC break interrupt request of channel A is enabled.

Notes: 1. Only 0 can be written to this bit to clear the flag.
2. Read the state wherein CMFA = 1 twice or more, when the CMFA is polled after inhibiting the PC break interruption.

6.2.4 Break Control Register B (BCRB)

BCRB is the channel B break control register. The bit configuration is the same as for BCRA.

6.3 Operation

The operation flow from break condition setting to PC break interrupt exception handling is shown in section 6.3.1, PC Break Interrupt Due to Instruction Fetch, and section 6.3.2, PC Break Interrupt Due to Data Access, taking the example of channel A.

6.3.1 PC Break Interrupt Due to Instruction Fetch

1. Set the break address in BARA.
   For a PC break caused by an instruction fetch, set the address of the first instruction byte as the break address.
2. Set the break conditions in BCR.
   Set bit 6 (CDA) to 0 to select the CPU because the bus master must be the CPU for a PC break caused by an instruction fetch. Set the address bits to be masked to bits 5 to 3 (BAMRA2 to BAMRA0). Set bits 2 and 1 (CSELA1 and CSELA0) to 00 to specify an instruction fetch as the break condition. Set bit 0 (BIEA) to 1 to enable break interrupts.
3. When the instruction at the set address is fetched, a PC break request is generated immediately before execution of the fetched instruction, and the condition match flag (CMFA) is set.
4. After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.2 PC Break Interrupt Due to Data Access

1. Set the break address in BARA.
   For a PC break caused by a data access, set the target ROM, RAM, I/O, or external address space address as the break address. Stack operations and branch address reads are included in data accesses.

2. Set the break conditions in BCRA.

   Select the bus master with bit 6 (CDA). Set the address bits to be masked to bits 5 to 3 (BAMRA2 to BAMRA0). Set bits 2 and 1 (CSELA1 and CSELA0) to 01, 10, or 11 to specify data access as the break condition. Set bit 0 (BIEA) to 1 to enable break interrupts.
3. After execution of the instruction that performs a data access on the set address, a PC break request is generated and the condition match flag (CMFA) is set.
4. After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.3 Notes on PC Break Interrupt Handling

- When a PC break interrupt is generated at the transfer address of an EEPMOV.B instruction

  PC break exception handling is executed after all data transfers have been completed and the EEPMOV.B instruction has ended.
- When a PC break interrupt is generated at a DTC transfer address

  PC break exception handling is executed after the DTC has completed the specified number of data transfers, or after data for which the DISEL bit is set to 1 has been transferred.

6.3.4 Operation in Transitions to Power-Down Modes

The operation when a PC break interrupt is set for an instruction fetch at the address after a SLEEP instruction is shown below.

- When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to sleep mode

  After execution of the SLEEP instruction, a transition is not made to sleep mode, and PC break exception handling is executed. After execution of PC break exception handling, the instruction at the address after the SLEEP instruction is executed (figure 6.2 (A)).
- When the SLEEP instruction causes a transition to software standby mode or watch mode

  After execution of the SLEEP instruction, a transition is made to the respective mode, and PC break exception handling is not executed. However, the CMFA or CMFB flag is set (figure 6.2 (B)).

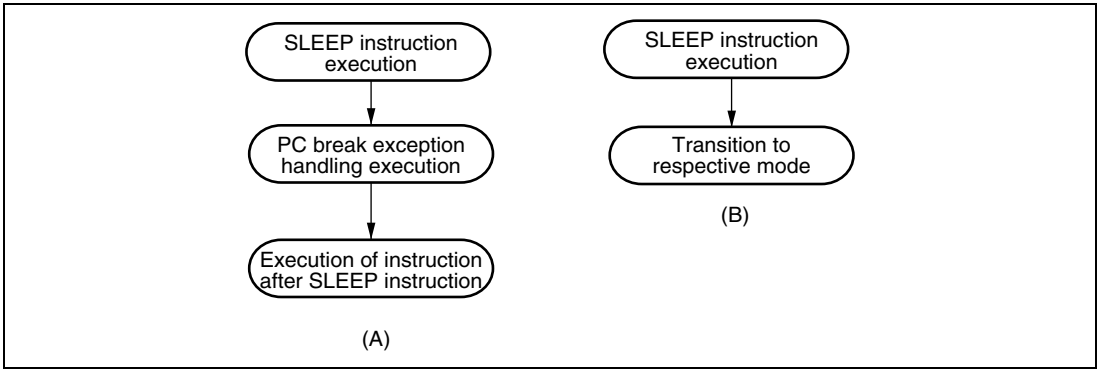


Figure 6.2 Operation in Power-Down Mode Transitions

6.3.5 When Instruction Execution Is Delayed by One State

While the break interrupt enable bit is set to 1, instruction execution in the following cases is one state later than usual.

- For 1-word branch instructions (Bcc d:8, BSR, JSR, JMP, TRAPA, RTE, and RTS) in on-chip ROM or RAM
- When break interruption by instruction fetch is set, the set address indicates on-chip ROM or RAM space, and that address is used for data access
- When break interruption by instruction fetch is set, if the instruction to be executed immediately before the set instruction has one of the addressing modes shown below, and that address indicates on-chip ROM or RAM

  Addressing modes: @ERn, @(d:16,ERn), @(d:32,ERn), @-ERn/ERn+, @aa:8, @aa:24, @aa:32, @(d:8,PC), @(d:16,PC), @@aa:8
- When break interruption by instruction fetch is set, if the instruction to be executed immediately before the set instruction is NOP or SLEEP, or has #xx, Rn as its addressing mode, and that instruction is located in on-chip ROM or RAM

6.4 Usage Notes

6.4.1 Module Stop Mode Setting

PBC operation can be disabled or enabled using the module stop control register. The initial setting is for PBC operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

6.4.2 PC Break Interrupts

The PC break interrupt is shared by channels A and B. The channel from which the request was issued must be determined by the interrupt handler.

6.4.3 CMFA and CMFB

The CMFA and CMFB flags are not automatically cleared to 0, so 0 must be written to CMFA or CMFB after first reading the flag while it is set to 1. If the flag is left set to 1, another interrupt will be requested after interrupt handling ends.

6.4.4 PC Break Interrupt when DTC is Bus Master

A PC break interrupt generated when the DTC is the bus master is accepted after the bus mastership has been transferred to the CPU by the bus controller.

6.4.5 PC Break Set for Instruction Fetch at Address Following BSR, JSR, JMP, TRAPA, RTE, or RTS Instruction

Even if the instruction at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction is fetched, it is not executed, and so a PC break interrupt is not generated by the instruction fetch at the next address.

6.4.6 I Bit Set by LDC, ANDC, ORC, or XORC Instruction

When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break interrupt becomes valid two states after the end of the executing instruction. If a PC break interrupt is set for the instruction following one of these instructions, since interrupts, including NMI, are disabled for a 3-state period in the case of LDC, ANDC, ORC, and XOR, the next instruction is always executed. For details, see section 5, Interrupt Controller.

6.4.7 PC Break Set for Instruction Fetch at Address Following Bcc Instruction

A PC break interrupt is generated if the instruction at the next address is executed in accordance with the branch condition, and is not generated if the instruction at the next address is not executed.

6.4.8 PC Break Set for Instruction Fetch at Branch Destination Address of Bcc Instruction

A PC break interrupt is generated if the instruction at the branch destination is executed in accordance with the branch condition, and is not generated if the instruction at the branch destination is not executed.

Section 7 Bus Controller

This LSI has a built-in bus controller (BSC) that manages the external address space divided into eight areas. As the bus controller has a bus mastership arbitration function, it controls the operation of the CPU (the internal bus master) and the data transfer controller (DTC).

7.1 Features

- Manages external address space in area units
  - Manages the external address space as 8 areas in 2-Mbyte units
  - Bus specifications can be set independently for each area
  - Burst ROM interface can be set
- Basic bus interface
  - H8S/2552 Series, H8S/2506 Series: Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for areas 0 to 7.
  - H8S/2556 Series: Chip select signals ($\overline{CS0}$, $\overline{CS3}$ to $\overline{CS7}$) can be output for areas 0 and 3 to 7.
  - 8-bit access or 16-bit access can be selected for each area
  - 2-state access or 3-state access can be selected for each area
  - Program wait states can be inserted for each area
- Burst ROM interface
  - Burst ROM interface can be selected for area 0
  - One or two states can be selected for the burst cycle
- Idle cycle insertion
  - Idle cycle can be inserted between consecutive read accesses to different external areas
  - Idle cycle can be inserted before a write access to an external area immediately after a read access to an external area
- Bus mastership arbitration
  - The on-chip bus arbiter arbitrates the bus mastership among CPU and DTC.
- Other features
  - External bus mastership release function

Figure 7.1 shows a block diagram of the bus controller.

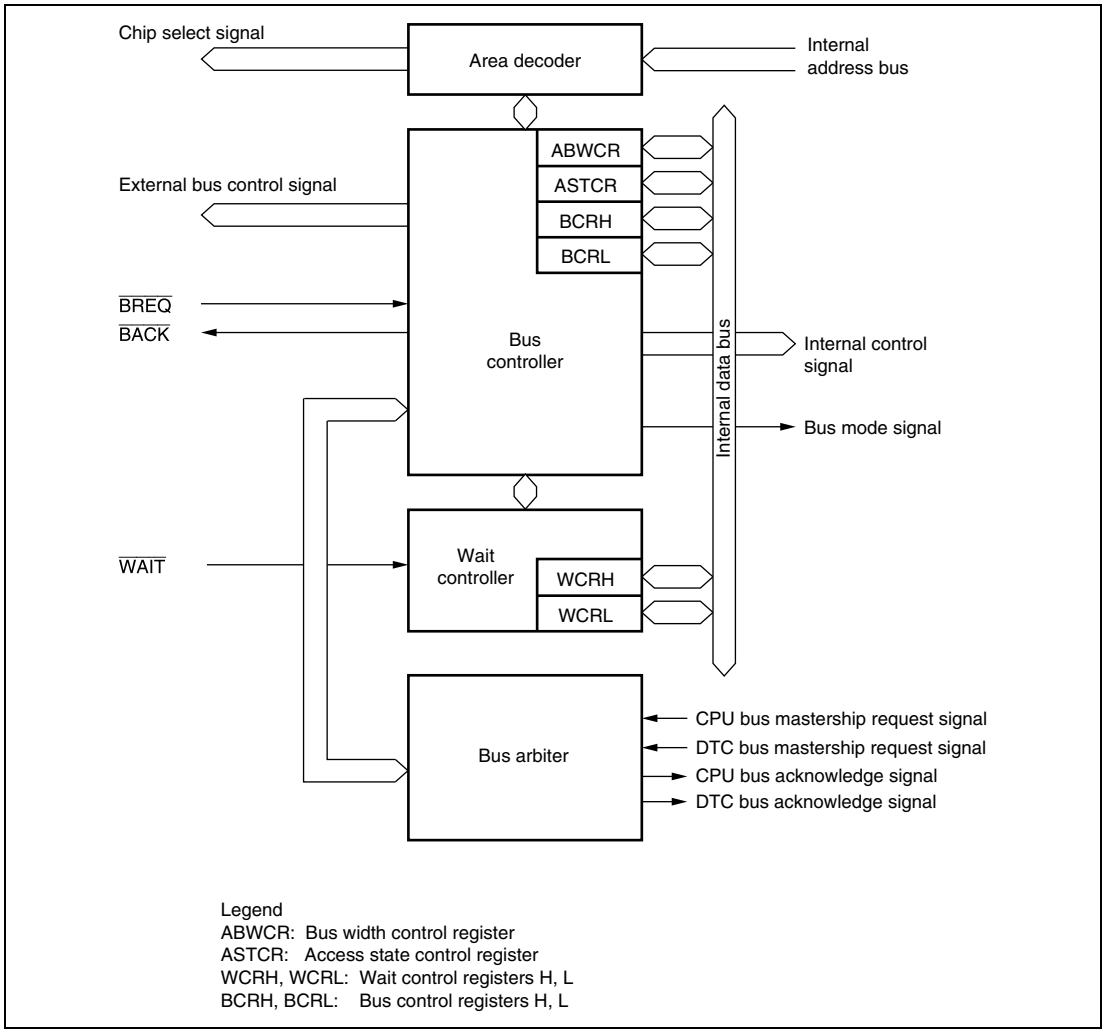


Figure 7.1 Block Diagram of Bus Controller

7.2 Input/Output Pins

Table 7.1 summarizes the pins of the bus controller.

Table 7.1 Pin Configuration

Name	Symbol	I/O	Function
Address strove	$\overline{AS}$	Output	Strobe signal indicating that address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that external address space is being read.
High write	$\overline{HWR}$	Output	Strobe signal indicating that external address space is to be written, and upper half (D15 to D8) of data bus is enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external address space is to be written, and lower half (D7 to D0) of data bus is enabled.
Chip select 0 to 7	$\overline{CS0}$ to $\overline{CS7}$*	Output	Strobe signal indicating that areas 0 to 7 are selected.
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external 3-state access space.
Bus mastership request	$\overline{BREQ}$	Input	Request signal that releases bus to external device.
Bus mastership request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has been released.

Note: * $\overline{CS1}$ and $\overline{CS2}$ are not provided in the H8S/2556 Series.

7.3 Register Descriptions

The bus controller has the following registers.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register H (WCRH)
- Wait control register L (WCRL)
- Bus control register H (BCRH)
- Bus control register L (BCRL)
- Pin function control register (PFCR)

7.3.1 Bus Width Control Register (ABWCR)

ABWCR designates each area as either an 8-bit access space or a 16-bit access space.

ABWCR sets the data bus width for the external memory space. The bus width for on-chip memory and internal I/O registers is fixed regardless of the settings in ABWCR.

Bit	Bit Name	Initial Value	R/W	Description
7	ABW7	1	R/W	Area 7 to 0 Bus Width Control
6	ABW6	1	R/W	These bits select whether the corresponding area is to be designated for 8-bit access or 16-bit access.
5	ABW5	1	R/W	
4	ABW4	1	R/W	0: Area n is designated for 16-bit access
3	ABW3	1	R/W	1: Area n is designated for 8-bit access
2	ABW2	1	R/W	Legend n: 7 to 0
1	ABW1	1	R/W	
0	ABW0	1	R/W	

7.3.2 Access State Control Register (ASTCR)

ASTCR designates each area as either a 2-state access space or a 3-state access space.

ASTCR sets the number of access states for the external memory space. The number of access states for on-chip memory and internal I/O registers is fixed regardless of the settings in ASTCR.

Bit	Bit Name	Initial Value	R/W	Description
7	AST7	1	R/W	Area 7 to 0 Access State Control
6	AST6	1	R/W	These bits select whether the corresponding area is to be designated as a 2-state access space or a 3-state access space. Wait state insertion is enabled or disabled at the same time.
5	AST5	1	R/W	
4	AST4	1	R/W	
3	AST3	1	R/W	0: Area n is designated for 2-state access
2	AST2	1	R/W	Wait state insertion in area n external space is disabled
1	AST1	1	R/W	1: Area n is designated for 3-state access
0	AST0	1	R/W	Wait state insertion in area n external space is enabled
				Legend n: 7 to 0

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

WCRH and WCRL select the number of program wait states for each area.

Program wait states are not inserted in the case of on-chip memory or internal I/O registers.

- WCRH

Bit	Bit Name	Initial Value	R/W	Description
7	W71	1	R/W	Area 7 Wait Control 1 and 0
6	W70	1	R/W	These bits select the number of program wait states when area 7 in external address space is accessed while the AST7 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
5	W61	1	R/W	Area 6 Wait Control 1 and 0
4	W60	1	R/W	These bits select the number of program wait states when area 6 in external address space is accessed while the AST6 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
3	W51	1	R/W	Area 5 Wait Control 1 and 0
2	W50	1	R/W	These bits select the number of program wait states when area 5 in external address space is accessed while the AST5 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
1	W41	1	R/W	Area 4 Wait Control 1 and 0
0	W40	1	R/W	These bits select the number of program wait states when area 4 in external address space is accessed while the AST4 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted

- WCRL

Bit	Bit Name	Initial Value	R/W	Description
7	W31	1	R/W	Area 3 Wait Control 1 and 0
6	W30	1	R/W	These bits select the number of program wait states when area 3 in external address space is accessed while the AST3 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
5	W21	1	R/W	Area 2 Wait Control 1 and 0
4	W20	1	R/W	These bits select the number of program wait states when area 2 in external address space is accessed while the AST2 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
3	W11	1	R/W	Area 1 Wait Control 1 and 0
2	W10	1	R/W	These bits select the number of program wait states when area 1 in external address space is accessed while the AST1 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
1	W01	1	R/W	Area 0 Wait Control 1 and 0
0	W00	1	R/W	These bits select the number of program wait states when area 0 in external address space is accessed while the AST0 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted

7.3.4 Bus Control Register H (BCRH)

BCRH selects enabling or disabling of idle cycle insertion, and the memory interface for area 0.

Bit	Bit Name	Initial Value	R/W	Description
7	ICIS1	1	R/W	Idle Cycle Insertion 1 Selects whether or not one idle cycle state is to be inserted between bus cycles when consecutive external read cycles are performed in different areas. 0: Idle cycle is not inserted in case of consecutive external read cycles in different areas 1: Idle cycle is inserted in case of consecutive external read cycles in different areas
6	ICIS0	1	R/W	Idle Cycle Insertion 0 Selects whether or not one idle cycle state is to be inserted between bus cycles when consecutive external read and write cycles are performed. 0: Idle cycle is not inserted in case of consecutive external read and write cycles 1: Idle cycle is inserted in case of consecutive external read and write cycles
5	BRSTRM	0	R/W	Burst ROM Enable Selects whether area 0 is used as a burst ROM interface. 0: Area 0 is basic bus interface 1: Area 0 is burst ROM interface
4	BRSTS1	1	R/W	Burst Cycle Select 1 Selects the number of burst cycles for the burst ROM interface. 0: Burst cycle comprises 1 state 1: Burst cycle comprises 2 states
3	BRSTS0	0	R/W	Burst Cycle Select 0 Selects the number of words that can be accessed in a burst ROM interface burst access. 0: Max. 4 words in burst access 1: Max. 8 words in burst access
2 to 0	—	0	R/W	Reserved The write value should always be 0.

7.3.5 Bus Control Register L (BCRL)

BCRL performs selection of the external bus-released state protocol, and enabling or disabling of the $\overline{\text{WAIT}}$ pin input.

Bit	Bit Name	Initial Value	R/W	Description
7	BRLE	0	R/W	Bus Release Enable Enables or disables external bus release. 0: External bus release is disabled. $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ can be used as I/O ports. 1: External bus release is enabled.
6	—	0	R/W	Reserved The write value should always be 0.
5	—	0	–	Reserved This bit is always read as 0 and cannot be modified.
4	—	0	R/W	Reserved The write value should always be 0.
3	—	1	R/W	Reserved The write value should always be 1.
2, 1	—	All 0	R/W	Reserved The write value should always be 0.
0	WAITE	0	R/W	WAIT Pin Enable Selects enabling or disabling of wait input by the $\overline{\text{WAIT}}$ pin. 0: Wait input by the $\overline{\text{WAIT}}$ pin is disabled. The $\overline{\text{WAIT}}$ pin can be used as I/O port. 1: Wait input by the $\overline{\text{WAIT}}$ pin is enabled.

7.3.6 Pin Function Control Register (PFCR)

PFCR performs address output control in external extended mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	BUZZE	0	R/W	BUZZ Output Enable Enables/disables BUZZ output of the PF1 pin. Input clock of WDT_1 selected by the PSS, CKS2 to CKS0 bits is output as BUZZ signal. 0 : Functions as PF1 input/output pins 1 : Functions as BUZZ output pins
4	—	0	R/W	Reserved The write value should always be 0.
3	AE3	0	R/W	Address Output Enable 3 to 0 These bits select enabling or disabling of address outputs A8 to A23 in ROM extended mode. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. When a pin is disabled for address output, it becomes an output port when the corresponding DDR bit is set to 1. 0000: A8 to A23 output disabled. 0001: A8 output enabled. A9 to A23 output disabled. 0010: A8 and A9 output enabled. A10 to A23 output disabled. 0011: A8 to A10 output enabled. A11 to A23 output disabled. 0100: A8 to A11 output enabled. A12 to A23 output disabled. 0101: A8 to A12 output enabled. A13 to A23 output disabled. 0110: A8 to A13 output enabled. A14 to A23 output disabled. 0111: A8 to A14 output enabled. A15 to A23 output disabled. 1000: A8 t o A15 output enabled. A16 to A23 output disabled. 1001: A8 to A16 output enabled. A17 to A23 output disabled. 1010: A8 to A17 output enabled. A18 to A23 output disabled. 1011: A8 to A18 output enabled. A19 to A23 output disabled. 1100: A8 to A19 output enabled. A20 to A23 output disabled. 1101: A8 to A20 output enabled. A21 to A23 output disabled. 1110: A8 to A21 output enabled. A22 and A23 output disabled. 1111: A8 to A23 output enabled.
2	AE2	0	R/W	
1	AE1	0	R/W	
0	AE0	0	R/W	

7.4 Bus Control

7.4.1 Area Divisions

In advanced mode, the bus controller partitions the 16-Mbyte address space into eight areas, area 0 to area 7, in 2-Mbyte units, and performs bus control for external address space in area units. In normal mode*, it controls a 64-kbyte address space comprising part of area 0.

Figure 7.2 shows an outline of the memory map.

The chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area.

Note: * Not available in this LSI.

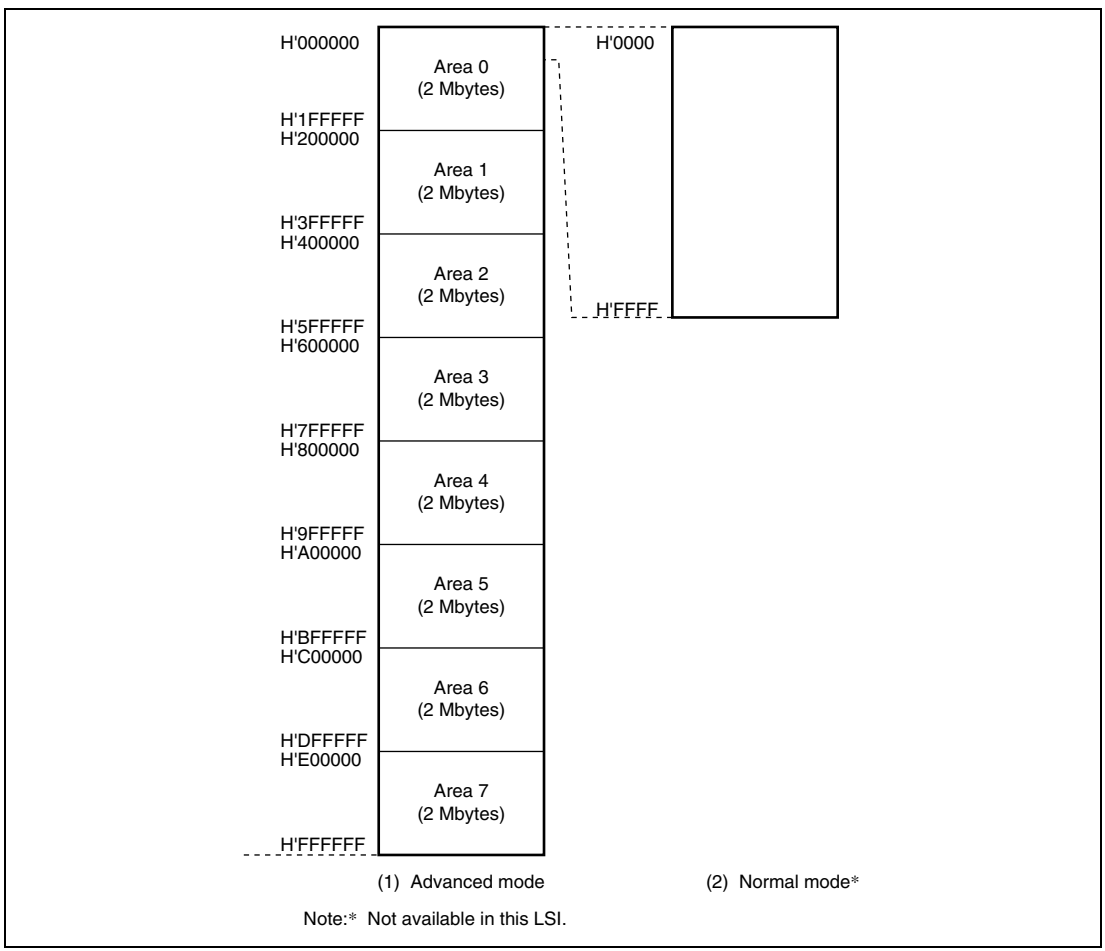


Figure 7.2 Overview of Area Divisions

7.4.2 Bus Specifications

The external address space bus specifications consist of three elements: bus width, number of access states, and number of program wait states.

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

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

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is designated for 16-bit access, 16-bit bus mode is set. When the burst ROM interface is designated, 16-bit bus mode is always set.

Number of access states: Two or three access states can be selected with ASTCR. An area for which 2-state access is selected functions as a 2-state access space, and an area for which 3-state access is selected functions as a 3-state access space.

With the burst ROM interface, the number of access states may be determined without regarding to ASTCR.

When 2-state access space is designated, wait insertion is disabled.

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

Table 7.2 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Number of Access States	Number of 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		3	0
			1			1
		1	0			2
			1			3

7.4.3 Bus Interface for Each Area

The initial state of each area is basic bus interface, 3-state access space. The initial bus width is selected according to the operating mode. The bus specifications described here cover basic items only, and sections 7.6, Basic Bus Interface, and 7.7, Burst ROM Interface, on each memory interface should be referred to for further details.

Area 0: Area 0 includes on-chip ROM, and in ROM-enabled extended mode, space excluding on-chip ROM is external address space.

When external address space of area 0 is accessed, the $\overline{CS0}$ signal can be output.

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

Areas 1 to 6: In external extended mode, all of areas 1 to 6 are external address spaces. When external address spaces of areas 1 to 6 are accessed, the $\overline{CS1}$ to $\overline{CS6}$ pin signals can be output respectively. Only the basic bus interface can be used for areas 1 to 6.

Area 7: Area 7 includes on-chip RAM and internal l/O registers. In external extended mode, the space excluding on-chip RAM and internal l/O registers, is external address 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 address space.

When external address space of area 7 is accessed, the $\overline{CS7}$ signal can be output.

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

7.4.4 Chip Select Signals

This LSI can output chip select signals ($\overline{CS0}$ to $\overline{CS7}$) to areas 0 to 7, and these signals are driven low respectively when the corresponding external address space area is accessed. Figure 7.3 shows an example of $\overline{CSn}$ (n = 0 to 7) signal output timing. Enabling or disabling of the $\overline{CSn}$ signal is performed by setting the data direction register (DDR) for the port corresponding to the particular $\overline{CSn}$ pin.

In ROM-enabled extended mode, pins $\overline{CS0}$ to $\overline{CS7}$ are all placed in the input state after a power-on reset, and so the corresponding DDR should be set to 1 when outputting signals $\overline{CS0}$ to $\overline{CS7}$. For details, see section 9, I/O Ports.

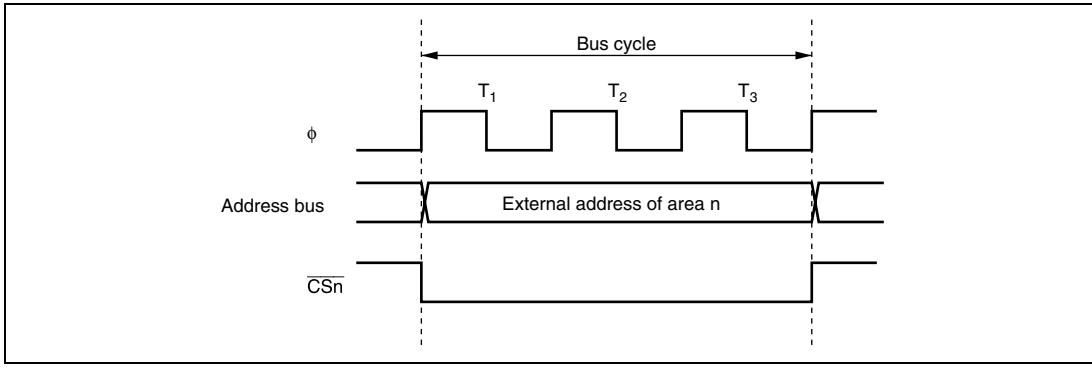


Figure 7.3 $\overline{CSn}$ Signal Output Timing (n = 0 to 7)

7.5 Basic Timing

The CPU is driven by a system clock, denoted by the symbol ϕ. The period from one rising edge of ϕ to the next is referred to as a "state." The memory cycle or bus cycle consists of one, two, or three states. Different methods are used to access on-chip memory, on-chip peripheral modules, and the external address space.

7.5.1 On-Chip Memory (ROM, RAM) Access Timing

On-chip memory is accessed in one state. The data bus width is 16 bits, enabling both byte and word transfer. Figure 7.4 shows the on-chip memory access cycle. Figure 7.5 shows the pin states.

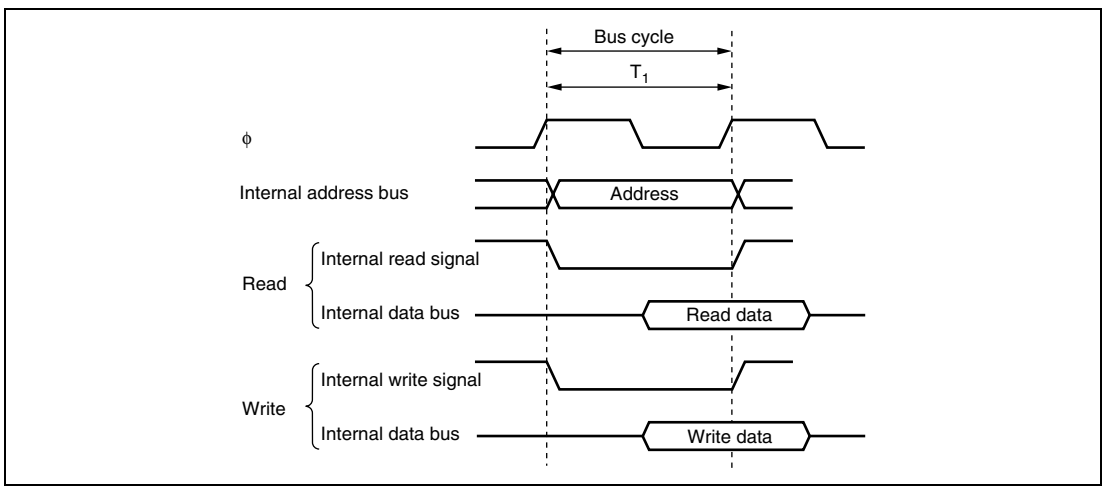


Figure 7.4 On-Chip Memory Access Cycle

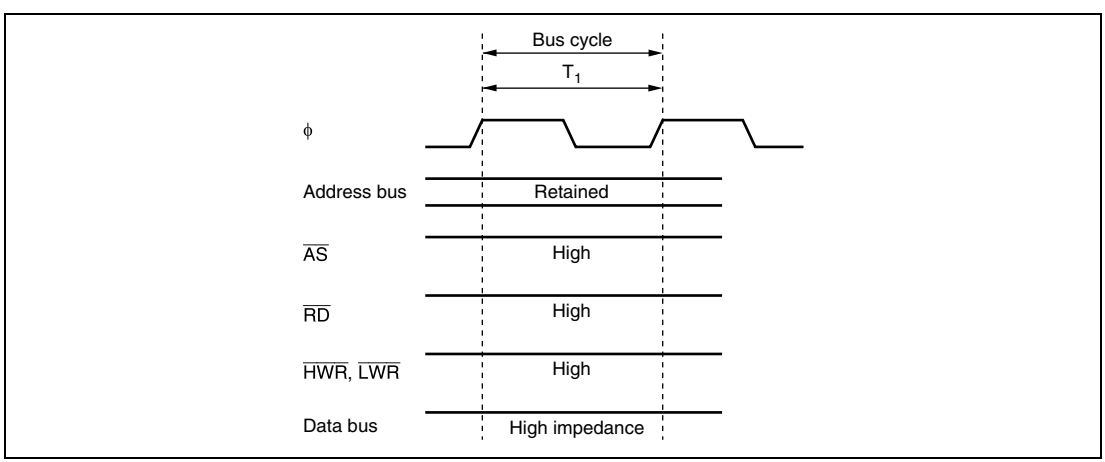


Figure 7.5 Pin States during On-Chip Memory Access

7.5.2 On-Chip Peripheral Module Access Timing

On-chip peripheral module access timing excluding port H, port J, IIC2, IEB, and HCAN: The on-chip peripheral modules are accessed in two states except for port H, port J, IIC2, IEB, and HCAN. The data bus width is either 8 bits or 16 bits, depending on the particular internal I/O register being accessed. Figure 7.6 shows the access timing for the on-chip peripheral modules. Figure 7.7 shows the pin states.

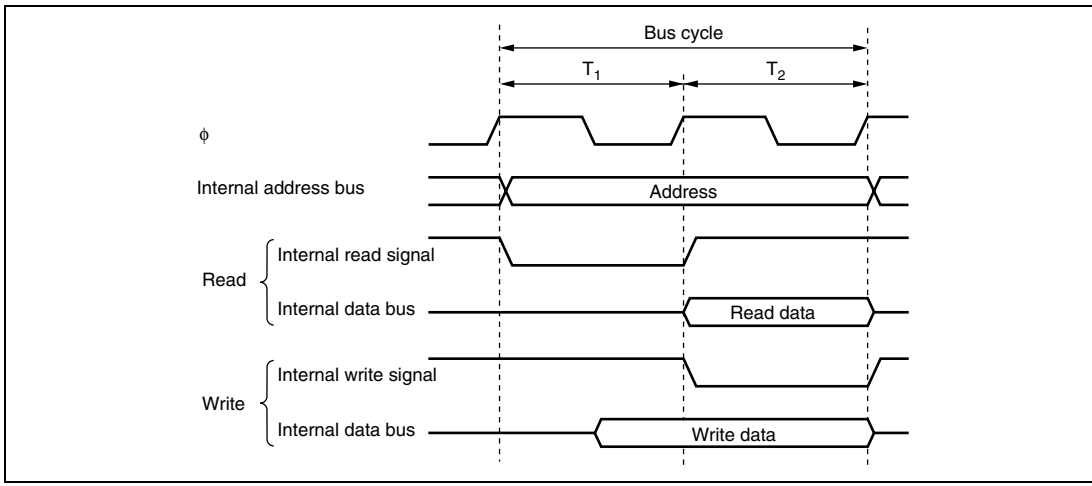


Figure 7.6 On-Chip Peripheral Module Access Cycle

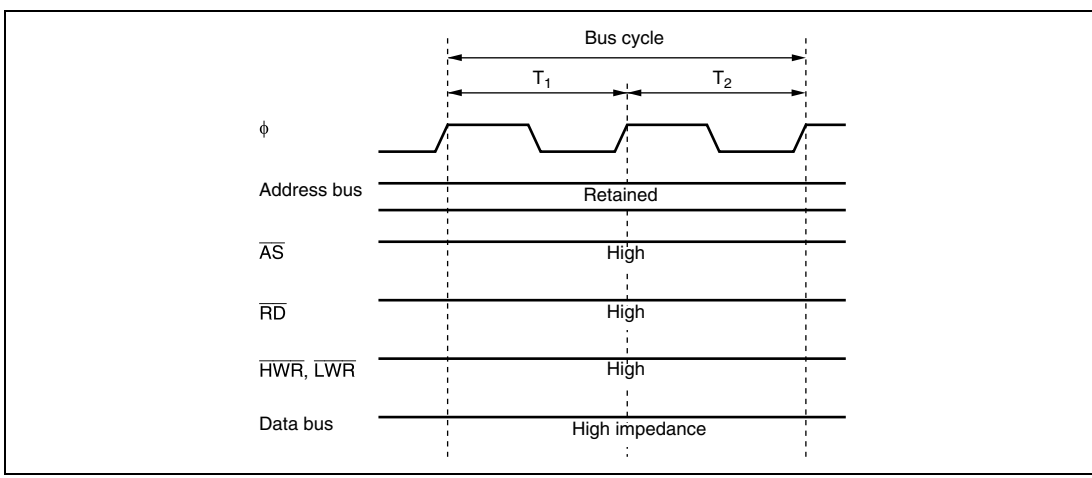


Figure 7.7 Pin States during On-Chip Peripheral Module Access

On-chip port H, port J, and IIC2 module access timing: On-chip port H, port J, and IIC2 modules are accessed in four states. At this time, the data bus width is 8 bits. Figure 7.8 shows on-chip port H, port J, and IIC2 module access timing, and figure 7.9 shows the pin states.

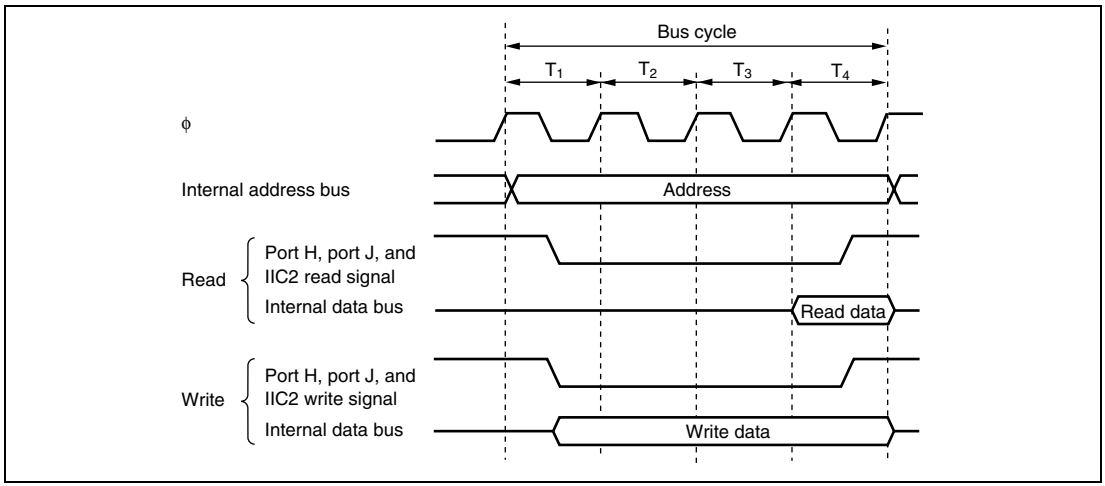


Figure 7.8 On-Chip Port H, Port J, and IIC2 Module Access Cycle

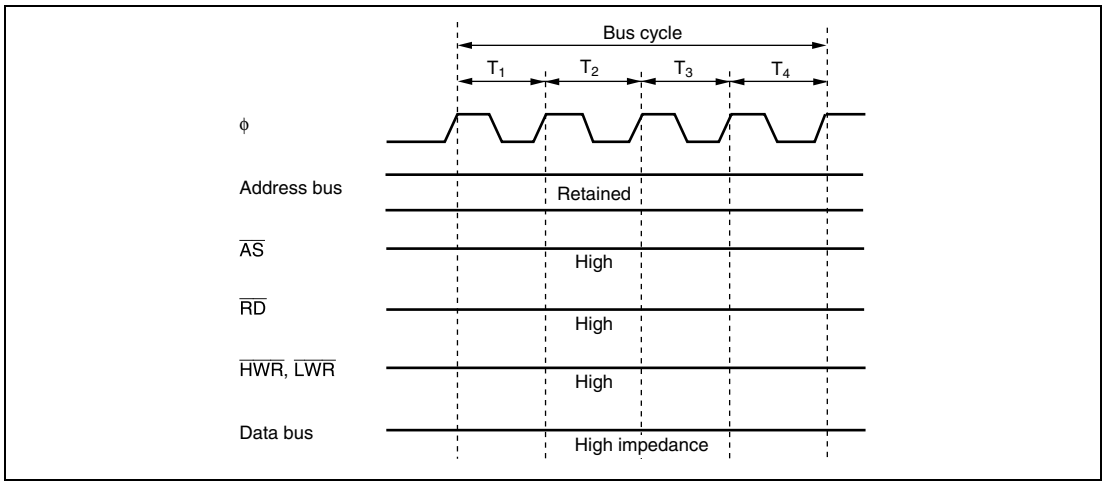


Figure 7.9 Pin States during On-Chip Port H, Port J, and IIC2 Module Access

On-chip IEB module access timing (H8S/2552 Series only): On-chip IEB module is accessed in five states. At this time, the data bus width is 8 bits. Figure 7.10 shows on-chip IEB module access timing, and figure 7.11 shows the pin states.

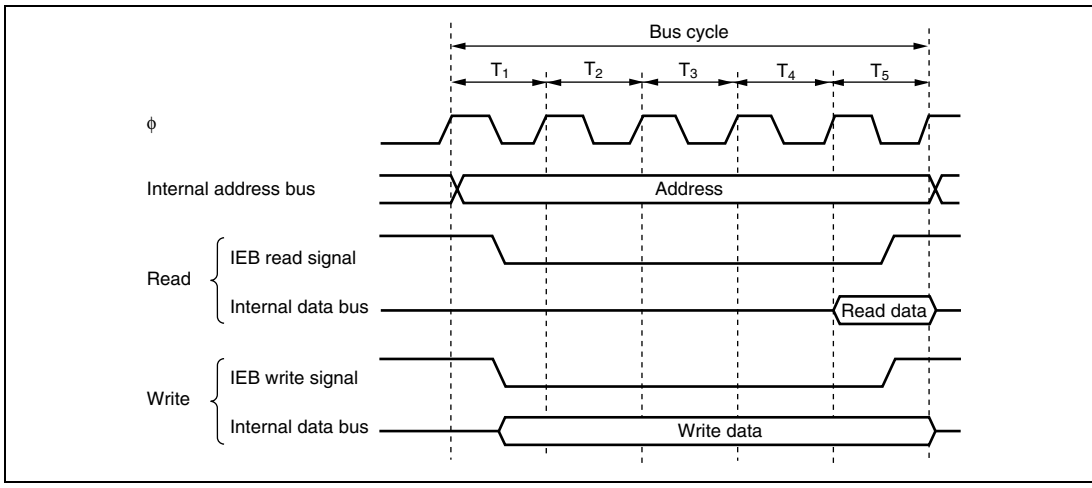


Figure 7.10 On-Chip IEB Module Access Cycle

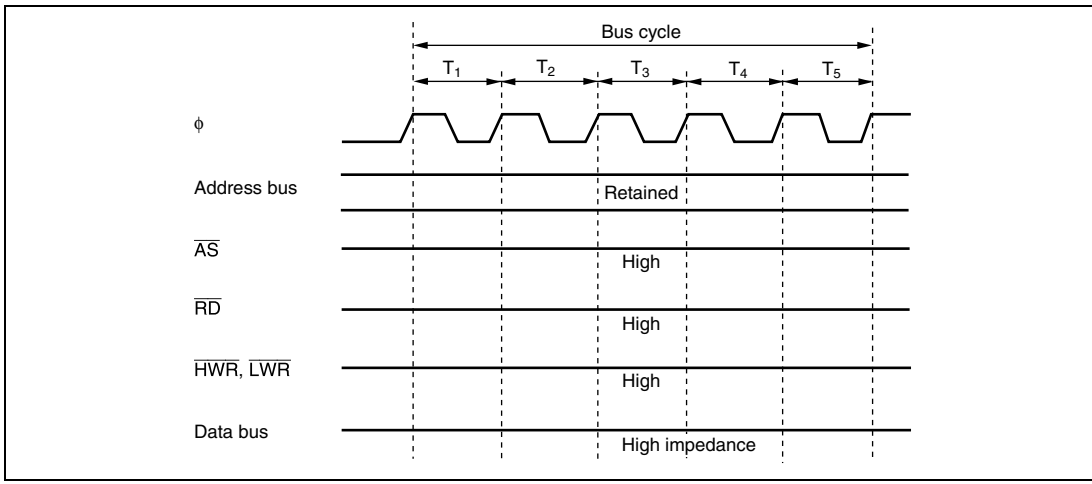


Figure 7.11 Pin States during On-Chip IEB Module Access

On-chip HCAN module access timing (H8S/2556 Series only): On-chip HCAN module is accessed in five states. At this time, the data bus width is 16 bits. Figure 7.12 shows on-chip HCAN module access timing, and figure 7.13 shows the pin states.

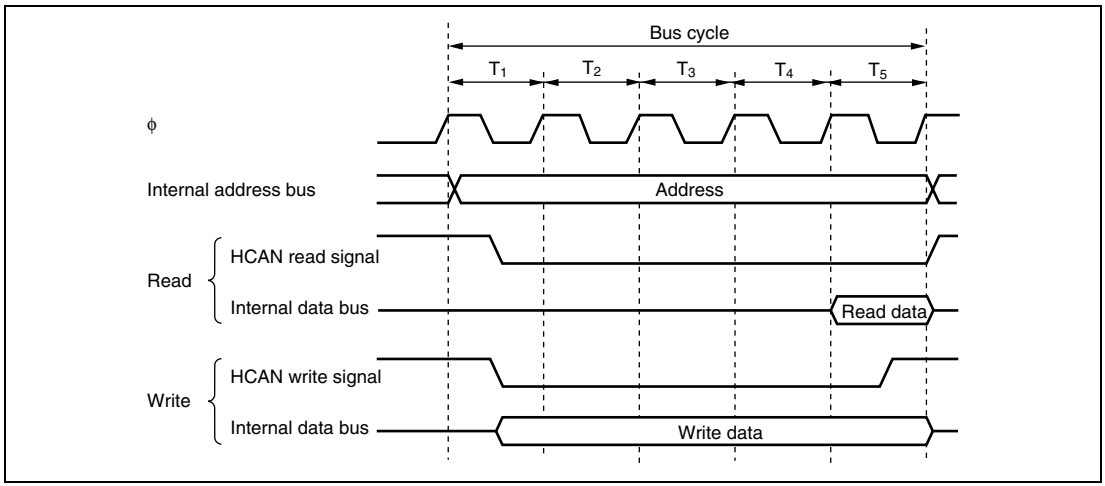


Figure 7.12 On-Chip HCAN Module Access Cycle

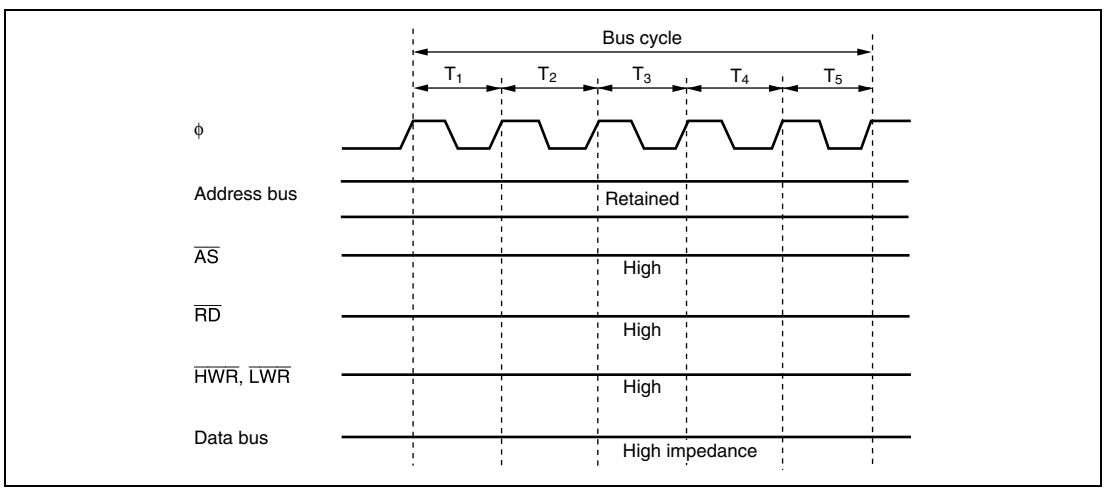


Figure 7.13 Pin States during On-Chip HCAN Module Access

7.5.3 External Address Space Access Timing

The external address space is accessed with an 8-bit or 16-bit data bus width in a two-state or three-state bus cycle. In three-state access, wait states can be inserted. For further details, refer to section 7.6.3, Basic Timing.

7.6 Basic Bus Interface

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

7.6.1 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. When accessing external address space, it controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used, according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-bit access space: Figure 7.14 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word transfer instruction is performed as two-byte accesses, and a longword transfer instruction, as four-byte accesses.

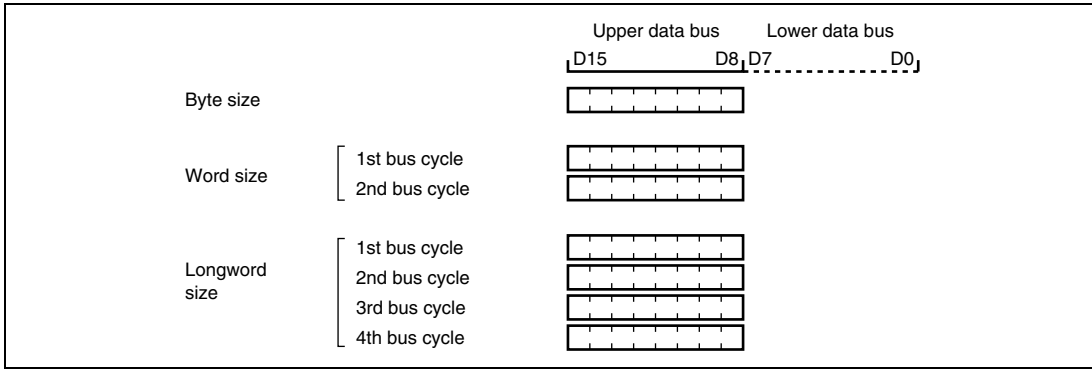


Figure 7.14 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-bit access space: Figure 7.15 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword transfer instruction is performed 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.

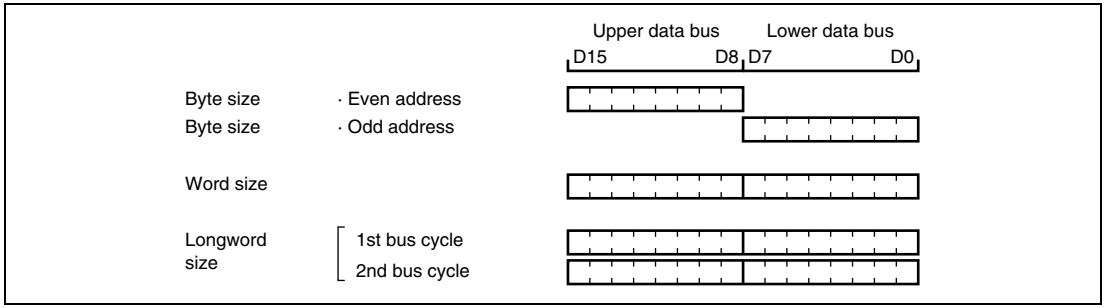


Figure 7.15 Access Sizes and Data Alignment Control (16-Bit Access Space)

7.6.2 Valid Strobes

Table 7.3 shows the data buses used and valid strobes for the access spaces.

In read access, the $\overline{RD}$ signal is valid without discrimination between the upper and lower halves of the data bus.

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

Table 7.3 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower Data Bus (D7 to D0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$	Valid	Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd	$\overline{RD}$	Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Legend

Hi-Z: High impedance

Invalid: Input state; input value is ignored.

7.6.3 Basic Timing

8-bit 2-state access space: Figure 7.16 shows the bus timing for an 8-bit 2-state access space. When an 8-bit access space is accessed , the upper half (D15 to D8) of the data bus is used.

Wait states cannot be inserted.

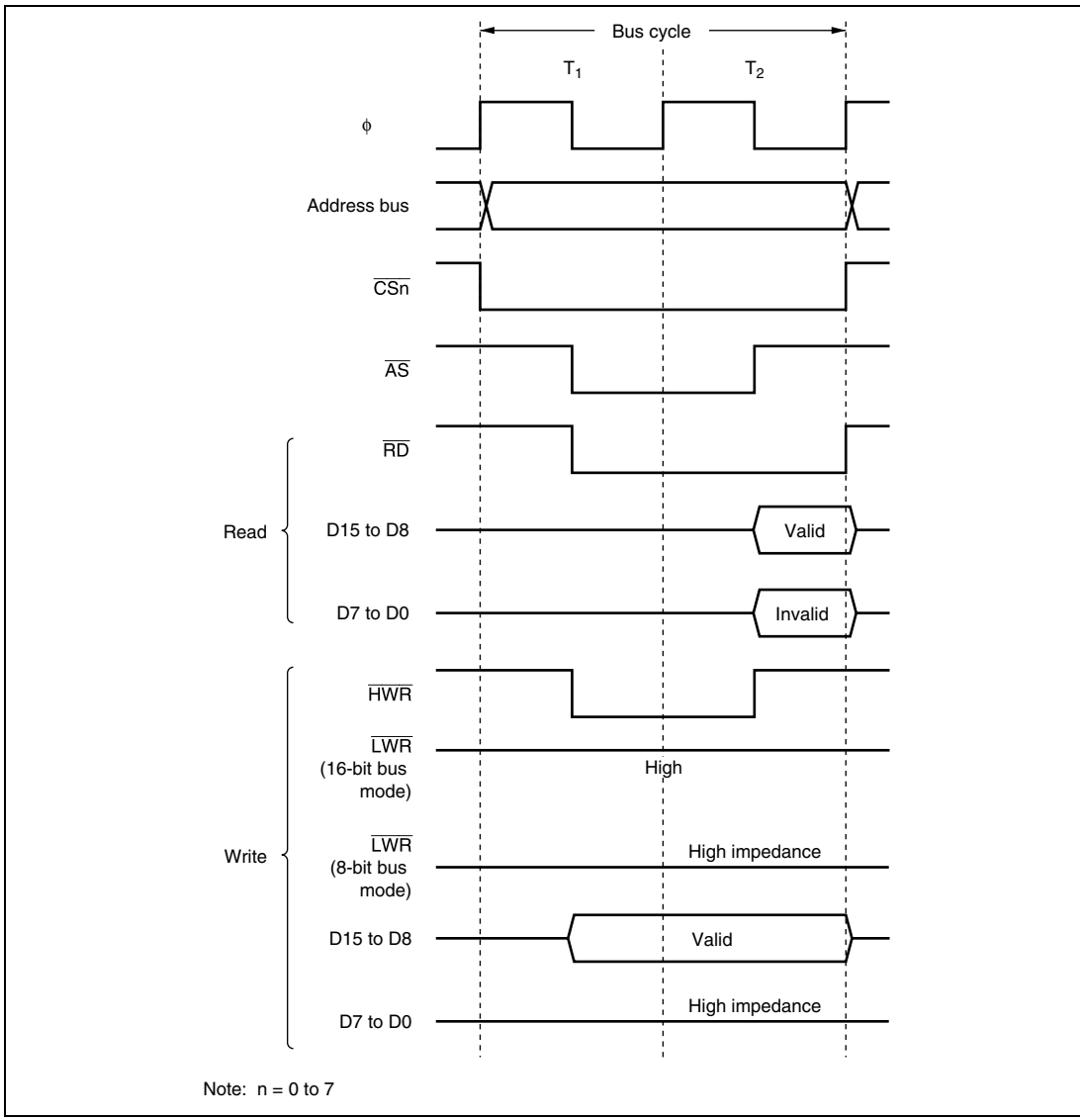


Figure 7.16 Bus Timing for 8-Bit 2-State Access Space

8-bit 3-state access space: Figure 7.17 shows the bus timing for an 8-bit 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

Wait states can be inserted.

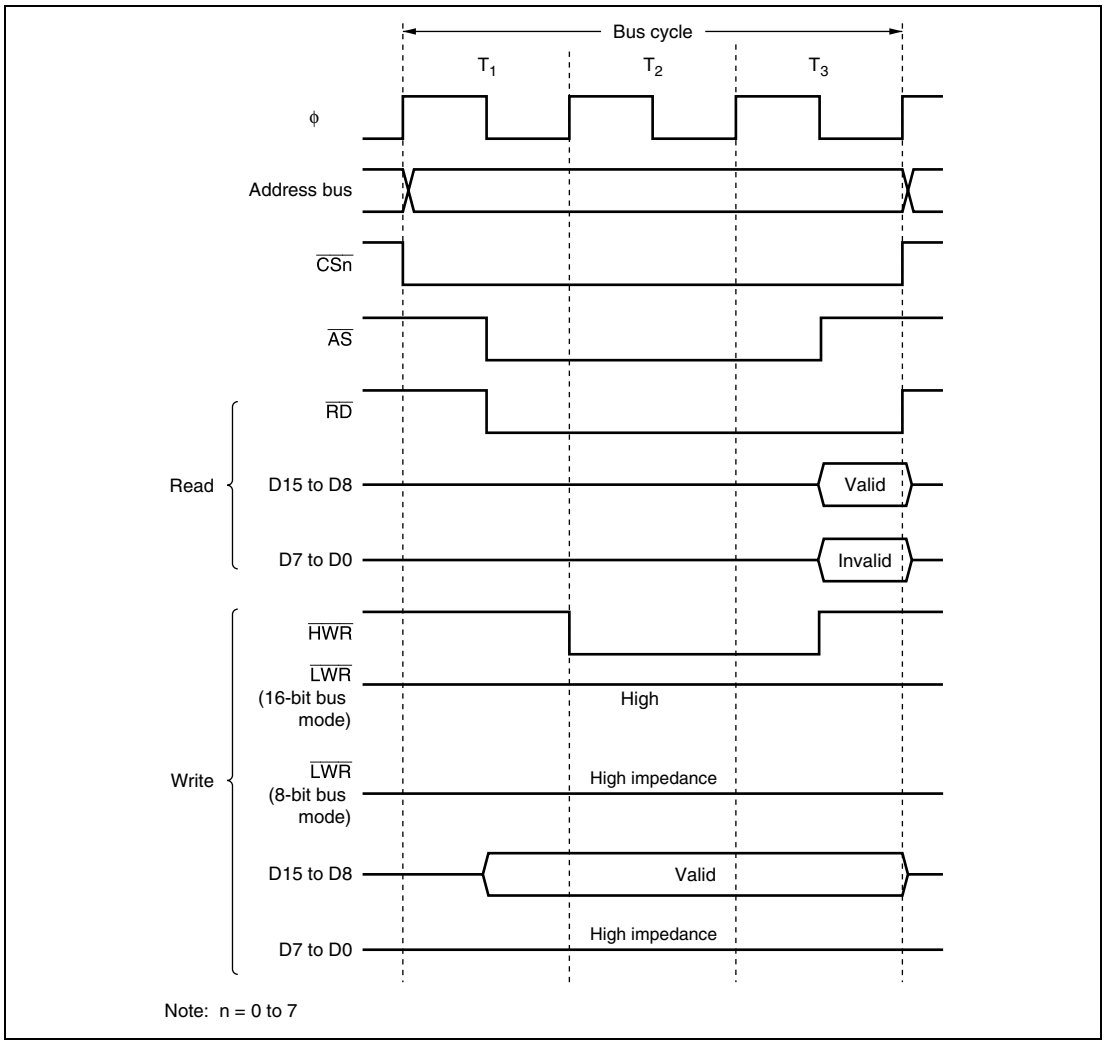


Figure 7.17 Bus Timing for 8-Bit 3-State Access Space

16-bit 2-state access space: Figures 7.18 to 7.20 show bus timings for a 16-bit 2-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for an even address, and the lower half (D7 to D0) for an odd address.

Wait states cannot be inserted.

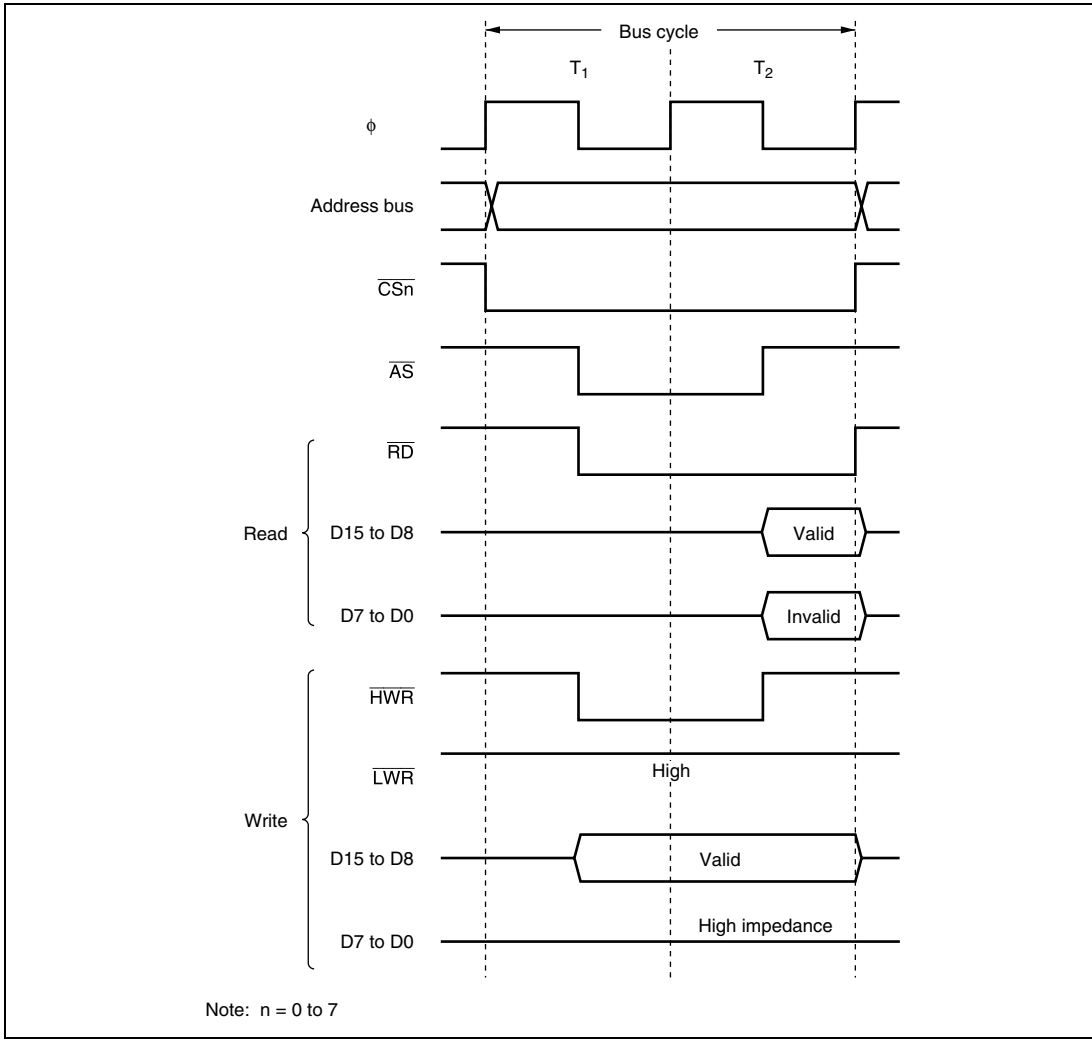


Figure 7.18 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte Access)

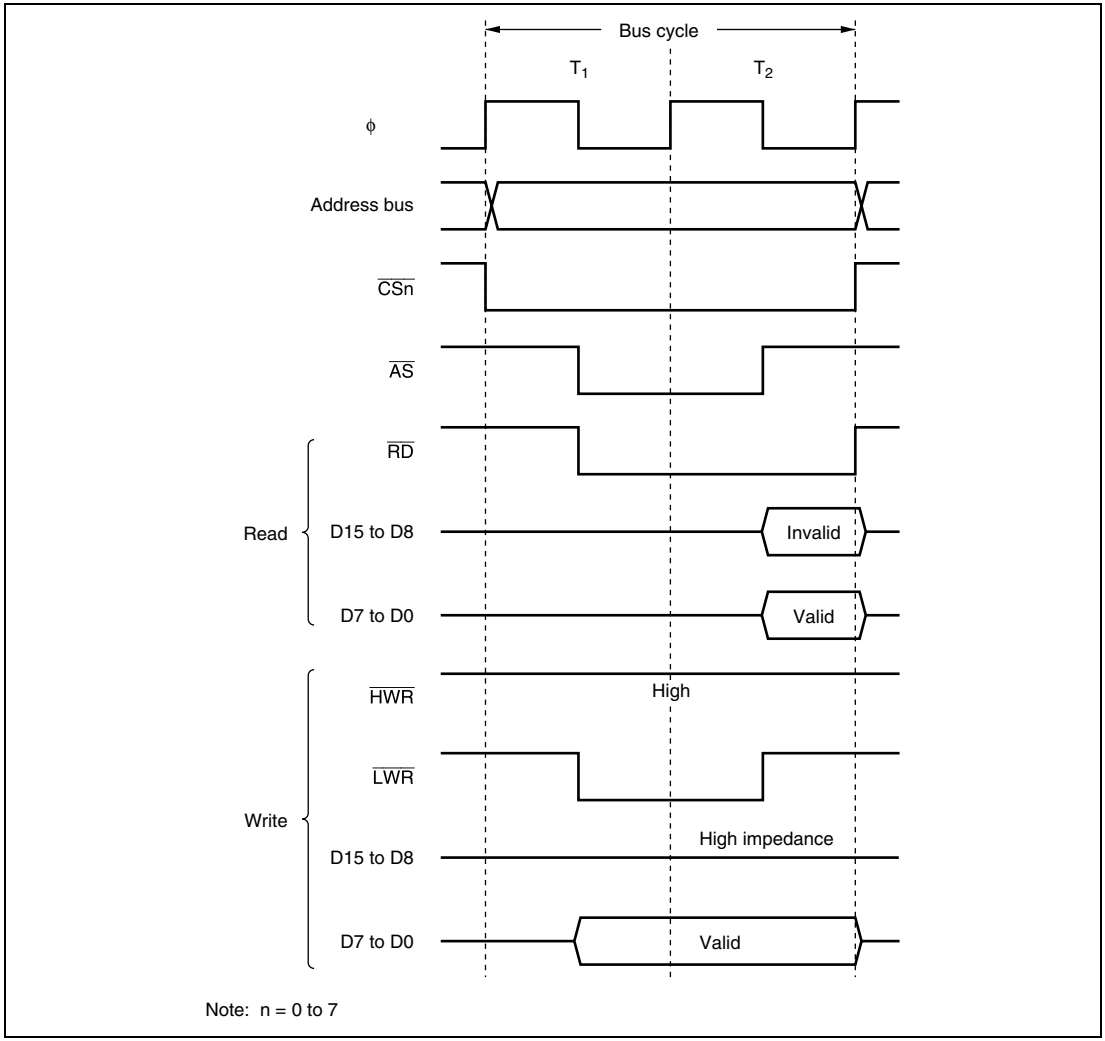


Figure 7.19 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte Access)

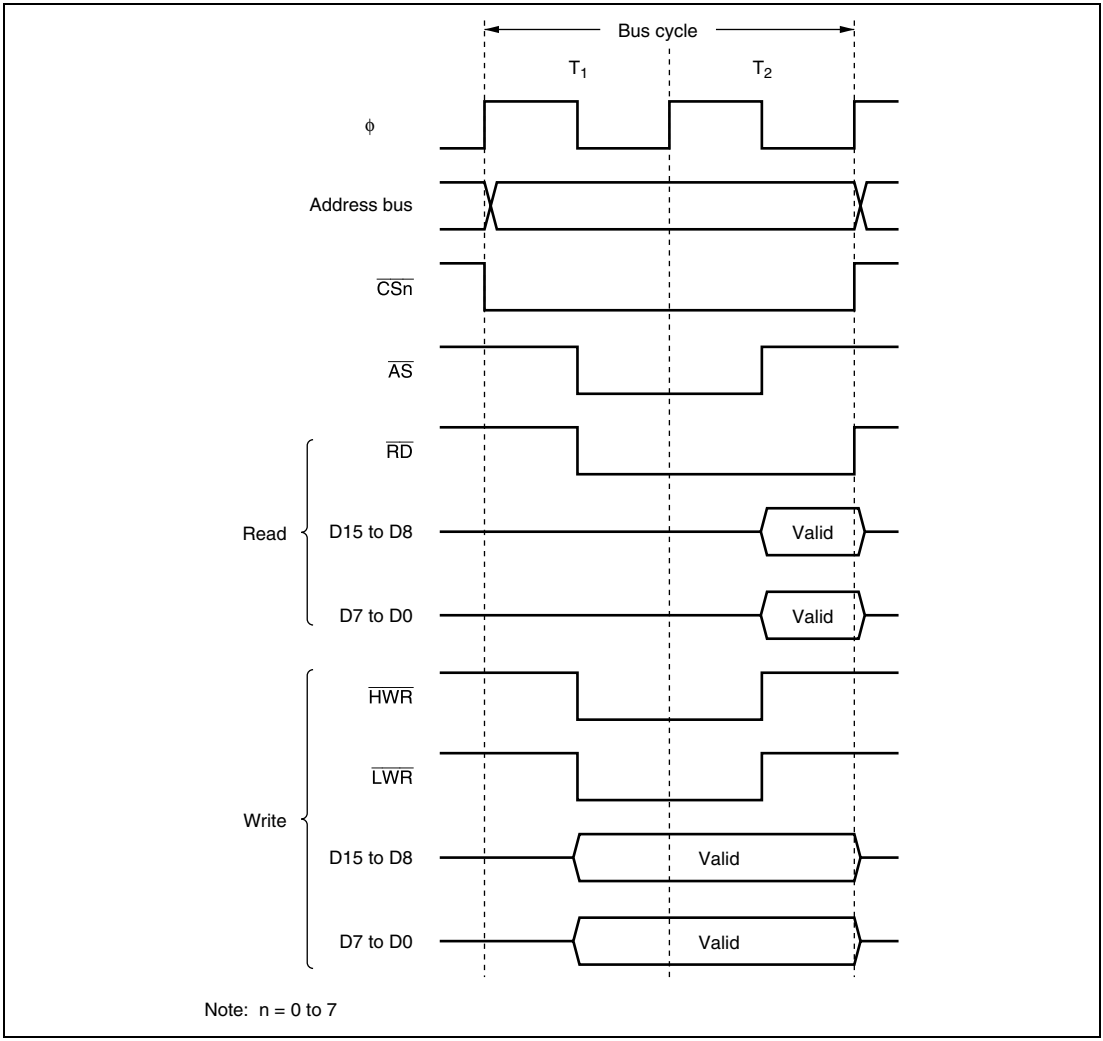


Figure 7.20 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access)

16-bit 3-state access space: Figures 7.21 to 7.23 show bus timings for a 16-bit 3-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for an even address, and the lower half (D7 to D0) for an odd address.

Wait states can be inserted.

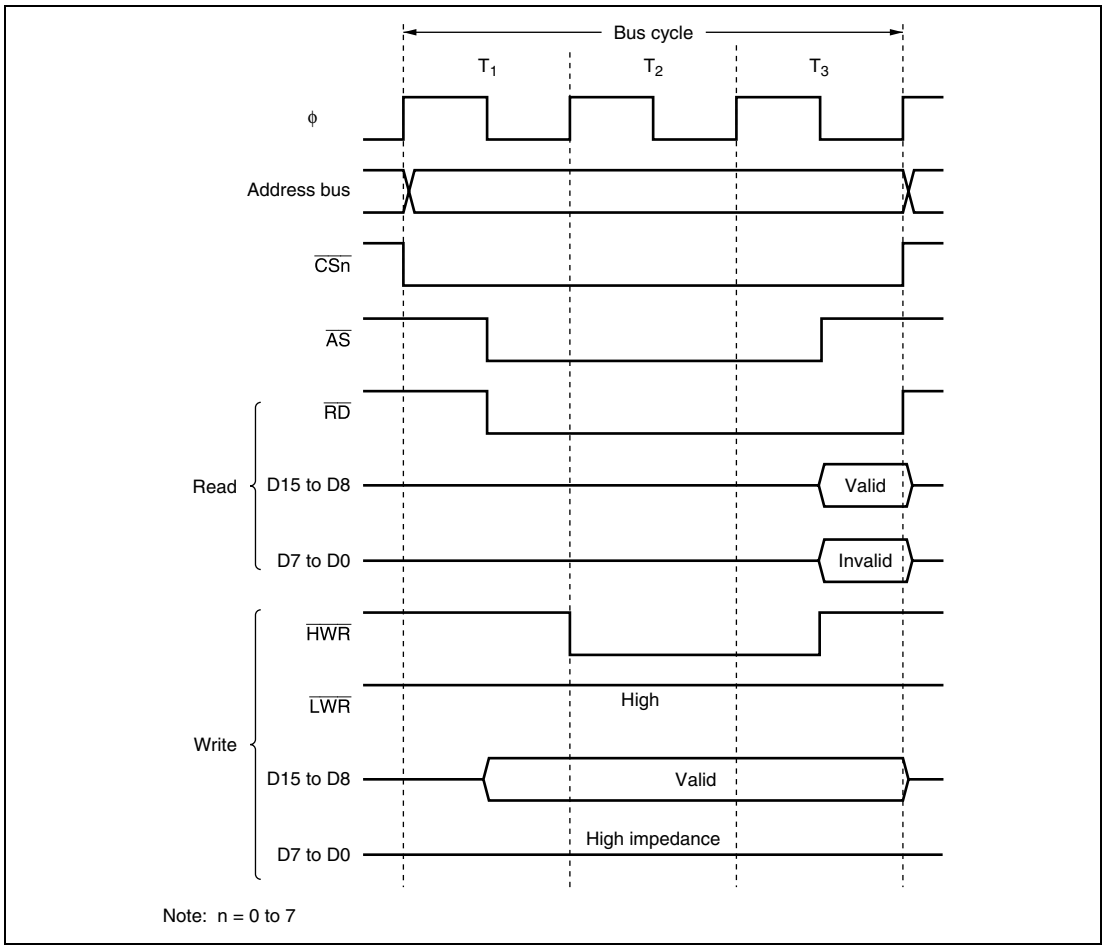


Figure 7.21 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte Access)

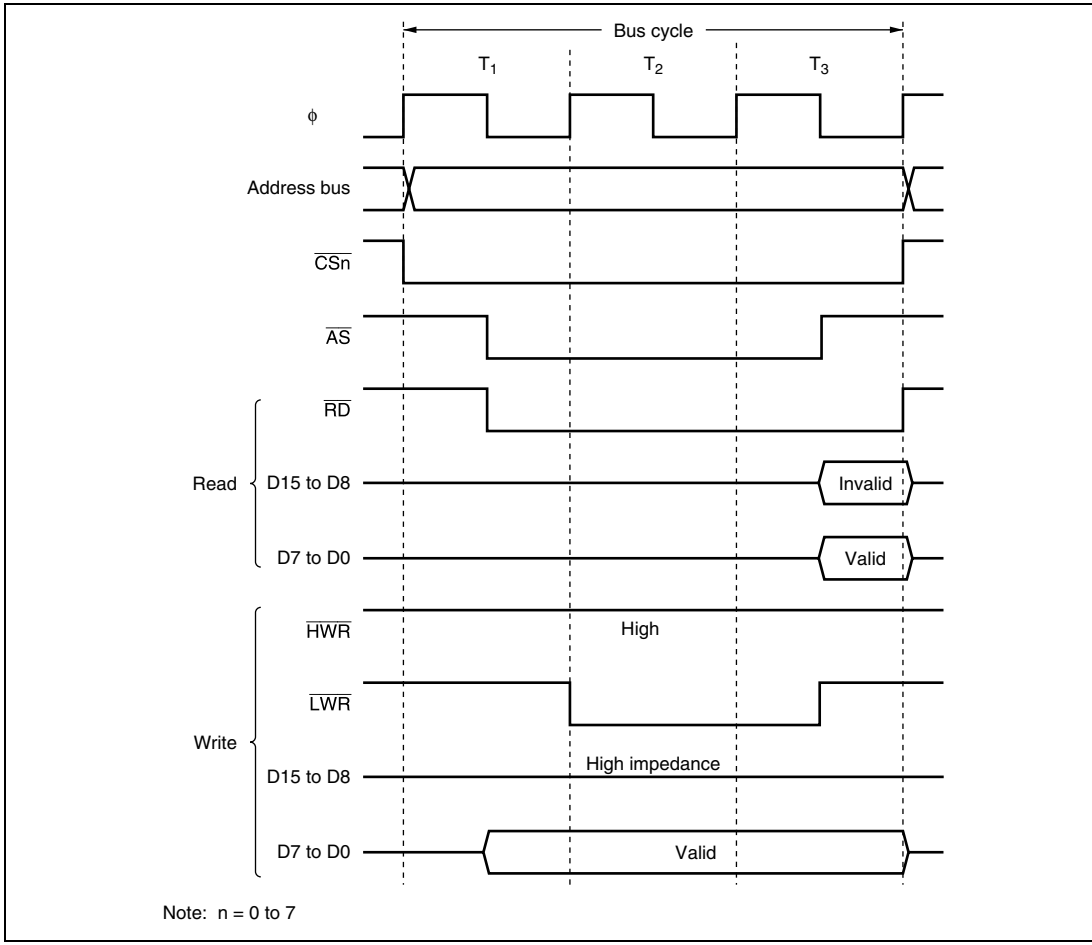


Figure 7.22 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byte Access)

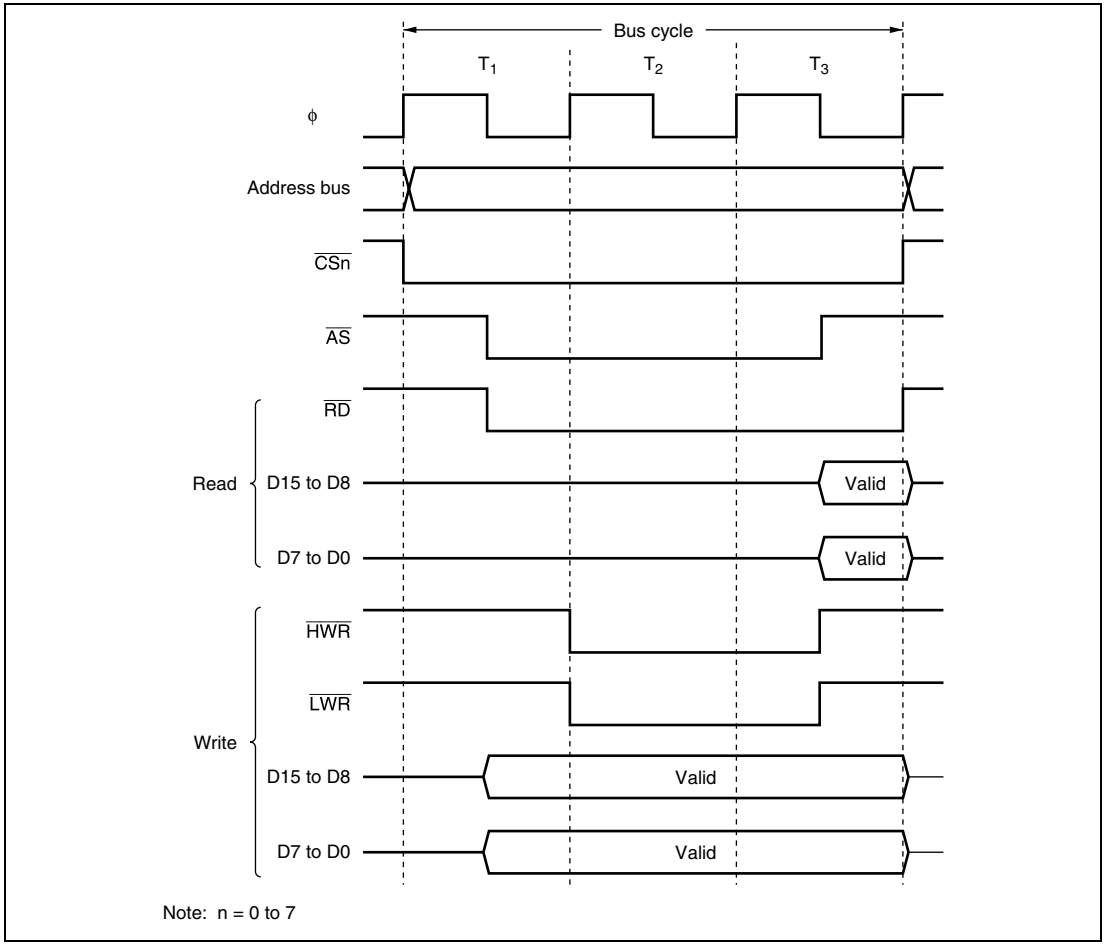


Figure 7.23 Bus Timing for 16-Bit 3-State Access Space (3) (Word Access)

7.6.4 Wait Control

When accessing external address space, this LSI can extend the bus cycle by inserting one or more wait states (Tw).

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 3-state access space, according to the settings of WCRH and WCRL.

Pin wait insertion: Setting the WAITE bit in BCRH to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. When external address space is accessed in this state, program wait insertion is first carried out according to the settings of WCRH and WCRL. Then, if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, a T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

Figure 7.24 shows an example of wait state insertion timing.

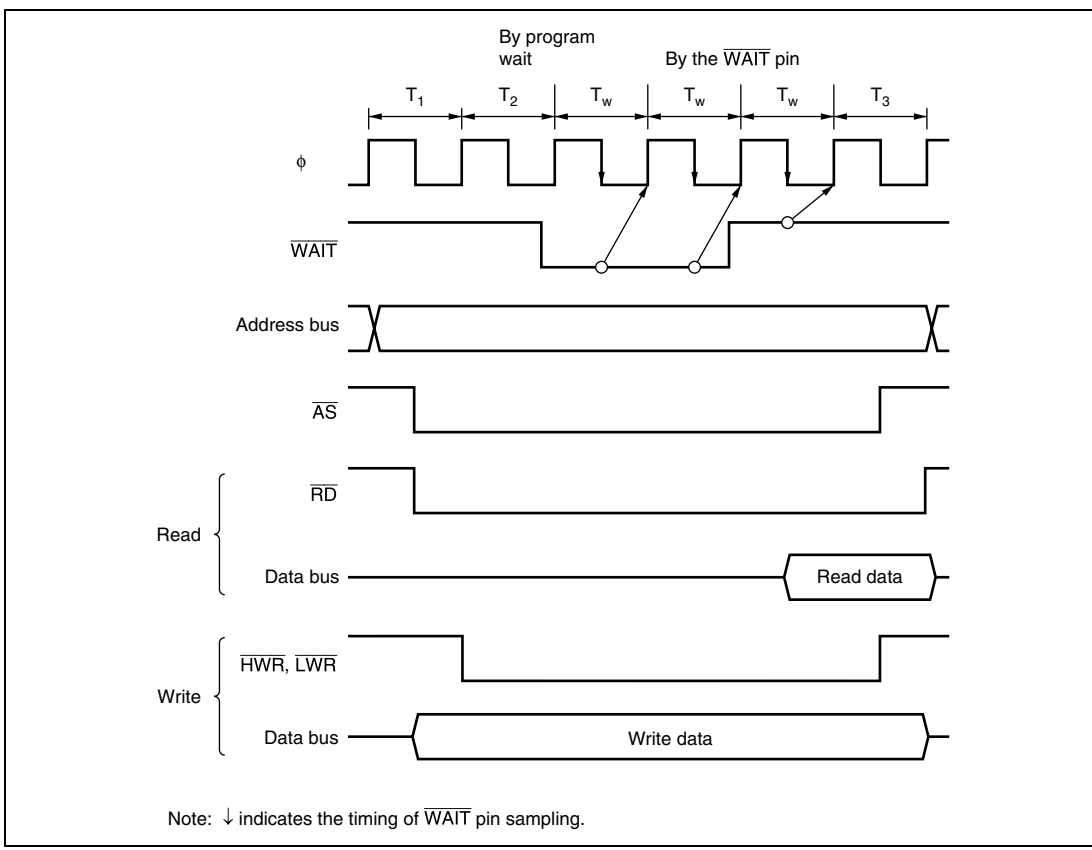


Figure 7.24 Example of Wait State Insertion Timing

7.7 Burst ROM Interface

With this LSI, external address space of area 0 can be designated as burst ROM space, and burst ROM interfacing can be performed. The burst ROM space interface enables 16-bit configuration ROM with burst access capability to be accessed at high speed.

Area 0 can be designated as burst ROM space by means of the BRSTRM bit in BCRH. Consecutive burst accesses of a maximum of 4 words or 8 words can be performed for CPU instruction fetches only. One or two states can be selected for burst access.

7.7.1 Basic Timing

The number of access states in the initial cycle (full access) of the burst ROM interface is in accordance with the setting of the AST0 bit in ASTCR. Also, when the AST0 bit is set to 1, wait states can be inserted. One or two states can be selected for the burst cycle, according to the setting of the BRSTS1 bit in BCRH. Wait states cannot be inserted. When area 0 is designated as burst ROM space, it becomes 16-bit access space regardless of the setting of the ABW0 bit in ABWCR.

When the BRSTS0 bit in BCRH is cleared to 0, burst access of up to 4 words is performed; when the BRSTS0 bit is set to 1, burst access of up to 8 words is performed.

The basic access timing for burst ROM space is shown in figures 7.25 and 7.26. The timing shown in figure 7.25 is for the case where the AST0 and BRSTS1 bits are both set to 1, and that in figure 7.26 is for the case where both these bits are cleared to 0.

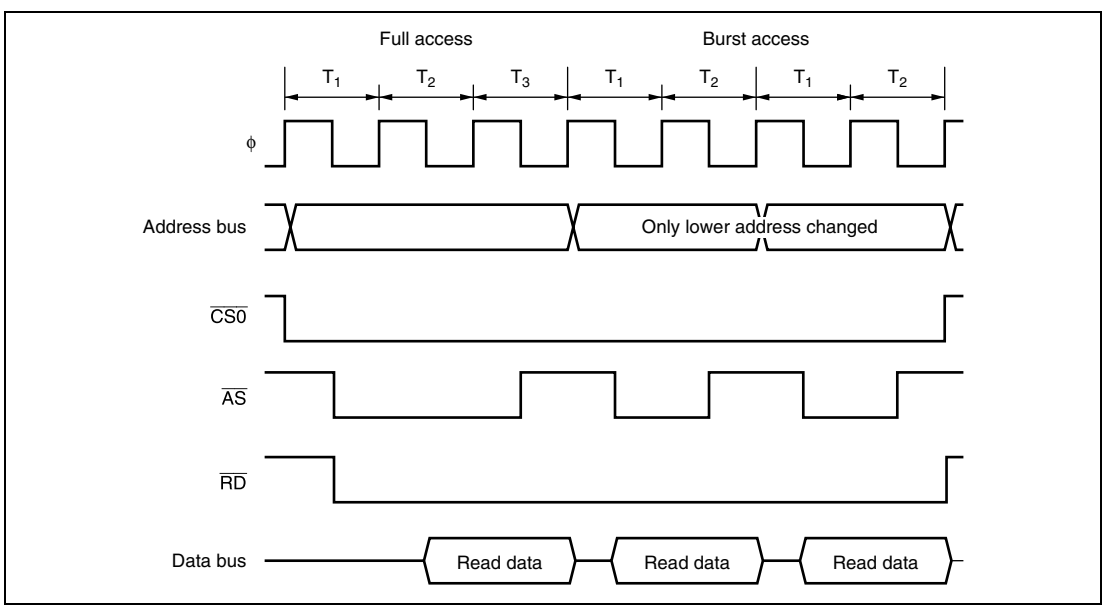


Figure 7.25 Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 1)

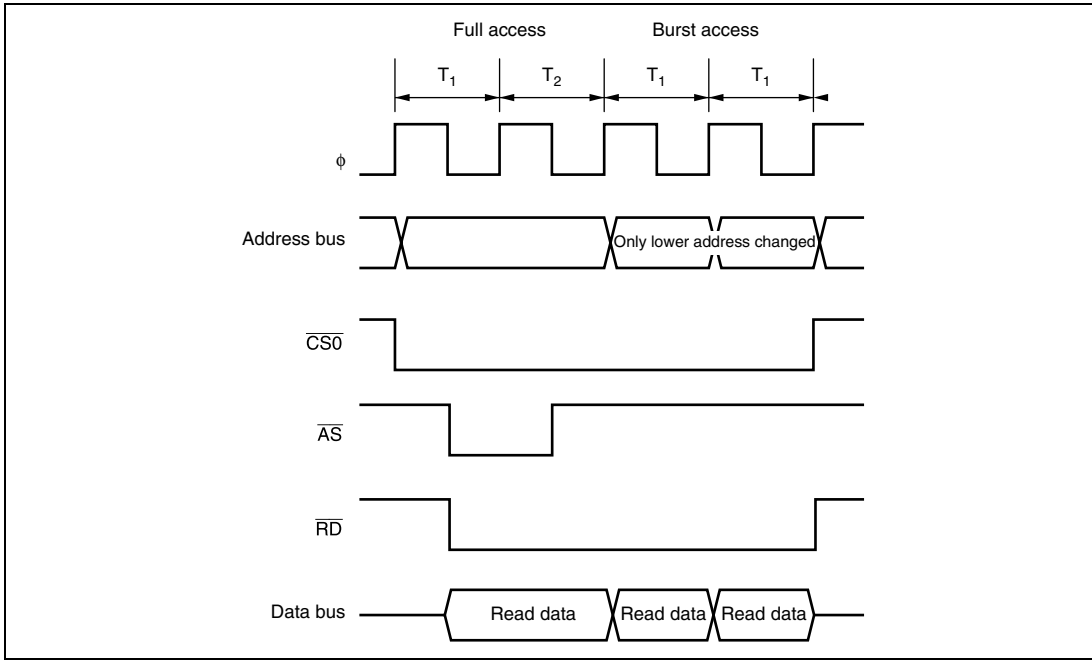


Figure 7.26 Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 0)

7.7.2 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the $\overline{\text{WAIT}}$ pin can be used in the initial cycle (full access) of the burst ROM interface. See section 7.6.4, Wait Control.

Wait states cannot be inserted in a burst cycle.

7.8 Idle Cycle

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

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

Figure 7.27 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.

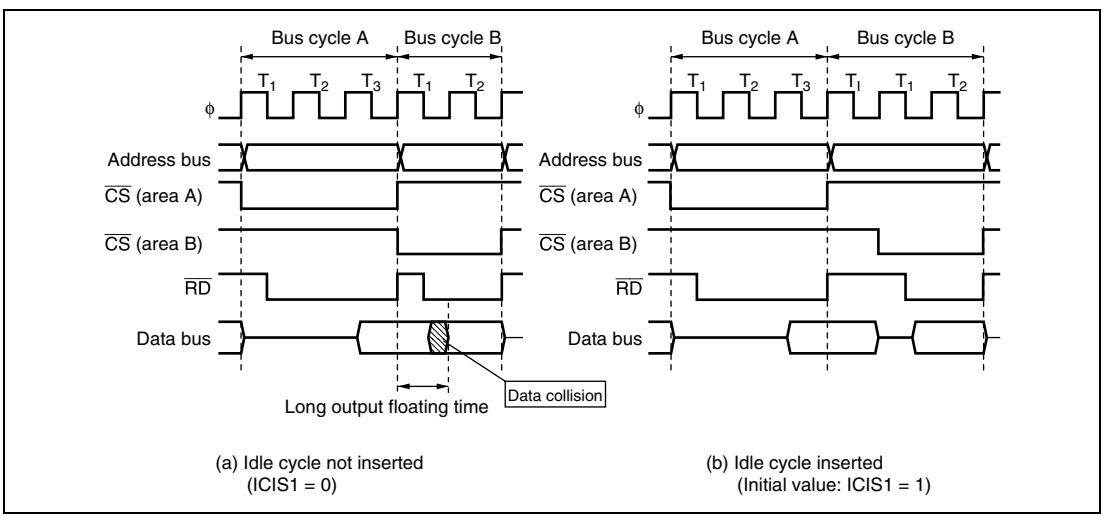


Figure 7.27 Example of Idle Cycle Operation (1)

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

Figure 7.28 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.

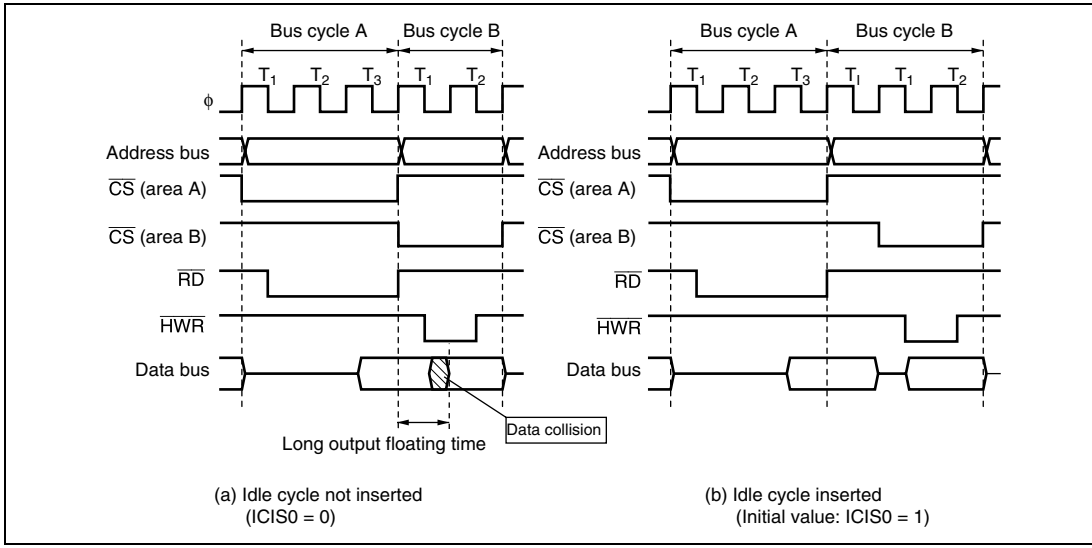


Figure 7.28 Example of Idle Cycle Operation (2)

Relationship between chip select ($\overline{CS}$) signal and read ($\overline{RD}$) signal: Depending on the system's load conditions, the $\overline{RD}$ signal may lag behind the $\overline{CS}$ signal. An example is shown in figure 7.29.

In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A $\overline{RD}$ signal and the bus cycle B $\overline{CS}$ signal.

Setting idle cycle insertion, as in (b), however, will prevent any overlap between the $\overline{RD}$ and $\overline{CS}$ signals.

In the initial state after reset release, idle cycle insertion (b) is set.

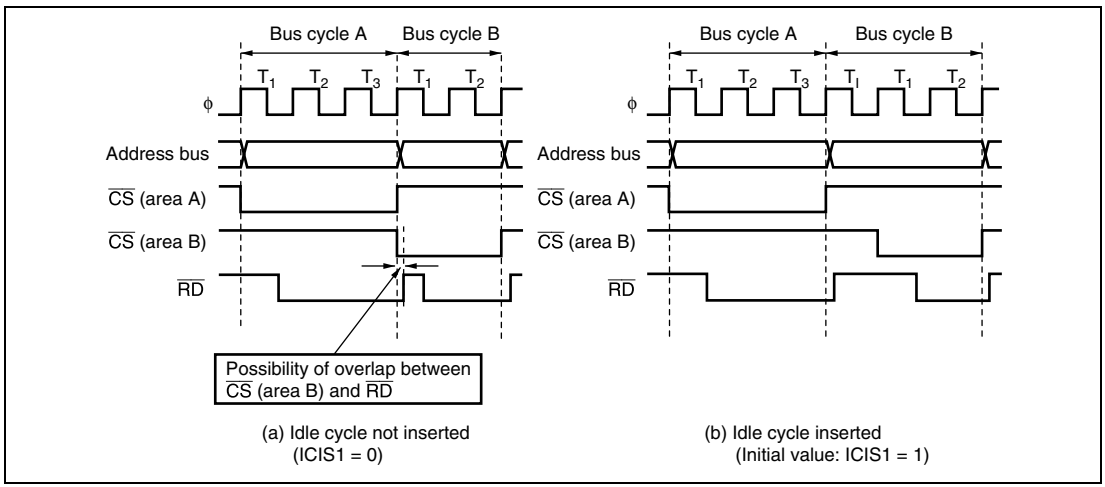


Figure 7.29 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

Table 7.4 shows the pin states in an idle cycle.

Table 7.4 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of next bus cycle
D15 to D0	High impedance
$\overline{CSn}$	High
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High

7.9 Bus Release

This LSI can release the external bus in response to a bus mastership request from an external device. In the external bus mastership released state, the internal bus master continues to operate as long as there is no external access.

In external extended mode, the bus mastership can be released to an external device by setting the BRLE bit in BCRL to 1. Driving the $\overline{\text{BREQ}}$ pin low issues an external bus mastership request to this LSI. When the $\overline{\text{BREQ}}$ pin is sampled, the $\overline{\text{BACK}}$ pin is driven low at the prescribed timing, and the address bus, data bus, and bus control signals are placed in the high-impedance state, establishing the external bus mastership released state.

In the external bus mastership released state, an internal bus master can perform accesses using the internal bus. When an internal bus master wants to make an external access, it temporarily defers activation of the bus cycle, and waits for the bus mastership request from the external bus master to be dropped.

When the $\overline{\text{BREQ}}$ pin is driven high, the $\overline{\text{BACK}}$ pin is driven high at the prescribed timing and the external bus mastership released state is terminated.

In the event of simultaneous external bus mastership release request and external access request generation, the order of priority is as follows:

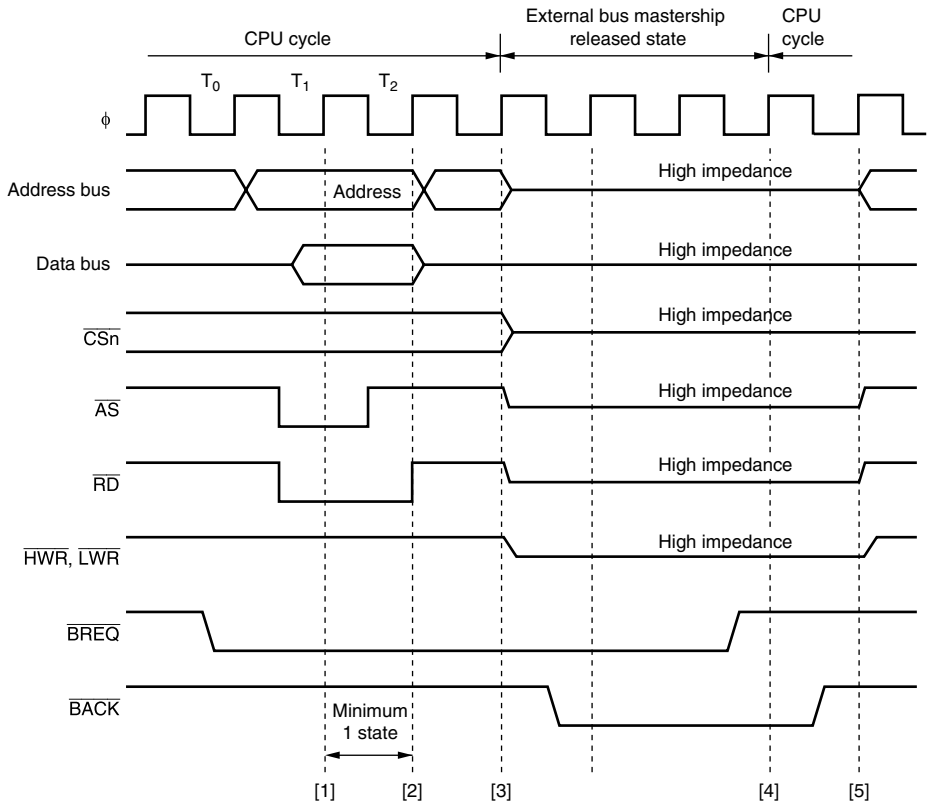
(High) External bus mastership release > Internal bus master external access (Low)

Table 7.5 shows the pin states in the external bus mastership released state.

Table 7.5 Pin States in Bus Mastership Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{CSn}}$	High impedance
$\overline{\text{AS}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{HWR}}$	High impedance
$\overline{\text{LWR}}$	High impedance

Figure 7.30 shows the timing for transition to the bus mastership released state.



[1] The low level of the $\overline{\text{BREQ}}$ pin is sampled at the rise of T_2 state.
[2] The $\overline{\text{BACK}}$ pin is driven low at one state after the end of CPU read cycle, releasing bus mastership to external bus master.
[3] The $\overline{\text{BREQ}}$ pin state is still sampled in external bus mastership released state.
[4] The high level of the $\overline{\text{BREQ}}$ pin is sampled.
[5] The $\overline{\text{BACK}}$ pin is driven high, ending the bus mastership release cycles.

Note : n = 0 to 7

Figure 7.30 Bus Mastership Released State Transition Timing

7.9.1 Usage Note for Bus Mastership Release

In the state where MSTPCR is set to H′FFFFFF, and transition to sleep mode is made, external bus mastership release function is aborted. When external bus mastership release function is used in sleep mode, MSTPCR should not be set to H′FFFFFF.

7.10 Bus Arbitration

This LSI has a bus arbiter that arbitrates bus master operations.

There are two bus masters, the CPU and DTC, which perform read/write operations when they have possession of the bus. Each bus master requests the bus mastership by means of a bus request signal. The bus arbiter determines priorities at the prescribed timing, and permits use of the bus by means of a bus mastership request acknowledge signal. The selected bus master then takes possession of the bus mastership and begins its operation.

7.10.1 Operation

The bus arbiter detects the bus masters' bus mastership request signals, and if the bus mastership is requested, sends a bus mastership request acknowledge signal to the bus master. If there are bus mastership requests from more than one bus master, the bus mastership request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus mastership request acknowledge signal, it takes possession of the bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DTC > CPU (Low)

An internal bus access by an internal bus master, and external bus mastership release, can be executed in parallel.

In the event of simultaneous external bus mastership release request, and internal bus master external access request generation, the order of priority is as follows:

(High) External bus mastership release > Internal bus master external access (Low)

7.10.2 Bus Mastership Transfer Timing

Even if a bus mastership request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus mastership is not necessarily transferred immediately. There are specific times at which each bus master can relinquish the bus mastership.

CPU: The CPU is the lowest-priority bus master, and if a bus mastership request is received from the DTC, the bus arbiter transfers the bus mastership to the bus master that issued the request. The timing for transfer of the bus mastership is as follows:

- The bus mastership is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus mastership is not transferred between the operations.

- If the CPU is in sleep mode, it transfers the bus mastership immediately.

DTC: The DTC sends the bus arbiter a request for the bus mastership when an activation request is generated.

The DTC can release the bus mastership after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus mastership during a register information read (3 states), a single data transfer, or a register information write (3 states).

7.10.3 Usage Note for External Bus Mastership Release

External bus mastership release can be performed on completion of an external bus cycle. The $\overline{CS}$ signal remains low until the end of the external bus cycle. Therefore, when external bus mastership release is performed, the $\overline{CS}$ signal may change from the low level to the high-impedance state.

7.11 Resets and the Bus Controller

In a power-on reset, this LSI, including the bus controller, enters the reset state at that point, and an executing bus cycle is discontinued.

In a manual reset, the bus controller's registers and internal state are maintained, and an executing external bus cycle is completed. In this case, $\overline{WAIT}$ input is ignored and write data is not guaranteed.

Section 8 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 8.1 shows a block diagram of the DTC.

The DTC's register information is stored in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. As 32-bit bus connects the DTC to on-chip RAM (1 kbyte), 32-bit/1-state reading and writing of the DTC register information is enabled.

8.1 Features

- Transfer is possible over any number of channels
  - One activation source can trigger a number of data transfers (chain transfer)
- Three transfer modes
  - Normal, repeat, and block transfer modes are available
- The direct specification of 16-Mbyte address space is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
- Activation by software is possible
- Module stop mode can be set

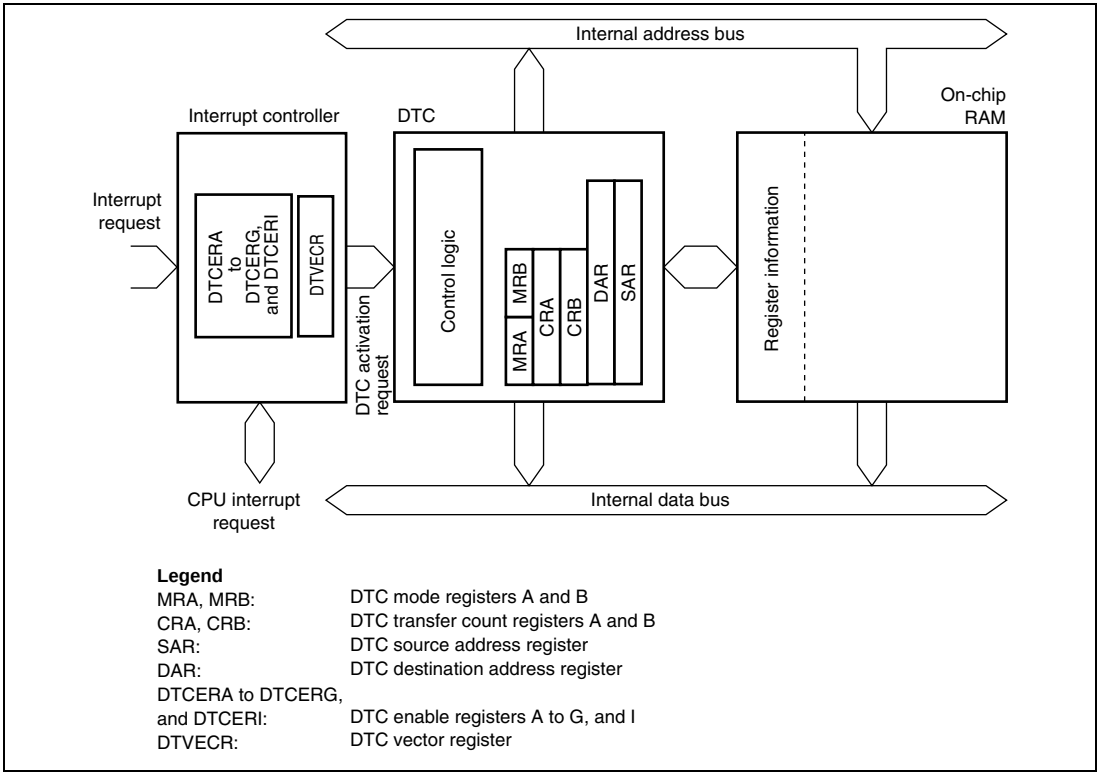


Figure 8.1 Block Diagram of DTC

8.2 Register Descriptions

The DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

These six registers cannot be directly accessed from the CPU. When activated, the DTC reads a set of register information that is stored in on-chip RAM and transfers data to the corresponding DTC registers. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable registers A to G, and I (DTCERA to DTCERG, and DTCERI)
- DTC vector register (DTVECR)

8.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1 and 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0X: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0X: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode 1 and 0
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: —
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area
0	Sz	Undefined	—	DTC Data Transfer Size Specifies the size of data to be transferred. 0: Byte-size transfer 1: Word-size transfer

Legend

X: Don't care

8.2.2 DTC Mode Register B (MRB)

MRB is an 8-bit register that selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable This bit specifies a chain transfer. For details, refer to section 8.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the interrupt source flag, and clearing of DTCER, are not performed. 0: DTC data transfer completed (waiting for start) 1: DTC data transfer (reads new register information and transfers data)
6	DISEL	Undefined	—	DTC Interrupt Select This bit specifies whether CPU interrupt is disabled or enabled after a data transfer. 0: Interrupt request is issued to the CPU when the specified data transfer is completed. (The DTC clears the interrupt request flag that causes the activation.) 1: The DTC issues interrupt request to the CPU in every data transfer. (The DTC does not clear the interrupt request flag that causes the activation.)
5 to 0	—	Undefined	—	Reserved These bits have no effect on the DTC operation. The write value should always be 0.

8.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

8.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

8.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times that data is transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts; the upper 8 bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00. These operations are repeated.

8.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times that data is transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

8.2.7 DTC Enable Registers A to G, and I (DTCERA to DTCERG, and DTCERI)

DTCER is a set of registers to specify the DTC activation interrupt source, and comprised of eight registers; DTCERA to DTCERG, and DTCERI. The correspondence between interrupt sources and DTCE bits, and vector numbers generated by the interrupt controller are shown in table 8.2. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. When multiple activation sources are to be set at one time, only at the initial setting, writing data is enabled after executing a dummy read on the relevant register with all the interrupts being masked.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCEn7	0	R/W	DTC Activation Enable
6	DTCEn6	0	R/W	1: Disables an interrupt for DTC activation.
5	DTCEn5	0	R/W	0: Specifies a relevant interrupt source as a DTC activation source.
4	DTCEn4	0	R/W	
3	DTCEn3	0	R/W	[Clearing conditions]
2	DTCEn2	0	R/W	• When the DISEL bit in MRB is 1 and the data transfer has ended
1	DTCEn1	0	R/W	• When the specified number of transfers have ended
0	DTCEn0	0	R/W	[Retaining condition]
				• When the DISEL bit is 0 and the specified number of transfers have not been completed

(n = A to G, and I)

8.2.8 DTC Vector Register (DTVECR)

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	DTC Software Activation Enable
				Enables or disables the DTC software activation.
				0: Disables the DTC software activation.
				1: Enables the DTC software activation.
				[Clearing conditions]
				• When the DISEL bit is 0 and the specified number of transfers have not ended
				• When 0 is written after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU
				[Retaining conditions]
				• When the DISEL bit is 1 and data transfer has ended
				• When the specified number of transfers have ended
				• When the software-activated data transfer is in process

Bit	Bit Name	Initial Value	R/W	Description
6	DTVEC6	0	R/W	DTC Software Activation Vector 6 to 0
5	DTVEC5	0	R/W	These bits specify a vector number for the DTC software activation.
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	These bits are writable when SWDTE = 0.
0	DTVEC0	0	R/W	

8.3 Activation Sources

The DTC operates when activated by an interrupt request or by a write to DTVECR by software. An activation interrupt request is specified by DTCER. When the corresponding bit is set to 1, it becomes DTC activation source and when it is cleared to 0, it becomes CPU interrupt source. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the interrupt flag of activation source or corresponding DTCER bit is cleared. Table 8.1 shows the relationship between the activation source and DTCER clearing. The activation source flag, in the case of RXI0, for example, is the RDRF flag in SCI_0.

When an interrupt has been designated as a DTC activation source, the existing CPU mask level and interrupt controller priorities have no effect. If there are more than one activation sources at the same time, the DTC operates in accordance with the default priority of the interrupt sources.

Figure 8.2 shows a block diagram of the DTC activation source control. For details, see section 5, Interrupt Controller.

Table 8.1 Activation Source and DTCER Clearing

Activation Source	The DISEL Bit is 0, and Transfer Counts Specified have not Ended	The DISEL Bit is 1, or Transfer Counts Specified have Ended
Software activation	• The SWDTE bit is cleared to 0	• The SWDTE bit retains 1 • The interrupt request is sent to the CPU
Interrupt activation	• The corresponding DTCER bit retains 1 • The activation source flag is cleared to 0	• The corresponding DTCER bit is cleared to 0 • The activation source flag retains 1 • The interrupt request which becomes an activation source is sent to the CPU

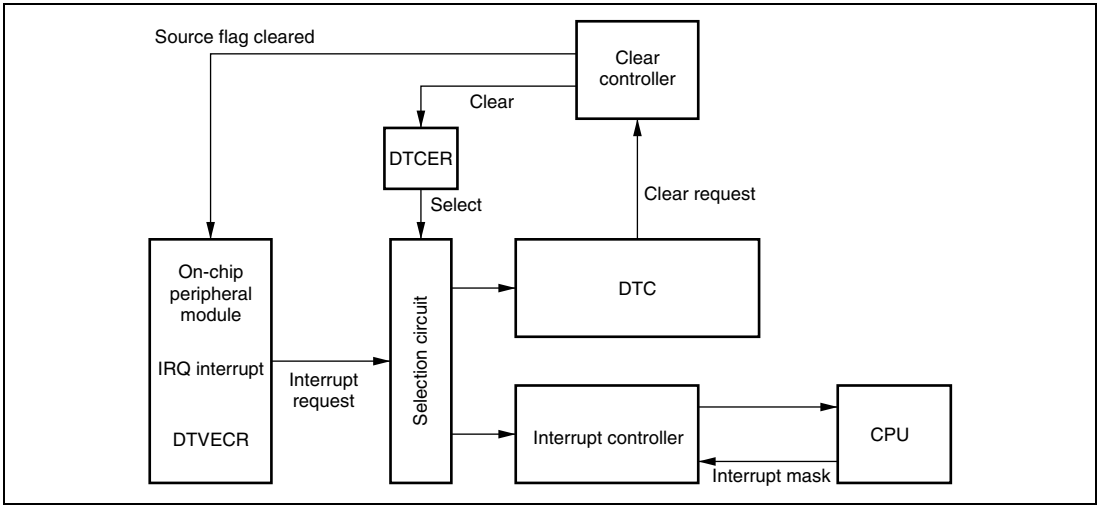


Figure 8.2 Block Diagram of DTC Activation Source Control

8.4 Location of Register Information and DTC Vector Table

Locate the register information in on-chip RAM (addresses: H'FFEBC0 to H'FFEFBF). Register information should be located at an address that is a multiple of four within the range. Locating the register information in address space is shown in figure 8.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information.

In the case of chain transfer, register information should be located in consecutive areas as shown in figure 8.3, and the register information start address should be located at the vector address corresponding to the interrupt source. Figure 8.4 shows the correspondence between the DTC vector address and register information. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420.

The configuration of the vector address is the same in both normal* and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.

Note: Normal mode is not supported in this LSI.

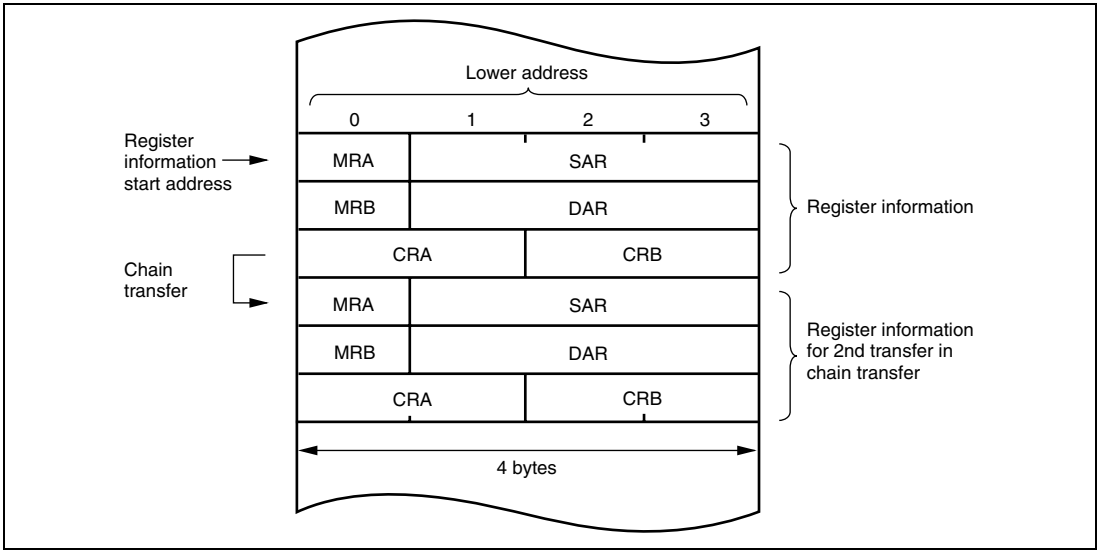


Figure 8.3 Location of DTC Register Information in Address Space

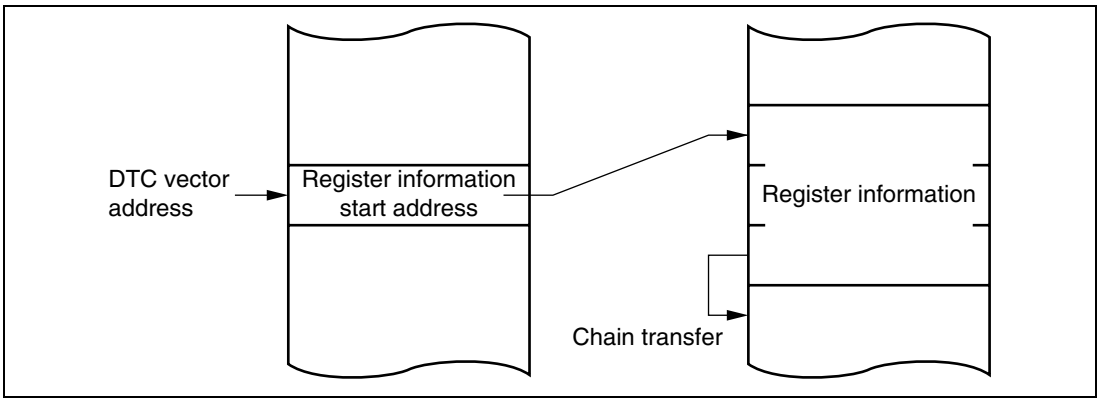


Figure 8.4 Correspondence between DTC Vector Address and Register Information

Table 8.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (vector number × 2)	—	High
External pin	IRQ0	16	H'0420	DTCEA7	
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
	IRQ4	20	H'0428	DTCEA3	
	IRQ5	21	H'042A	DTCEA2	
	IRQ6	22	H'042C	DTCEA1	
	IRQ7	23	H'042E	DTCEA0	
A/D converter	ADI (A/D conversion end)	28	H'0438	DTCEB6	
TPU channel 0	TGI0A	32	H'0440	DTCEB5	
	TGI0B	33	H'0442	DTCEB4	
	TGI0C	34	H'0444	DTCEB3	
	TGI0D	35	H'0446	DTCEB2	
TPU channel 1	TGI1A	40	H'0450	DTCEB1	
	TGI1B	41	H'0452	DTCEB0	
TPU channel 2	TGI2A	44	H'0458	DTCEC7	
	TGI2B	45	H'045A	DTCEC6	
TPU channel 3	TGI3A	48	H'0460	DTCEC5	
	TGI3B	49	H'0462	DTCEC4	
	TGI3C	50	H'0464	DTCEC3	
	TGI3D	51	H'0466	DTCEC2	
TPU channel 4	TGI4A	56	H'0470	DTCEC1	
	TGI4B	57	H'0472	DTCEC0	
TPU channel 5	TGI5A	60	H'0478	DTCED5	
	TGI5B	61	H'047A	DTCED4	
8-bit timer channel 0	CMIA0	64	H'0480	DTCED3	
	CMIB0	65	H'0482	DTCED2	
8-bit timer channel 1	CMIA1	68	H'0488	DTCED1	
	CMIB1	69	H'048A	DTCED0	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	DTC Vector Address	DTCE*	Priority
SCI channel 0	RXI0	81	H'04A2	DTCEE3	High
	TXI0	82	H'04A4	DTCEE2	
SCI channel 1	RXI1	85	H'04AA	DTCEE1	
	TXI1	86	H'04AC	DTCEE0	
SCI channel 2	RXI2	89	H'04B2	DTCEF7	
	TXI2	90	H'04B4	DTCEF6	
8-bit timer channel 2	CMIA2	92	H'04B8	DTCEF5	
	CMIB2	93	H'04BA	DTCEF4	
8-bit timer channel 3	CMIA3	96	H'04C0	DTCEF3	
	CMIB3	97	H'04C2	DTCEF2	
IEB (H8S/2552 Series only)	IERxI	105	H'04D2	DTCEG6	
	IETxI	106	H'04D4	DTCEG5	
HCAN (H8S/2556 Series only)	RM0	109	H'04DA	DTCEG2	
SCI channel 3	RXI3	121	H'04F2	DTCEI7	
	TXI3	122	H'04F4	DTCEI6	
SCI channel 4	RXI4	125	H'04FA	DTCEI5	
	TXI4	126	H'04FC	DTCEI4	Low

Note: * The DTCE bits with no corresponding interrupt are reserved, and should be written with 0.

8.5 Operation

Register information is stored in on-chip RAM. When activated, the DTC reads register information in on-chip RAM and transfers data. After the data transfer, the DTC writes updated register information back to on-chip RAM.

The pre-storage of register information in on-chip RAM makes it possible to transfer data over any required number of channels. The transfer mode can be specified as normal, repeat, or block transfer mode. Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers with a single activation source (chain transfer).

The 24-bit SAR designates the DTC transfer source address, and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed depending on its register information.

Figure 8.5 shows the flowchart of DTC operation.

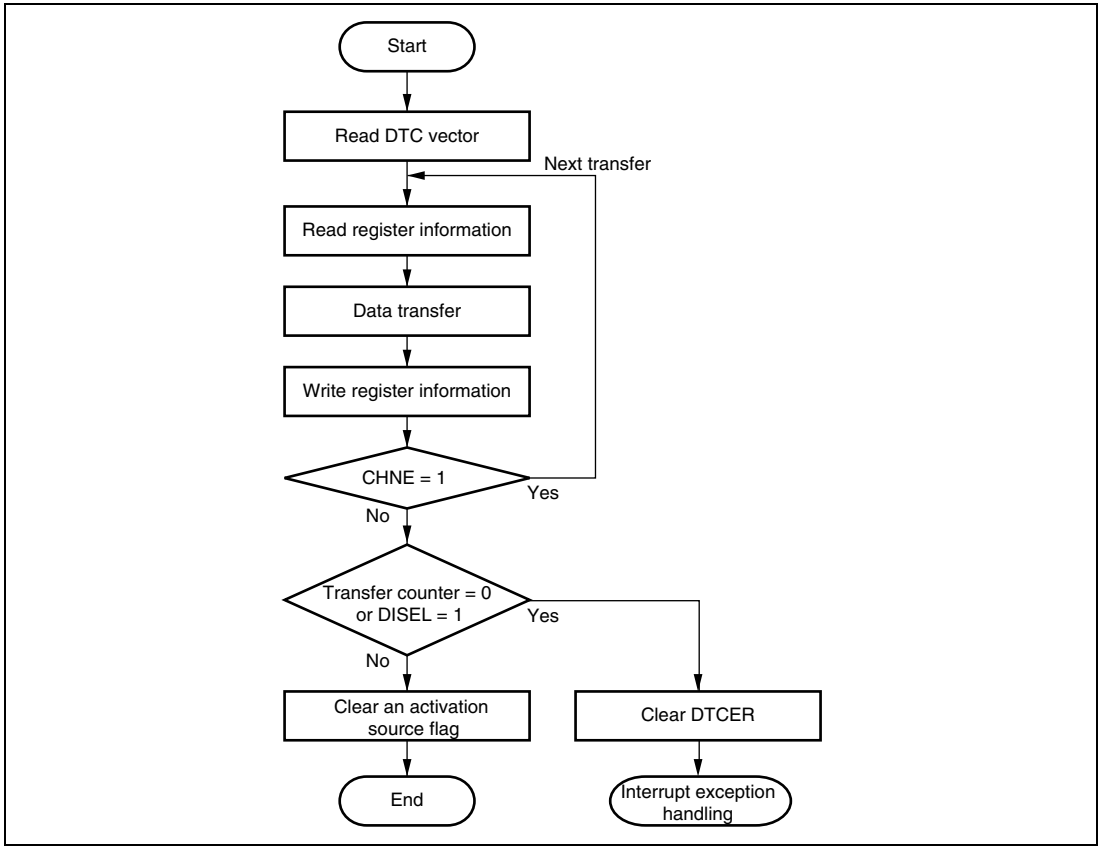


Figure 8.5 Flowchart of DTC Operation

8.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have been completed, a CPU interrupt can be requested.

Table 8.3 lists the register function in normal mode. Figure 8.6 shows the memory mapping in normal mode.

Table 8.3 Register Function in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates transfer source address
DTC destination address register	DAR	Designates transfer destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used

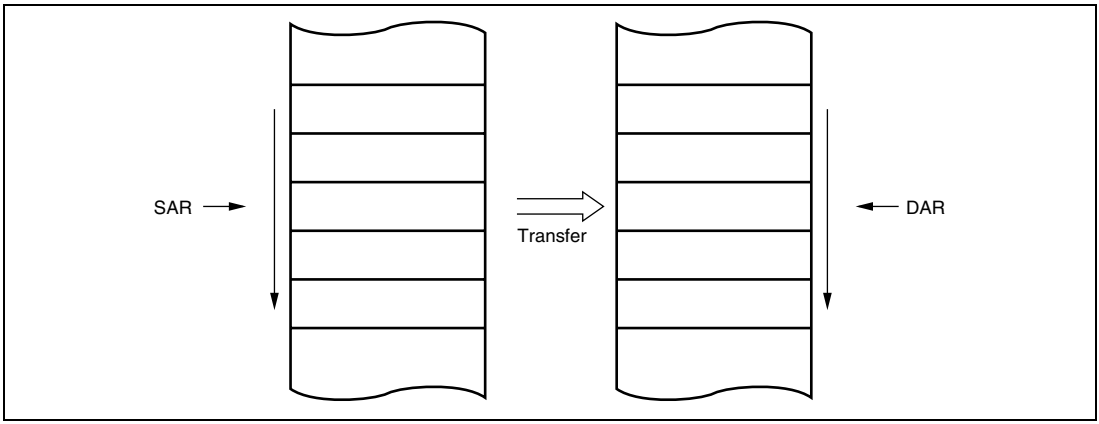


Figure 8.6 Memory Mapping in Normal Mode

8.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode, the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 8.4 lists the register function in repeat mode. Figure 8.7 shows the memory mapping in repeat mode.

Table 8.4 Register Function in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates transfer source address
DTC destination address register	DAR	Designates transfer destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used

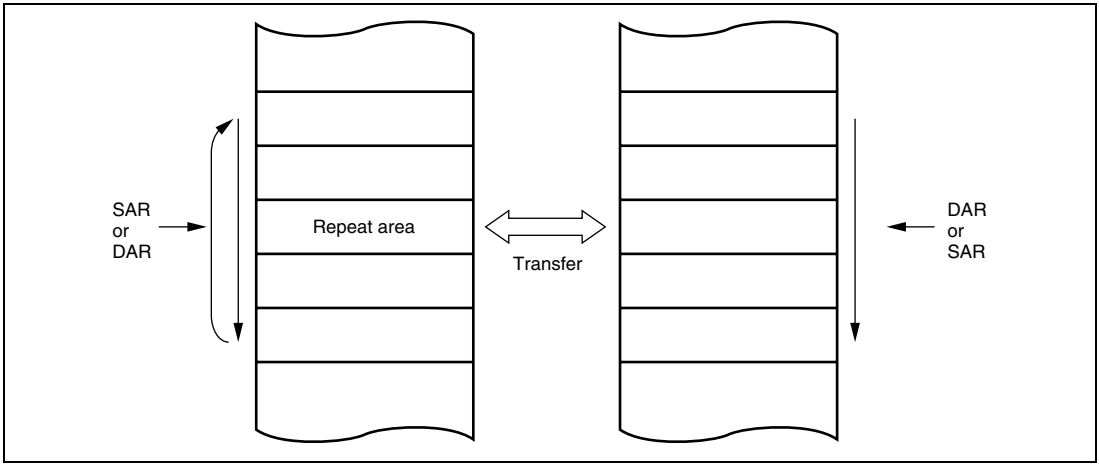


Figure 8.7 Memory Mapping in Repeat Mode

8.5.3 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area.

The block size can be between 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed depending on its register information.

From 1 to 65,536 transfers can be specified. Once the specified numbers of transfers have been completed, a CPU interrupt is requested.

Table 8.5 lists the register function in block transfer mode. Figure 8.8 shows the memory mapping in block transfer mode.

Table 8.5 Register Function in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates transfer source address
DTC destination address register	DAR	Designates transfer destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Designates transfer count

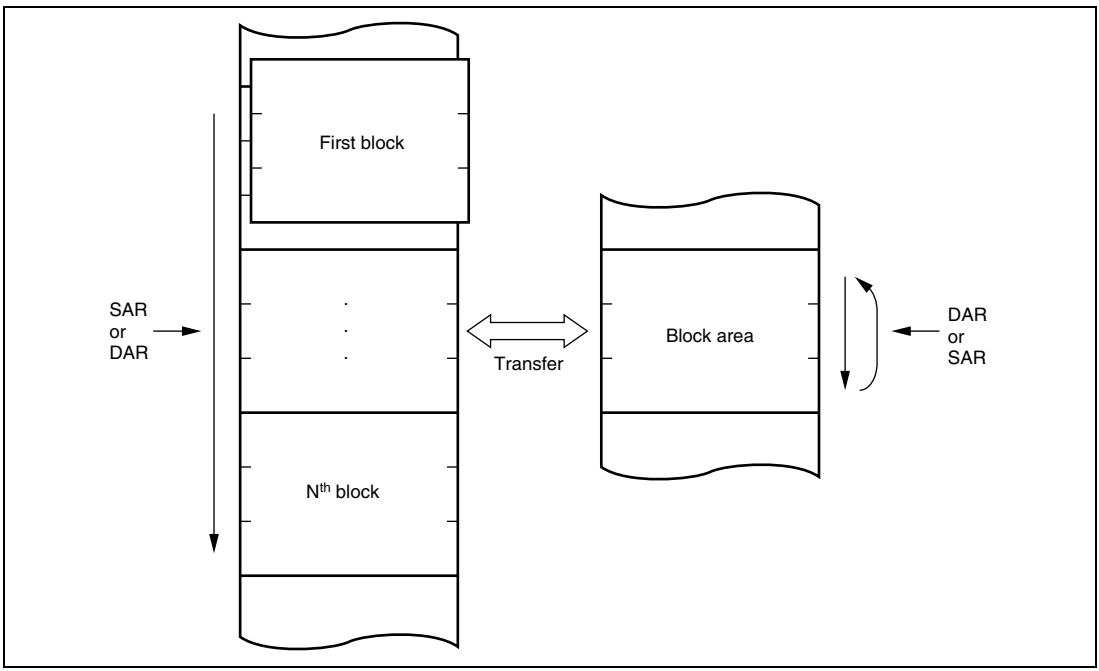


Figure 8.8 Memory Mapping in Block Transfer Mode

8.5.4 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 8.9 shows the memory map for chain transfer. When activated, the DTC reads the register information start address stored at the vector address, which corresponds to the activation request, and then reads the first register information at that start address. After the data transfer, the CHNE bit will be tested. When it has been set to 1, the DTC reads the next register information located in

a consecutive area and performs the data transfer. These sequences are repeated until the CHNE bit is cleared to 0.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt request flag for the activation source is not affected.

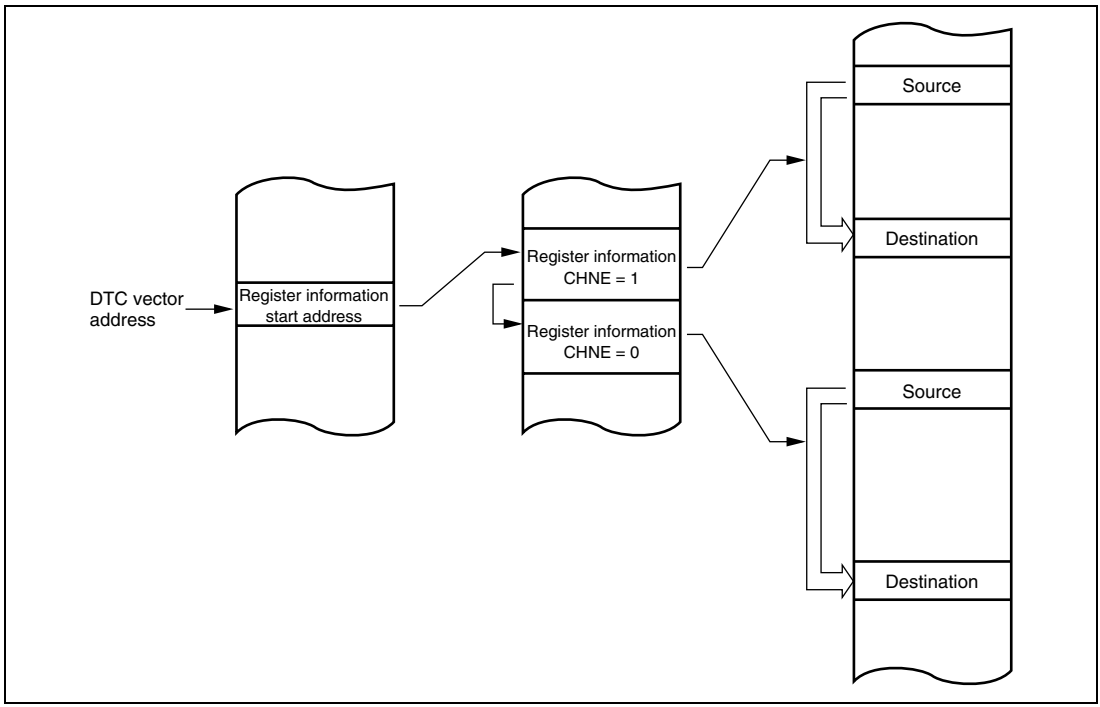


Figure 8.9 Chain Transfer Operation

8.5.5 Interrupts

An interrupt request is issued to the CPU when the DTC has completed the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of software activation, a software-activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has been completed, or the specified number of transfers have been completed, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated after the end of data transfer. The interrupt handling routine will then clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

8.5.6 Operation Timing

Figures 8.10 to 8.12 show the DTC operation timing.

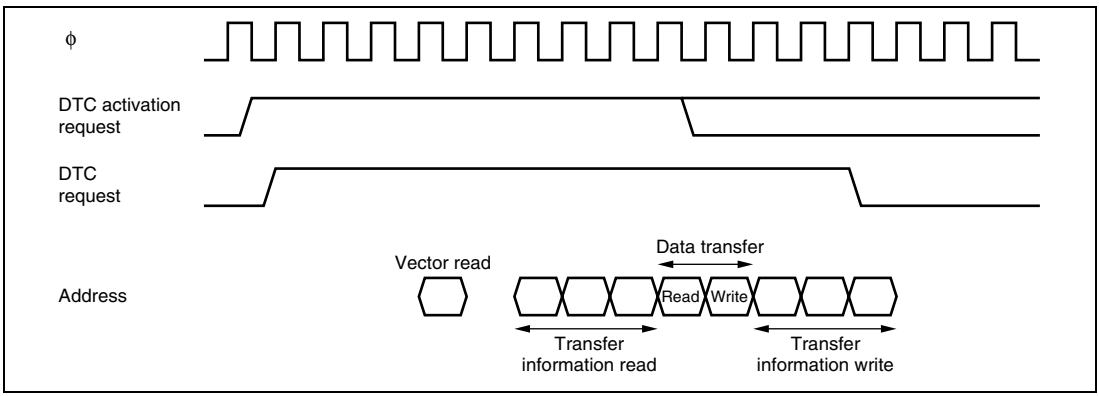


Figure 8.10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

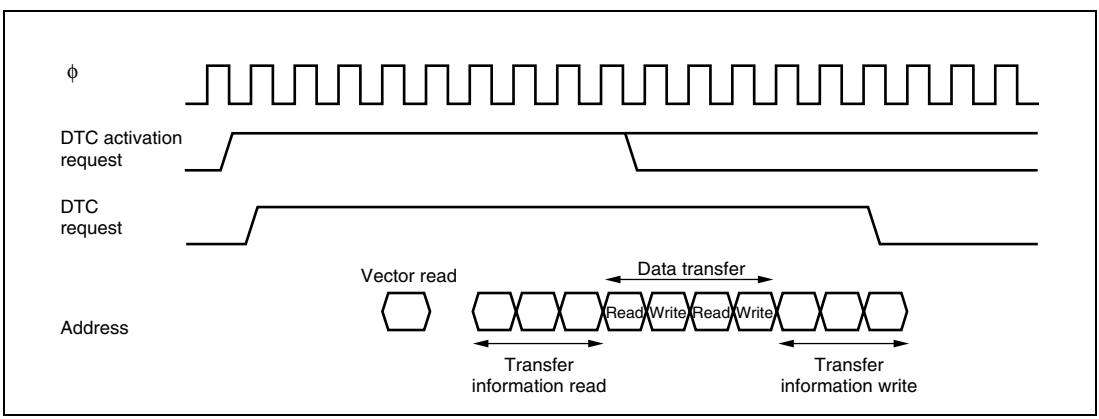


Figure 8.11 DTC Operation Timing (Example in Block Transfer Mode, with Block Size of 2)

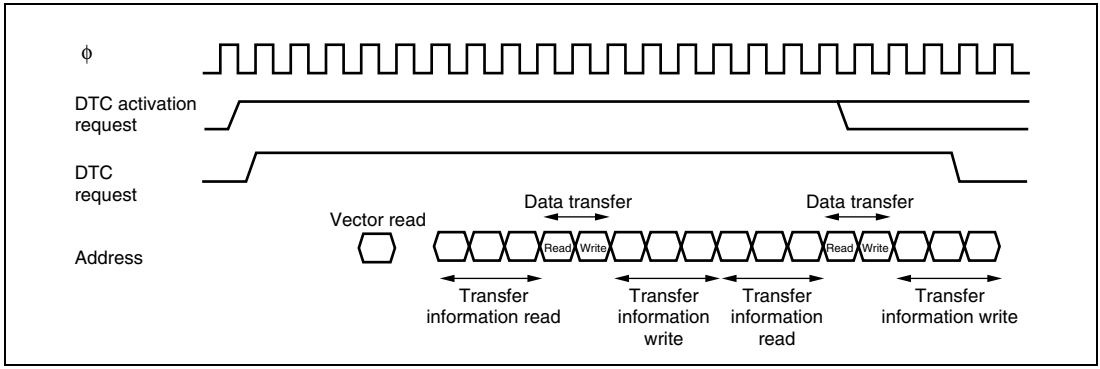


Figure 8.12 DTC Operation Timing (Example of Chain Transfer)

8.5.7 Number of DTC Execution States

Table 8.6 lists execution status for a single DTC data transfer, and table 8.7 lists the number of states required for each execution status.

Table 8.6 DTC Execution Status

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

Legend

N: Block size (initial setting of CRAH and CRAL)

Table 8.7 Number of States Required for Each Execution Status

Object to be Accessed		On-Chip RAM	On-Chip ROM	Internal I/O Registers				External Devices			
				IEB*[1]	HCAN*[2]	Other than IEB and HCAN					
Bus width		32	16	8	16	8	16	8	8	16	16
Access states		1	1	5	5	2	2	2	3	2	3
Execution Status	Vector read S_I	—	1	—	—	—	—	4	6 + 2m	2	3 + m
	Register information read/write S_J	1	—	—	—	—	—	—	—	—	—
	Byte data read S_K	1	1	5	5	2	2	2	3 + m	2	3 + m
	Word data read S_K	1	1	—	5	4	2	4	6 + 2m	2	3 + m
	Byte data write S_L	1	1	5	5	2	2	2	3 + m	2	3 + m
	Word data write S_L	1	1	—	5	4	2	4	6 + 2m	2	3 + m
	Internal operation S_M	1	1	1	1	1	1	1	1	1	1

Legend

m: The number of wait states for accessing external devices.

Notes: 1. H8S/2552 Series only.

2. H8S/2556 Series only.

The number of execution states is calculated from using the formula below. Note that Σ is the sum of all transfers activated by one activation source (the number in which the CHNE bit is set to 1, plus 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address is located in on-chip ROM, normal mode is set, and data is transferred from on-chip ROM to an internal I/O register, the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

8.6 Procedures for Using DTC

8.6.1 Activation by Interrupt

The procedure for using the DTC with interrupt activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After one data transfer has been completed, or after the specified number of data transfers has been completed, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

8.6.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After one data transfer has been completed, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers has been completed, the SWDTE bit is held at 1 and a CPU interrupt is requested.

8.7 Examples of Use of the DTC

8.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. MRA sets the source address fixed (SM1 = SM0 = 0), destination address increment (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can be set to any value. MRB performs one data transfer by one interrupt (CHNE = 0, DISEL = 0). SAR sets the RDR address in SCI, DAR sets the start address of the RAM area where the data will be received in, and CRA sets 128 (H'0080). CRB can be set to any value.
2. Set the start address of the register information in the DTC vector address.

3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time the reception of one byte of data has been completed on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have been completed, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine will perform wrap-up processing.

8.7.2 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the transfer destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. MRA sets the source address increment (SM1 = 1, SM0 = 0), destination address increment (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can be set to any value. MRB performs one block transfer by one interrupt (CHNE = 0). SAR sets the transfer source address (H'1000), DAR sets the transfer destination address (H'2000), and CRA sets 128 (H'8080). CRB sets 1 (H'0001).
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write has failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

8.8 Usage Notes

8.8.1 Module Stop Mode Setting

The DTC operation can be disabled or enabled using the module stop control register. The initial setting is for the DTC operation to be enabled. Register access is disabled by setting module stop mode. Module stop mode cannot be set during the DTC operation. For details, refer to section 22, Power-Down Modes.

8.8.2 On-Chip RAM

The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR should not be cleared to 0.

8.8.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. When multiple activation sources are to be set at one time, only at the initial setting, writing data is enabled after executing a dummy read on the relevant register with all the interrupts being masked.

Section 9 I/O Ports

Table 9.1 summarizes the port functions. The pins of each port also have other functions such as input/output or interrupt input pins of on-chip peripheral modules. Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have DDR and DR registers.

Ports A to E have a built-in input pull-up MOS function and an input pull-up MOS control register (PCR) to control the on/off state of the input pull-up MOS.

Ports 3 and A include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

All the I/O ports can drive a single TTL load and a 30 pF capacitive load.

The P34 and P35 pins on port 3 are NMOS push pull outputs.

The $\overline{\text{IRQ}}$ pin is a schmitt trigger input.

Table 9.1 Port Functions

Port	Description	Mode 6	Mode 7	Input/Output and Output Type
Port 1	General I/O port also functioning as TPU I/O pins and interrupt input pins	P17/TIOCB2/TCLKD P16/TIOCA2/$\overline{\text{IRQ1}}$ P15/TIOCB1/TCLKC P14/TIOCA1/$\overline{\text{IRQ0}}$ P13/TIOCD0/TCLKB P12/TIOCC0/TCLKA P11/TIOCB0 P10/TIOCA0		Schmitt trigger input ($\overline{\text{IRQ1}}$, $\overline{\text{IRQ0}}$)
Port 2	General I/O port also functioning as TPU I/O pins	P27/TIOCB5 P26/TIOCA5 P25/TIOCB4 P24/TIOCA4 P23/TIOCD3 P22/TIOCC3 P21/TIOCB3 P20/TIOCA3		
Port 3	General I/O port also functioning as I^2C bus interface 2 I/O pins, SCI I/O pins, and interrupt input pins	P37/TxD4 P36/RxD4 P35/SCK1/SCK4/SCL0/$\overline{\text{IRQ5}}$ P34/RxD1/SDA0 P33/TxD1/SCL1 P32/SCK0/SDA1/$\overline{\text{IRQ4}}$ P31/RxD0 P30/TxD0		Open drain output enabled Schmitt trigger input ($\overline{\text{IRQ5}}$, $\overline{\text{IRQ4}}$)
Port 4	General input port also functioning as A/D converter analog input pins	P47/AN7 P46/AN6 P45/AN5 P44/AN4 P43/AN3 P42/AN2 P41/AN1 P40/AN0		

Port	Description	Mode 6	Mode 7	Input/Output and Output Type
Port 5	General I/O port also functioning as SCI I/O pins	P52/SCK2		
		P51/RxD2		
		P50/TxD2		
Port 7	General I/O port also functioning as SCI I/O pins, TMR I/O pins, bus control output pins, and manual reset input pins	P77/TxD3		
		P76/RxD3		
		P75/TMO3/SCK3		
		P74/TMO2/$\overline{\text{MRES}}$		
		P73/TMO1/$\overline{\text{CS7}}$	P73/TMO1	
		P72/TMO0/$\overline{\text{CS6}}$	P72/TMO0	
		P71/TMRI23/TMCI23/$\overline{\text{CS5}}$	P71/TMRI23/TMCI23	
		P70/TMRI01/TMCI01/$\overline{\text{CS4}}$	P70/TMRI0/TMCI01	
Port 9	General input port also functioning as A/D converter analog input and D/A converter analog output pins	P97/AN15/DA1		
		P96/AN14/DA0		
		P95/AN13		
		P94/AN12		
		P93/AN11		
		P92/AN10		
		P91/AN9		
		P90/AN8		
Port A	General I/O port also functioning as address output pins	PA7/A23	PA7	Built-in input pull-up MOS Open drain output enabled
		PA6/A22	PA6	
		PA5/A21	PA5	
		PA4/A20	PA4	
		PA3/A19	PA3	
		PA2/A18	PA2	
		PA1/A17	PA1	
		PA0/A16	PA0	
Port B	General I/O port also functioning as address output pins	PB7/A15	PB7	Built-in input pull-up MOS
		PB6/A14	PB6	
		PB5/A13	PB5	
		PB4/A12	PB4	
		PB3/A11	PB3	
		PB2/A10	PB2	
		PB1/A9	PB1	
		PB0/A8	PB0	

Port	Description	Mode 6	Mode 7	Input/Output and Output Type
Port C	General I/O port also functioning as address output pins	PC7/A7	PC7	Built-in input pull-up MOS
		PC6/A6	PC6	
		PC5/A5	PC5	
		PC4/A4	PC4	
		PC3/A3	PC3	
		PC2/A2	PC2	
		PC1/A1	PC1	
		PC0/A0	PC0	
Port D	General I/O port also functioning as data I/O pins	D15	PD7	Built-in input pull-up MOS
		D14	PD6	
		D13	PD5	
		D12	PD4	
		D11	PD3	
		D10	PD2	
		D9	PD1	
		D8	PD0	
Port E	General I/O port also functioning as data I/O pins	PE7/D7	PE7	Built-in input pull-up MOS
		PE6/D6	PE6	
		PE5/D5	PE5	
		PE4/D4	PE4	
		PE3/D3	PE3	
		PE2/D2	PE2	
		PE1/D1	PE1	
		PE0/D0	PE0	
Port F	General I/O port also functioning as system clock output pins, interrupt input pins, bus control I/O pins, A/D converter input pins, and BUZZ output pins	PF7/ϕ	PF7/ϕ	Schmitt trigger input ($\overline{\text{IRQ3}}$, $\overline{\text{IRQ2}}$)
		$\overline{\text{AS}}$	PF6	
		$\overline{\text{RD}}$	PF5	
		$\overline{\text{HWR}}$	PF4	
		PF3/$\overline{\text{LWR}}$/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	
		PF2/$\overline{\text{WAIT}}$	PF2	
		PF1/$\overline{\text{BACK}}$/BUZZ	PF1/BUZZ	
		PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	

Port	Description	Mode 6	Mode 7	Input/Output and Output Type
Port G	General I/O port also functioning as bus control output pins, interrupt input pins, and IEB I/O pins*1	PG4/$\overline{CS0}$	PG4	Schmitt trigger input ($\overline{IRQ7}$, $\overline{IRQ6}$)
		PG3/$\overline{Rx}$*1/$\overline{CS1}$*2	PG3/$\overline{Rx}$*1*2	
		PG2/$\overline{Tx}$*1/$\overline{CS2}$*2	PG2/$\overline{Tx}$*1*2	
		PG1/$\overline{CS3}$/$\overline{IRQ7}$	PG1/$\overline{IRQ7}$	
		PG0/$\overline{IRQ6}$	PG0/$\overline{IRQ6}$	
Port H	General I/O port	PH7		
		PH6		
		PH5		
		PH4		
		PH3		
		PH2		
		PH1		
		PH0		
Port J	General I/O port	PJ7		
		PJ6		
		PJ5		
		PJ4		
		PJ3		
		PJ2		
		PJ1		
		PJ0		

Notes: 1. The $\overline{Rx}$ and $\overline{Tx}$ of IEB are valid only in the H8S/2552.
2. The PG3/$\overline{Rx}$/$\overline{CS1}$ and PG2/$\overline{Tx}$/$\overline{CS2}$ pins are not available in the H8S/2556.

9.1 Port 1

Port 1 is an 8-bit I/O port and has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)

9.1.1 Port 1 Data Direction Register (P1DDR)

P1DDR specifies input or output of the port 1 pins using the individual bits. P1DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

9.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

9.1.3 Port 1 Register (PORT1)

PORT1 shows port 1 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.
6	P16	—*	R	
5	P15	—*	R	
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: * Determined by the states of pins P17 to P10.

9.1.4 Pin Functions

Port 1 pins also function as TPU I/O pins and interrupt input pins. Port 1 pin functions are shown below.

- P17/TIOCB2/TCLKD

  The pin function is switched as shown below according to the combination of the TPU channel 2 setting, the TPSC2 to TPSC0 bits in TCR_0 or TCR_5, and the P17DDR bit.

TPU Channel 2 Setting*[1]	Output	Input or Initial Value	
P17DDR	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output
		TIOCB2 input*[2]	
	TCLKD input*[3]		

Notes 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCB2 input when TPU channel 2 timer operating mode is set to normal operation or phase counting mode and IOB3 in TIOR_2 is set to 1.
3. This pin functions as TCLKD input when TPSC2 to TPSC0 in TCR_0 or TCR_5 are set to 111. This pin also functions as TCLKD input when channel 2 or 4 is set to phase counting mode.

- P16/TIOCA2/$\overline{\text{IRQ1}}$

  The pin function is switched as shown below according to the combination of the TPU channel 2 setting and the P16DDR bit.

TPU Channel 2 Setting*[1]	Output	Input or Initial Value	
P16DDR	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output
		TIOCA2 input*[2]	
	$\overline{\text{IRQ1}}$ input*[3]		

Notes 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCA2 input when TPU channel 2 timer operating mode is set to normal operation or phase counting mode and IOA3 in TIOR_2 is set to 1.
3. When this pin is used as an external interrupt pin, do not specify other functions.

- P15/TIOCB1/TCLKC

  The pin function is switched as shown below according to the combination of the TPU channel 1 setting, the TPSC2 to TPSC0 bits in TCR_0, TCR_2, TCR_4, or TCR_5, and the P15DDR bit.

TPU Channel 1 Setting*[1]	Output	Input or Initial Value	
P15DDR	—	0	1
Pin function	TIOCB1 output	P15 input	P15 output
		TIOCB1 input*[1]	
	TCLKC input*[2]		

Notes 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCB1 input when TPU channel 1 timer operating mode is set to normal operation or phase counting mode and IOB3 to IOB0 in TIOR_1 are set to10xx.
3. This pin functions as TCLKC input when TPSC2 to TPSC0 in TCR_0 or TCR_2 are set to 110, or when TPSC2 to TPSC0 in TCR_4 or TCR_5 are set to 101. This pin also functions as TCLKC input when channel 2 or 4 is set to phase counting mode.

- P14/TIOCA1/$\overline{\text{IRQ0}}$

  The pin function is switched as shown below according to the combination of the TPU channel 1 setting and the P14DDR bit.

TPU Channel 1 Setting*[1]	Output	Input or Initial Value	
P14DDR	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output
		TIOCA1 input*[2]	
	$\overline{\text{IRQ0}}$ input*[3]		

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCA1 input when TPU channel 1 timer operating mode is set to normal operation or phase counting mode and IOA3 to IOA0 in TIOR_1 are set to 10xx.
3. When this pin is used as an external interrupt pin, do not specify other functions.

- P13/TIOCD0/TCLKB

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting, the TPSC2 to TPSC0 bits in TCR_0, TCR_1, or TCR_2, and the P13DDR bit.

TPU Channel 0 Setting*[1]	Output	Input or Initial Value	
P13DDR	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output
		TIOCD0 input*[2]	
	TCLKB input*[3]		

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCD0 input when TPU channel 0 timer operating mode is set to normal operation and IOD3 to IOD0 in TIORL_0 are set to 10xx.
3. This pin functions as TCLKB input when TPSC2 to TPSC0 are set to 101 in any of TCR_0, TCR_1, and TCR_2. This pin also functions as TCLKB input when channel 1 or 5 is set to phase counting mode.

- P12/TIOCC0/TCLKA

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting, the TPSC2 to TPSC0 bits in TCR_0, TCR_1, TCR_2, TCR_3, TCR_4, or TCR_5, and the P12DDR bit.

TPU Channel 0 Setting*[1]	Output	Input or Initial Value	
P12DDR	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output
		TIOCC0 input*[2]	
	TCLKA input*[3]		

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCC0 input when TPU channel 0 timer operating mode is set to normal operation and IOC3 to IOC0 in TIORL_0 are set to 10xx.
3. This pin functions as TCLKA input when TPSC2 to TPSC0 are set to 100 in any of TCR_0, TCR_1, TCR_2, TCR_3, TCR_4, and TCR_5. This pin also functions as TCLKA input when channel 1 or 5 is set to phase counting mode.

- P11/TIOCB0

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting and the P11DDR bit.

TPU Channel 0 Setting*[1]	Output	Input or Initial Value	
P11DDR	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output
		TIOCB0 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCB0 input when TPU channel 0 timer operating mode is set to normal operation and IOB3 to IOB0 in TIORH_0 are set to 10xx.

- P10/TIOCA0

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting and the P10DDR bit.

TPU Channel 0 Setting*[1]	Output	Input or Initial Value	
P10DDR	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output
		TIOCA0 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCA0 input when TPU channel 0 timer operating mode is set to normal operation and IOA3 to IOA0 in TIORH_0 are set to 10xx.

9.2 Port 2

Port 2 is an 8-bit I/O port and has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 register (PORT2)

9.2.1 Port 2 Data Direction Register (P2DDR)

P2DDR specifies input or output of the port 2 pins using the individual bits. P2DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 2 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

9.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	P26DR	0	R/W	
5	P25DR	0	R/W	
4	P24DR	0	R/W	
3	P23DR	0	R/W	
2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	

9.2.3 Port 2 Register (PORT2)

PORT2 shows port 2 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P27	—*	R	If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read. If a port 2 read is performed while P2DDR bits are cleared to 0, the pin states are read.
6	P26	—*	R	
5	P25	—*	R	
4	P24	—*	R	
3	P23	—*	R	
2	P22	—*	R	
1	P21	—*	R	
0	P20	—*	R	

Note: * Determined by the states of pins P27 to P20.

9.2.4 Pin Functions

Port 2 pins also function as TPU I/O pins. Port 2 pin functions are shown below.

- P27/TIOCB5

  The pin function is switched as shown below according to the combination of the TPU channel 5 setting and the P27DDR bit.

TPU Channel 5 Setting*[1]	Output	Input or Initial Value	
P27DDR	—	0	1
Pin function	TIOCB5 output	P27 input	P27 output
		TIOCB5 input*[2]	

Notes 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).

2. This pin functions as TIOCB5 input when TPU channel 5 timer operating mode is set to normal operation or phase counting mode and IOB3 in TIOR_5 is set to 1.

- P26/TIOCA5

  The pin function is switched as shown below according to the combination of the TPU channel 5 setting and the P26DDR bit.

TPU Channel 5 Setting*1	Output	Input or Initial Value	
P26DDR	—	0	1
Pin function	TIOCA5 output	P26 input	P26 output
		TIOCA5 input*2	

Notes 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCA5 input when TPU channel 5 timer operating mode is set to normal operation or phase counting mode and IOA3 in TIOR_5 is set to 1.

- P25/TIOCB4

  The pin function is switched as shown below according to the combination of the TPU channel 4 setting and the P25DDR bit.

TPU Channel 4 Setting*1	Output	Input or Initial Value	
P25DDR	—	0	1
Pin function	TIOCB4 output	P25 input	P25 output
		TIOCB4 input*1	

Notes 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCB4 input when TPU channel 4 timer operating mode is set to normal operation or phase counting mode and IOB3 to IOB0 in TIOR_4 are set to10xx.

- P24/TIOCA4

  The pin function is switched as shown below according to the combination of the TPU channel 4 setting and the P24DDR bit.

TPU Channel 4 Setting*1	Output	Input or Initial Value	
P24DDR	—	0	1
Pin function	TIOCA4 output	P24 input	P24 output
		TIOCA4 input*2	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCA4 input when TPU channel 4 timer operating mode is set to normal operation or phase counting mode and IOA3 to IOA0 in TIOR_4 are set to 10xx.

- P23/TIOCD3

  The pin function is switched as shown below according to the combination of the TPU channel 3 setting and the P23DDR bit.

TPU Channel 3 Setting*[1]	Output	Input or Initial Value	
P23DDR	—	0	1
Pin function	TIOCD3 output	P23 input	P23 output
		TIOCD3 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCD3 input when TPU channel 3 timer operating mode is set to normal operation and IOD3 to IOD0 in TIORL_3 are set to 10xx.

- P22/TIOCC3

  The pin function is switched as shown below according to the combination of the operating mode, the TPU channel 3 setting, and the P22DDR bit.

TPU Channel 3 Setting*[1]	Output	Input or Initial Value	
P22DDR	—	0	1
Pin function	TIOCC0 output	P22 input	P22 output
		TIOCC3 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCC3 input when TPU channel 3 timer operating mode is set to normal operation and IOC3 to IOC0 in TIORL_3 are set to 10xx.

- P21/TIOCB3

  The pin function is switched as shown below according to the combination of the operating mode, the TPU channel 3 setting, and the P21DDR bit.

TPU Channel 3 Setting*[1]	Output	Input or Initial Value	
P21DDR	—	0	1
Pin function	TIOCB3 output	P21 input	P21 output
		TIOCB3 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCB3 input when TPU channel 3 timer operating mode is set to normal operation and IOB3 to IOB0 in TIORH_3 are set to 10xx.

- P20/TIOCA3

  The pin function is switched as shown below according to the combination of the operating mode, the TPU channel 3 setting, and the P20DDR bit.

TPU Channel 3 Setting*[1]	Output	Input or Initial Value	
P20DDR	—	0	1
Pin function	TIOCA3 output	P20 input	P20 output
		TIOCA3 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TPU).
2. This pin functions as TIOCA3 input when TPU channel 3 timer operating mode is set to normal operation and IOA3 to IOA0 in TIORH_3 are set to 10xx.

9.3 Port 3

Port 3 is an 8-bit I/O port and has the following registers.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 register (PORT3)
- Port 3 open drain control register (P3ODR)

9.3.1 Port 3 Data Direction Register (P3DDR)

P3DDR specifies input or output of the port 3 pins using the individual bits. P3DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 3 pin an output port. Clearing this bit to 0 makes the pin an input port.
6	P36DDR	0	W	
5	P35DDR	0	W	
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

9.3.2 Port 3 Data Register (P3DR)

P3DR stores output data for port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	P36DR	0	R/W	
5	P35DR	0	R/W	
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

9.3.3 Port 3 Register (PORT3)

PORT3 shows port 3 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P37	—*	R	If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read. If a port 3 read is performed while P3DDR bits are cleared to 0, the pin states are read.
6	P36	—*	R	
5	P35	—*	R	
4	P34	—*	R	
3	P33	—*	R	
2	P32	—*	R	
1	P31	—*	R	
0	P30	—*	R	

Note: * Determined by the states of pins P37 to P30.

9.3.4 Port 3 Open Drain Control Register (P3ODR)

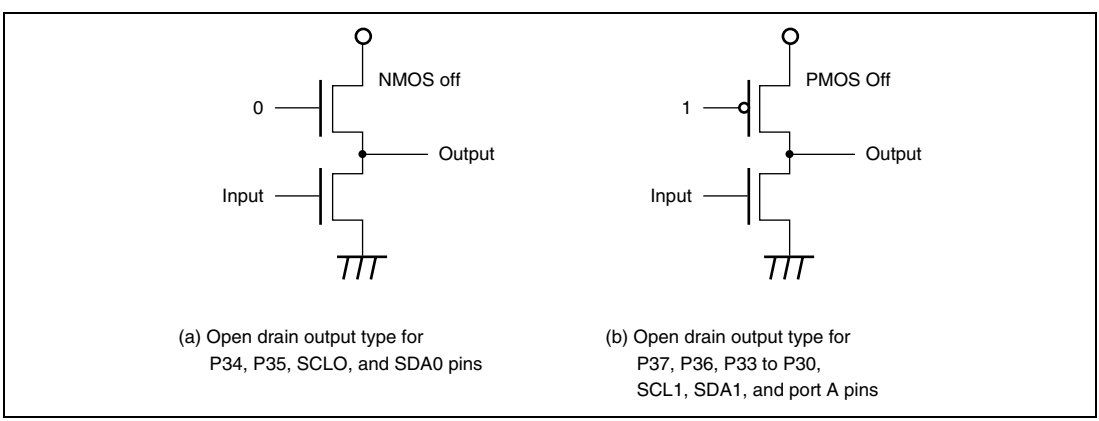
P3ODR controls on/off state of the PMOS for port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37ODR	0	R/W	When each of P37ODR, P36ODR, and P33ODR to P30ODR bits is set to 1, the corresponding pins P37, P36, and P33 to P30 function as NMOS open drain outputs. When cleared to 0, the corresponding pins function as CMOS outputs. When each of P35ODR and P34ODR bits is set to 1, the corresponding pins P35 and P34 function as NMOS open drain outputs. When cleared to 0, the corresponding pins function as NMOS push pull outputs.
6	P36ODR	0	R/W	
5	P35ODR	0	R/W	
4	P34ODR	0	R/W	
3	P33ODR	0	R/W	
2	P32ODR	0	R/W	
1	P31ODR	0	R/W	
0	P30ODR	0	R/W	

9.3.5 Pin Functions

The port 3 pins also function as SCI I/O pins, I2C bus interface 2 I/O pins, and interrupt input pins.

As shown in figure 9.1, when the pin P34, P35, SCL0, or SDA0 type open drain output is used, a bus line is not affected even if the power supply for this LSI fails. Use (a) type open drain output when using a bus line having a state in which the power is not supplied to this LSI.



(a) Open drain output type for P34, P35, SCLO, and SDA0 pins

(b) Open drain output type for P37, P36, P33 to P30, SCL1, SDA1, and port A pins

Figure 9.1 Types of Open Drain Outputs

- P37/TxD4

  The pin function is switched as shown below according to the combination of the TE bit in SCR_4 of SCI_4 and the P37DDR bit.

TE	0		1
P37DDR	0	1	—
Pin function	P37 input	P37 output*	TxD4 output

Note: * When P37ODR is set to 1, this pin functions as NMOS open drain output.

- P36/RxD4

  The pin function is switched as shown below according to the combination of the RE bit in SCR_4 of SCI_4 and the P36DDR bit.

RE	0		1
P36DDR	0	1	—
Pin function	P36 input	P36 output*	RxD4 output

Note: * When P36ODR is set to 1, this pin functions as NMOS open drain output.

- P35/SCK1/SCK4/SCL0/$\overline{\text{IRQ5}}$

  The pin function is switched as shown below according to the combination of the ICE bit in ICCR1_0 of IIC2_0, the C/$\overline{\text{A}}$ bit in SMR_1 of SCI_1 or in SMR_4 of SCI_4, the CKE0 and CKE1 bits in SCR_1 or SCR_4, and the P35DDR bit. The SCK1 and SCK4 are not set to outputs simultaneously.

ICE	0					1
CKE1	0				1	0
C/$\overline{\text{A}}$	0			1	—	0
CKE0	0		1	—	—	0
P35DDR	0	1	—	—	—	—
Pin function	P35 input	P35 output*[1]	SCK1/ SCK4 output*[1]	SCK1/ SCK4 output*[1]	SCK1/SCK 4 input	SCL0 input/ output
	$\overline{\text{IRQ5}}$ input*[2]					

Notes: 1. When P35ODR is set to 1, this pin functions as NMOS open drain output. When cleared to 0, this pin functions as NMOS push pull output.
2. When this pin is used as an external interrupt pin, do not specify other functions.

- P34/RxD1/SDA0

  The pin function is switched as shown below according to the combination of the ICE bit in ICCR1_0 of IIC2_0, the RE bit in SCR_1 of SCI_1, and the P34DDR bit.

ICE	0			1
RE	0		1	—
P34DDR	0	1	—	—
Pin function	P34 input	P34 output*	RxD1 input	SDAO input/output

Note: * When P34ODR is set to 1, this pin functions as NMOS open drain output. When cleared to 0, this pin functions as NMOS push pull output.

- P33/TxD1/SCL1

  The pin function is switched as shown below according to the combination of the ICE bit in ICCR1_1 of IIC2_1, the TE bit in SCR_1 of SCI_1, and the P33DDR bit.

ICE	0			1
TE	0		1	—
P33DDR	0	1	—	—
Pin function	P33 input	P33 output*	TxD1 output*	SCL1 input/output

Note: * When P33ODR is set to 1, this pin functions as NMOS open drain output.

- P32/SCK0/SDA1/$\overline{\text{IRQ4}}$

  The pin function is switched as shown below according to the combination of the ICE bit in ICCR1_1 of IIC2_1, the C/$\overline{\text{A}}$ bit in SMR_0 of SCI_0, the CKE0 and CKE1 bits in SCR_0, and the P32DDR bit.

ICE	0					1
CKE1	0				1	0
C/$\overline{\text{A}}$	0			1	—	0
CKE0	0		1	—	—	0
P32DDR	0	1	—	—	—	—
Pin function	P32 input	P32 output*[1]	SCK0 output*[1]	SCK0 output*[1]	SCK0 input	SDA1 input/ output
	$\overline{\text{IRQ4}}$ Input*[2]					

Notes: 1. When P32ODR is set to 1, this pin functions as NMOS open drain output.
2. When this pin is used as an external interrupt pin, do not specify other functions.

- P31/RxD0

  The pin function is switched as shown below according to the combination of the RE bit in SCR_0 of SCI_0 and the P31DDR bit.

RE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	RxD0 input

Note: * When P31ODR is set to 1, this pin functions as NMOS open drain output.

- P30/TxD0

  The pin function is switched as shown below according to the combination of the TE bit in SCR_0 of SCI_0 and the P30DDR bit.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	TxD0 output*

Note: * When P30ODR is set to 1, this pin functions as NMOS open drain output.

9.4 Port 4

Port 4 is an 8-bit input-only port and has the following register.

- Port 4 register (PORT4)

9.4.1 Port 4 Register (PORT4)

PORT4 shows port 4 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	—*	R	The pin states are always read when a port 4 read is performed.
6	P46	—*	R	
5	P45	—*	R	
4	P44	—*	R	
3	P43	—*	R	
2	P42	—*	R	
1	P41	—*	R	
0	P40	—*	R	

Note: * Determined by the states of pins P47 to P40.

9.4.2 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN0 to AN7).

9.5 Port 5

Port 5 is a 3-bit I/O port and has the following registers.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)
- Port 5 register (PORT5)

9.5.1 Port 5 Data Direction Register (P5DDR)

P5DDR specifies input or output of the port 5 pins using the individual bits. P5DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved These bits are always read as undefined value and cannot be modified.
2	P52DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 5 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
1	P51DDR	0	W	
0	P50DDR	0	W	

9.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved These bits are always read as undefined value and cannot be modified.
2	P52DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
1	P51DR	0	R/W	
0	P50DR	0	R/W	

9.5.3 Port 5 Register (PORT5)

PORT5 shows port 5 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved These bits are always read as undefined value and cannot be modified.
2	P52	—*	R	If a port 5 read is performed while P5DDR bits are set to 1, the P5DR values are read. If a port 5 read is performed while P5DDR bits are cleared to 0, the pin states are read.
1	P51	—*	R	
0	P50	—*	R	

Note: * Determined by the states of pins P52 to P50.

9.5.4 Pin Functions

Port 5 pins also function as SCI I/O pins. Port 5 pin functions are shown below.

- P52/SCK2

  The pin function is switched as shown below according to the combination of the C/$\overline{A}$ bit in SMR_2 of SCI_2, the CKE0 and CKE1 bits in SCR_2, and the P52DDR bit.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P52DDR	0	1	—	—	—
Pin function	P52 input	P52 output	SCK2 output	SCK2 output	SCK2 input

- P51/RxD2

  The pin function is switched as shown below according to the combination of the RE bit in SCR_2 of SCI_2 and the P51DDR bit.

RE	0		1
P51DDR	0	1	—
Pin function	P51 input	P51 output	RxD2 output

- P50/ TxD2

  The pin function is switched as shown below according to the combination of the TE bit in SCR_2 of SCI_2 and the P50DDR bit.

TE	0		1
P50DDR	0	1	—
Pin function	P50 input	P50 output	TxD2 input

9.6 Port 7

Port 7 is an 8-bit I/O port and has the following registers.

- Port 7 data direction register (P7DDR)
- Port 7 data register (P7DR)
- Port 7 register (PORT7)

9.6.1 Port 7 Data Direction Register (P7DDR)

P7DDR specifies input or output of the port 7 pins using the individual bits. P7DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P77DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 7 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	P76DDR	0	W	
5	P75DDR	0	W	
4	P74DDR	0	W	
3	P73DDR	0	W	
2	P72DDR	0	W	
1	P71DDR	0	W	
0	P70DDR	0	W	

9.6.2 Port 7 Data Register (P7DR)

P7DR stores output data for port 7 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P77DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	P76DR	0	R/W	
5	P75DR	0	R/W	
4	P74DR	0	R/W	
3	P73DR	0	R/W	
2	P72DR	0	R/W	
1	P71DR	0	R/W	
0	P70DR	0	R/W	

9.6.3 Port 7 Register (PORT7)

PORT7 shows port 7 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P77	—*	R	If a port 7 read is performed while P7DDR bits are set to 1, the P7DR values are read. If a port 7 read is performed while P7DDR bits are cleared to 0, the pin states are read.
6	P76	—*	R	
5	P75	—*	R	
4	P74	—*	R	
3	P73	—*	R	
2	P72	—*	R	
1	P71	—*	R	
0	P70	—*	R	

Note: * Determined by the states of pins P77 to P70.

9.6.4 Pin Functions

Port 7 pins also function as TMR I/O pins, bus control output pins, SCI I/O pins, and manual reset input pins. Port 7 pin functions are shown below.

- P77/TxD3

  The pin function is switched as shown below according to the combination of the TE bit in SCR_3 of SCI_3 and the P77DDR bit.

TE	0		1
P77DDR	0	1	—
Pin function	P77 input	P77 output	TxD3 output

- P76/RxD3

  The pin function is switched as shown below according to the combination of the RE bit in SCR_3 of SCI_3 and the P76DDR bit.

RE	0		1
P76DDR	0	1	—
Pin function	P76 input	P76 output	RxD3 input

- P75/TMO3/SCK3

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR_3 of TMR_3, the CKE1 and CKE0 bits in SCR_3 of SCI_3, the C/$\overline{A}$ bit in SMR_3, and the P75DDR bit.

OS3 to OS0	All bits are 0					Any bit is 1
CKE1	0				1	—
C/$\overline{A}$	0			1	—	—
CKE0	0		1	—	—	—
P75DDR	0	1	—	—	—	—
Pin function	P75 input	P75 output	SCK3 output	SCK3 output	SCK3 input	TMO3 output*

- P74/TMO2/$\overline{\text{MRES}}$

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR_2 of TMR_2, the MRESE bit in SYSCR, and the P74DDR bit.

MRESE	0			1
OS3 to OS0	All bits are 0		Any bit is 1	—
P74DDR	0	1	—	0
Pin function	P74 input	P74 output	TMO2 output	$\overline{\text{MRES}}$ input

- P73/TMO1/$\overline{\text{CS7}}$

  The pin function is switched as shown below according to the combination of the operating mode, the OS3 to OS0 bits in TCSR_1 of TMR_1, and the P73DDR bit.

Operating mode	Mode 6			Mode 7		
OS3 to OS0	All bits are 0		Any bit is 1	All bits are 0		Any bit is 1
P73DDR	0	1	—	0	1	—
Pin function	P73 input	$\overline{\text{CS7}}$ output	TMO1 output	P73 input	P73 output	TMO1 output

- P72/TMO0/$\overline{\text{CS6}}$

  The pin function is switched as shown below according to the combination of the operating mode, the OS3 to OS0 bits in TCSR_0 of TMR_0, and the P72DDR bit.

Operating mode	Mode 6			Mode 7		
OS3 to OS0	All bits are 0		Any bit is 1	All bits are 0		Any bit is 1
P72DDR	0	1	—	0	1	—
Pin function	P72 input	$\overline{\text{CS6}}$ output	TMO0 output	P72 input	P72 output	TMO0 output

- P71/TMRI23/TMCI23/$\overline{\text{CS5}}$

  The pin function is switched as shown below according to the combination of the operating mode and the P71DDR bit.

Operating mode	Mode 6		Mode 7	
P71DDR	0	1	0	1
Pin function	P71 input	$\overline{\text{CS5}}$ output	P71 input	P71 output
	TMRI23/TMCI23 input			

- P70/TMRI01/TMCI01/ $\overline{CS4}$

  The pin function is switched as shown below according to the combination of the operating mode and the P70DDR bit.

Operating mode	Mode 6		Mode 7	
P70DDR	0	1	0	1
Pin function	P70 input	$\overline{CS4}$ output	P70 input	P70 output
	TMRI01/TMCI01 input			

9.7 Port 9

Port 9 is an 8-bit input-only port and has the following register.

- Port 9 register (PORT9)

9.7.1 Port 9 Register (PORT9)

PORT9 shows port 9 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P97	—*	R	The pin states are always read when a port 9 read is performed.
6	P96	—*	R	
5	P95	—*	R	
4	P94	—*	R	
3	P93	—*	R	
2	P92	—*	R	
1	P91	—*	R	
0	P90	—*	R	

Note: * Determined by the states of pins P97 and P90.

9.7.2 Pin Functions

Port 9 pins also function as A/D converter analog input pins (AN15 and AN8) and D/A converter analog output pins (DA0 and DA1).

- P97/AN15/DA1

  The pin function is switched as shown below according to the combination of the DAE bit and the DAOE1 bit in DACR of D/A converter.

DAOE1	0		1
DAE	0	1	—
Pin function	P97 input	DA1 output	DA1 output
	AN15 input		

- P96/AN14/DA0

  The pin function is switched as shown below according to the combination of the DAE bit and the DAOE0 bit in DACR of D/A converter.

DAOE0	0		1
DAE	0	1	—
Pin function	P96 input	DA0 output	DA0 output
	AN14 input		

- P95/AN13, P94/AN12, P93/AN11, P92/AN10, P91/AN9, P90/AN8

Pin function	P95, P94, P93, P92, P91, P90 input pin
	AN13, AN12, AN11, AN10, AN9, AN8 input

9.8 Port A

Port A is an 8-bit I/O port and has the following registers.

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port A pull-up MOS control register (PAPCR)
- Port A open drain control register (PAODR)

9.8.1 Port A Data Direction Register (PADDR)

PADDR specifies input or output the port A pins using the individual bits. PADDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port A pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PA6DDR	0	W	
5	PA5DDR	0	W	
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

9.8.2 Port A Data Register (PADR)

PADR stores output data for port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

9.8.3 Port A Register (PORTA)

PORTA shows port A pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7	—*	R	If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.
6	PA6	—*	R	
5	PA5	—*	R	
4	PA4	—*	R	
3	PA3	—*	R	
2	PA2	—*	R	
1	PA1	—*	R	
0	PA0	—*	R	

Note: * Determined by the states of pins PA7 to PA0.

9.8.4 Port A Pull-Up MOS Control Register (PAPCR)

PAPCR controls on/off state of the input pull-up MOS for port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PA6PCR	0	R/W	
5	PA5PCR	0	R/W	
4	PA4PCR	0	R/W	
3	PA3PCR	0	R/W	
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

9.8.5 Port A Open Drain Control Register (PAODR)

PAODR selects the output type for port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	When these bits are set to 1, the corresponding pins function as NMOS open drain outputs. When cleared to 0, the corresponding pins function as CMOS outputs.
6	PA6ODR	0	R/W	
5	PA5ODR	0	R/W	
4	PA4ODR	0	R/W	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

9.8.6 Pin Functions

Port A pins also function as address output pins. Port A pin functions are shown below.

- PA7/A23

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA7DDR bit.

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	B'1111	Other than B'1111		—	
PA7DDR	—	0	1	0	1
Pin function	A23 output	PA7 input	PA7 output*	PA7 input	PA7 output*

Note: * When PA7ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

- PA6/A22

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA6DDR bit.

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	B'1111	Other than B'1111		—	
PA6DDR	—	0	1	0	1
Pin function	A22 output	PA6 input	PA6 output*	PA6 input	PA6 output*

Note: * When PA6ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

- PA5/A21

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA5DDR bit.

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	B'111x	Other than B'111x		—	
PA5DDR	—	0	1	0	1
Pin function	A21 output	PA5 input	PA5 output*	PA5 input	PA5 output*

Legend

x: Don't care

Note: * When PA5ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

- PA4/A20

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA4DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'1101 to B'1111	Other than B'1101 to B'1111		—	
PA4DDR	—	0	1	0	1
Pin function	A20 output	PA4 input	PA4 output*	PA4 input	PA4 output*

Note: * When PA4ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

- PA3/A19

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA3DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'11xx	Other than B'11xx		—	
PA3DDR	—	0	1	0	1
Pin function	A19 output	PA3 input	PA3 output*	PA3 input	PA3 output*

Legend

x: Don't care

Note: * When PA3ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

- PA2/A18

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA2DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'1011 or B'11xx	Other than B'1011 or B'11xx		—	
PA2DDR	—	0	1	0	1
Pin function	A18 output	PA2 input	PA2 output*	PA2 input	PA2 output*

Legend

x: Don't care

Note:* When PA2ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

- PA1/A17

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA1DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'101x or B'11xx	Other than B'101x or B'11xx		—	
PA1DDR	—	0	1	0	1
Pin function	A17 output	PA1 input	PA1 output*	PA1 input	PA1 output*

Legend

x: Don't care

Note: * When PA1ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

- PA0/A16

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PA0DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'0xxx or B'1000	B'0xxx or B'1000		—	
PA0DDR	—	0	1	0	1
Pin function	A16 output	PA0 input	PA0 output*	PA0 input	PA0 output*

Legend

x: Don't care

Note: * When PA0ODR in PAODR is set to 1, this pin functions as NMOS open drain output.

9.8.7 Input Pull-Up MOS Function (Port A)

Port A has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be specified as on or off on an individual bit basis. Table 9.2 summarizes the input pull-up MOS states in port A.

Table 9.2 Input Pull-Up MOS States (Port A)

Pin States	Power-on Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output and port output		OFF		OFF	
Port input				ON/OFF	

Legend:
OFF: Input pull-up MOS is always off.
ON/OFF: On when PADDR = 0 and PAPCR = 1; otherwise off.

9.9 Port B

Port B is an 8-bit I/O port and has the following registers.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B pull-up MOS control register (PBPCR)

9.9.1 Port B Data Direction Register (PBDDR)

PBDDR specifies input or output the port B pins using the individual bits. PBDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port B pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

9.9.2 Port B Data Register (PBDR)

PBDR stores output data for port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

9.9.3 Port B Register (PORTB)

PORTB shows port B pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—*	R	If a port B read is performed while PBDDR bits are set to 1, the PBDR values are read. If a port B read is performed while PBDDR bits are cleared to 0, the pin states are read.
6	PB6	—*	R	
5	PB5	—*	R	
4	PB4	—*	R	
3	PB3	—*	R	
2	PB2	—*	R	
1	PB1	—*	R	
0	PB0	—*	R	

Note: * Determined by the states of pins PB7 to PB0.

9.9.4 Port B Pull-Up MOS Control Register (PBPCR)

PBPCR controls on/off state of the input pull-up MOS for port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

9.9.5 Pin Functions

Port B pins also function as address output pins. Port B pin functions are shown below.

- PB7/A15

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB7DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'1xxx	Other than B'1xxx		—	
PB7DDR	—	0	1	0	1
Pin function	A15 output	PB7 input	PB7 output	PB7 input	PB7 output

Legend

x: Don't care

- PB6/A14

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB6DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'0111 or B'1xxx	Other than B'0111 or B'1xxx		—	
PB6DDR	—	0	1	0	1
Pin function	A14 output	PB6 input	PB6 output	PB6 input	PB6 output

Legend

x: Don't care

- PB5/A13

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB5DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'011x or B'1xxx	Other than B'111x or B'1xxx		—	
PB5DDR	—	0	1	0	1
Pin function	A13 output	PB5 input	PB5 output	PB5 input	PB5 output

Legend

x: Don't care

- PB4/A12

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB4DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'0100 or B'00xx	B'0100 or B'00xx		—	
PB4DDR	—	0	1	0	1
Pin function	A12 output	PB4 input	PB4 output	PB4 input	PB4 output

Legend

x: Don't care

- PB3/A11

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB3DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'00xx	B'00xx		—	
PB3DDR	—	0	1	0	1
Pin function	A11 output	PB3 input	PB3 output	PB3 input	PB3 output

Legend

x: Don't care

- PB2/A10

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB2DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'0010 or B'000x	B'0010 or B'000x		—	
PB2DDR	—	0	1	0	1
Pin function	A10 output	PB2 input	PB2 output	PB2 input	PB2 output

Legend

x: Don't care

- PB1/A9

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB1DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'000x	B'000x		—	
PB1DDR	—	0	1	0	1
Pin function	A9 output	PB1 input	PB1 output	PB1 input	PB1 output

Legend

x: Don't care

- PB0/A8

  The pin function is switched as shown below according to the combination of the operating mode, the AE3 to AE0 bits in PFCR, and the PB0DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'0000	B'0000		—	
PB0DDR	—	0	1	0	1
Pin function	A8 output	PB0 input	PB0 output	PB0 input	PB0 output

Legend

x: Don't care

9.9.6 Input Pull-Up MOS Function (Port B)

Port B has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be specified as on or off on an individual bit basis. Table 9.3 summarizes the input pull-up MOS states in port B.

Table 9.3 Input Pull-Up MOS States (Port B)

Pin States	Power-on Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output and port output		OFF		OFF	
Port input				ON/OFF	

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PBDDR = 0 and PBPCR = 1; otherwise off.

9.10 Port C

Port C is an 8-bit I/O port and has the following registers.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)
- Port C pull-up MOS control register (PCPCR)

9.10.1 Port C Data Direction Register (PCDDR)

PCDDR specifies input or output the port C pins using the individual bits. PCDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port C pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

9.10.2 Port C Data Register (PCDR)

PCDR stores output data for port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

9.10.3 Port C Register (PORTC)

PORTC shows port C pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	—*	R	If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read. If a port C read is performed while PCDDR bits are cleared to 0, the pin states are read.
6	PC6	—*	R	
5	PC5	—*	R	
4	PC4	—*	R	
3	PC3	—*	R	
2	PC2	—*	R	
1	PC1	—*	R	
0	PC0	—*	R	

Note: * Determined by the states of pins PC7 to PC0.

9.10.4 Port C Pull-Up MOS Control Register (PCPCR)

PCPCR controls on/off state of the input pull-up MOS for port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

9.10.5 Pin Functions

Port C pins also function as address output pins. Port C pin functions are shown below.

- PC7/A7, PC6/A6, PC5/A5, PC4/A4, PC3/A3, PC2/A2, PC1/A1, PC0/A0

  The pin function is switched as shown below according to the combination of the operating mode and the PCnDDR bit.

Operating mode	Mode 6		Mode 7	
PCnDDR	0	1	0	1
Pin function	PCn input	Address output	PCn input	PCn output

Note: n = 7 to 0

9.10.6 Input Pull-Up MOS Function (Port C)

Port C has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be used in mode 6 or mode 7, and can be specified as on or off on an individual bit basis. Table 9.4 summarizes the input pull-up MOS states in port C.

Table 9.4 Input Pull-Up MOS States (Port C)

Pin States	Power-on Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output (mode 6) and port output (mode 7)		OFF		OFF	
Port input				ON/OFF	

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PCDDR = 0 and PCPCR = 1; otherwise off.

9.11 Port D

Port D is an 8-bit I/O port and has the following registers.

- Port D data direction register (PDDDR)
- Port D data register (PDDR)
- Port D register (PORTD)
- Port D pull-up MOS control register (PDPCR)

9.11.1 Port D Data Direction Register (PDDDR)

PDDDR specifies input or output the port D pins using the individual bits. PDDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port D pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

9.11.2 Port D Data Register (PDDR)

PDDR stores output data for port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

9.11.3 Port D Register (PORTD)

PORTD shows port D pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	—*	R	If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.
6	PD6	—*	R	
5	PD5	—*	R	
4	PD4	—*	R	
3	PD3	—*	R	
2	PD2	—*	R	
1	PD1	—*	R	
0	PD0	—*	R	

Note: * Determined by the states of pins PD7 to PD0.

9.11.4 Port D Pull-Up MOS Control Register (PDPCR)

PDPCR controls on/off state of the input pull-up MOS for port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

9.11.5 Pin Functions

Port D pins also function as data I/O pins. Port D pin functions are shown below.

- PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8
  The pin function is switched as shown below according to the combination of the operating mode and the PDnDDR bit.

Operating mode	Mode 6	Mode 7	
PDnDDR	—	0	1
Pin function	Data input/output	PDn input	PDn output

Note: n = 7 to 0

9.11.6 Input Pull-Up MOS Function (Port D)

Port D has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be used in mode 7, and can be specified as on or off on an individual bit basis. Table 9.5 summarizes the input pull-up MOS states in port D.

Table 9.5 Input Pull-Up MOS States (Port D)

Pin States	Power-on Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Data input/output (mode 6) and port output (mode 7)		OFF		OFF	
Port input (mode 7)				ON/OFF	

Legend:
OFF: Input pull-up MOS is always off.
ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

9.12 Port E

Port E is an 8-bit I/O port and has the following registers.

- Port E data direction register (PEDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E pull-up MOS control register (PEPCR)

9.12.1 Port E Data Direction Register (PEDDR)

PEDDR specifies input or output the port E pins using the individual bits. PEDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port E pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

9.12.2 Port E Data Register (PEDR)

PEDR stores output data for port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

9.12.3 Port E Register (PORTE)

PORTE shows port E pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	—*	R	If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read. If a port E read is performed while PEDDR bits are cleared to 0, the pin states are read.
6	PE6	—*	R	
5	PE5	—*	R	
4	PE4	—*	R	
3	PE3	—*	R	
2	PE2	—*	R	
1	PE1	—*	R	
0	PE0	—*	R	

Note: * Determined by the states of pins PE7 to PE0.

9.12.4 Port E Pull-Up MOS Control Register (PEPCR)

PEPCR controls on/off state of the input pull-up MOS for port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PCR	0	R/W	When a pin is specified as an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

9.12.5 Pin Functions

Port E pins also function as data I/O pins. Port E pin functions are shown below.

- PE7/D7, PE6/D6, PE5/D5, PE4/D4, PE3/D3, PE2/D2, PE1/D1, PE0/D0

  The pin function is switched as shown below according to the combination of the operating mode and the PEnDDR bit.

Operating mode	Mode 6			Mode 7	
Bus mode	8-bit bus mode		16-bit bus mode	—	
PEnDDR	0	1	—	0	1
Pin function	PEn input	PEn output	Data input/output	PEn input	PEn output

Note: n = 7 to 0

9.12.6 Input Pull-Up MOS Function (Port E)

Port E has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be used in mode 7 or 8-bit bus mode in modes 4 to 6, and can be specified as on or off on an individual bit basis. Table 9.6 summarizes the input pull-up MOS states in port E.

Table 9.6 Input Pull-Up MOS States (Port E)

Pin States	Power-on Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Data input/output (16-bit bus in mode 6) and port output (8-bit bus in mode 6, mode 7)		OFF		OFF	
Port input (8-bit bus in mode 6, mode 7)				ON/OFF	

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PEDDR = 0 and PEPCR = 1; otherwise off.

9.13 Port F

Port F is an 8-bit I/O port and has the following registers.

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)

9.13.1 Port F Data Direction Register (PFDDR)

PFDDR specifies input or output the port F pins using the individual bits. PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	0/1*	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port F pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

Note: * PF7DDR is initialized to 1 in mode 6 and 0 in mode 7.

9.13.2 Port F Data Register (PFDR)

PFDR stores output data for port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

9.13.3 Port F Register (PORTF)

PORTF shows port F pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	—*	R	If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F read is performed while PFDDR bits are cleared to 0, the pin states are read.
6	PF6	—*	R	
5	PF5	—*	R	
4	PF4	—*	R	
3	PF3	—*	R	
2	PF2	—*	R	
1	PF1	—*	R	
0	PF0	—*	R	

Note: * Determined by the states of pins PF7 to PF0.

9.13.4 Pin Functions

Port F pins also function as bus control I/O pins, interrupt input pins, system clock output pins, A/D trigger input pins, and BUZZ output pins. Port F pin functions are shown below.

- PF7/ϕ

  The pin function is switched as shown below according to the PF7DDR bit.

PF7DDR	0	1
Pin function	PF7 input	ϕ output

- PF6/$\overline{AS}$

  The pin function is switched as shown below according to the combination of the operating mode and the PF6DDR bit.

Operating mode	Mode 6	Mode 7	
PF6DDR	—	0	1
Pin function	$\overline{AS}$ output	PF6 input	PF6 output

- PF5/$\overline{RD}$

  The pin function is switched as shown below according to the combination of the operating mode and the PF5DDR bit.

Operating mode	Mode 6	Mode 7	
PF5DDR	—	0	1
Pin function	$\overline{RD}$ output	PF5 input	PF5 output

- PF4/$\overline{HWR}$

  The pin function is switched as shown below according to the combination of the operating mode and the PF4DDR bit.

Operating mode	Mode 6	Mode 7	
PF4DDR	—	0	1
Pin function	$\overline{HWR}$ output	PF4 input	PF4 output

- PF3/$\overline{LWR}$/$\overline{ADTRG}$/$\overline{IRQ3}$

  The pin function is switched as shown below according to the combination of the operation mode, the bus mode, the TRGS1 and TRGS0 bits in ADCR of the A/D converter, and the PF3DDR bit.

Operating mode	Mode 6			Mode 7	
Bus mode	16-bit bus mode	8-bit bus mode		—	
PF3DDR	—	0	1	0	1
Pin function	$\overline{LWR}$ output	PF3 input	PF3 output	PF3 input	PF3 output
		$\overline{ADTRG}$ input*1			
		$\overline{IRQ3}$ input*2			

Notes: 1. When TRGS0 = TRGS1 = 1, port F is used as the $\overline{ADTRG}$ input pin.
2. When this port is used as an external interrupt pin, do not specify other functions.

- PF2/$\overline{WAIT}$

  The pin function is switched as shown below according to the combination of the operating mode, the WAITE bit, and the PF2DDR bit.

Operating mode	Mode 6			Mode 7	
WAITE	0		1		—
PF2DDR	0	1	—	0	1
Pin function	PF2 input	PF2 output	$\overline{WAITE}$ input	PF2 input	PF2 output

- PF1/$\overline{\text{BACK}}$/BUZZ

  The pin function is switched as shown below according to the combination of the operating mode, the BRLE bit, the BUZZ bit in PFCR, and the PF1DDR bit.

Operating mode	Mode 6				Mode 7		
BRLE	0			1	—		
BUZZE	0		1	—	0		1
PF1DDR	0	1	—	—	0	1	—
Pin function	PF1 input	PF1 output	BUZZ output	$\overline{\text{BACK}}$ output	PF1 input	PF1 output	BUZZ output

- PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$

  The pin function is switched as shown below according to the combination of the operating mode, the BRLE bit, and the PF0DDR bit.

Operating mode	Mode 6			Mode 7	
BRLE	0		1	—	
PF0DDR	0	1	—	0	1
Pin function	PF0 input	PF0 output	$\overline{\text{BREQ}}$ input	PF0 input	PF0 output
	$\overline{\text{IRQ2}}$ input*				

Note: * When this port is used as an external interrupt pin, do not specify other functions.

9.14 Port G

Port G is a 5-bit I/O port and has the following registers.

- Port G data direction register (PGDDR)
- Port G data register (PGDR)
- Port G register (PORTG)

9.14.1 Port G Data Direction Register (PGDDR)

PGDDR specifies input or output the port G pins using the individual bits. PGDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are always read as undefined value and cannot be modified.
4	PG4DDR	0	W	When a pin is specified as a general purpose I/O port, setting these bits to 1 makes the corresponding port G pin an output pin. Clearing this bit to 0 makes the pin an input pin.
3	PG3DDR*	0	W	
2	PG2DDR*	0	W	
1	PG1DDR	0	W	
0	PG0DDR	0	W	

Note: * Reserved in the H8S/2556 Series. This bit is set to 0.

9.14.2 Port G Data Register (PGDR)

PGDR stores output data for port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are always read as undefined value and cannot be modified.
4	PG4DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
3	PG3DR*	0	R/W	
2	PG2DR*	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

Note: * Reserved in the H8S/2556 Series. This bit is set to 0.

9.14.3 Port G Register (PORTG)

PORTG shows port G pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved This bit is always read as undefined value and cannot be modified.
4	PG4	—*[1]	R	If these bits are read while the corresponding PGDDR bits are set to 1, the PGDR value is read. If these bits are read while PGDDR bits are cleared to 0, the pin states are read.
3	PG3*[2]	—*[1]	R	
2	PG2*[2]	—*[1]	R	
1	PG1	—*[1]	R	
0	PG0	—*[1]	R	

Notes: 1. Determined by the states of pins PG4 to PG0
2. Reserved in the H8S/2556 Series. An undefined value will be read.

9.14.4 Pin Functions

Port G pins also function as bus control I/O pins, interrupt input pins, and IEB I/O pins. Port G pin functions are shown below.

- PG4/$\overline{CS0}$

  The pin function is switched as shown below according to the combination of the operating mode and the PG4DDR bit.

Operating mode	Mode 6		Mode 7	
PG4DDR	0	1	0	1
Pin function	PG4 input	$\overline{CS0}$ output	PG4 input	PG4 output

- PG3/$\overline{Rx}$/$\overline{CS1}$

  In the H8S/2552 and H8S/2506 Series, the pin function is switched as shown below according to the combination of the IEE bit in IECTR of IEB*, the operating mode, and the PG3DDR bit. This pin is not available in the H8S/2556 Series.

IEE	0				1
Operating mode	Mode 6		Mode 7		—
PG3DDR	0	1	0	1	—
Pin function	PG3 input	$\overline{CS1}$ output	PG3 input	PG3 output	$\overline{Rx}$ input

Note: * IEB is supported only by the H8S/2552 Series.

- PG2/$\overline{\text{Tx}}$/$\overline{\text{CS2}}$

  In the H8S/2552 and H8S/2506 Series, the pin function is switched as shown below according to the combination of the IEE bit in IECTR of IEB*, the operating mode, and the PG2DDR bit. This pin is not available in the H8S/2556 Series.

IEE	0				1
Operating mode	Mode 6		Mode 7		—
PG2DDR	0	1	0	1	—
Pin function	PG2 input	$\overline{\text{CS2}}$ output	PG2 input	PG2 output	$\overline{\text{Tx}}$ input

Note:* IEB is supported only by the H8S/2552 Series.

- PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$

  The pin function is switched as shown below according to the combination of the operating mode and the PG1DDR bit.

Operating mode	Mode 6		Mode 7	
PG1DDR	0	1	0	1
Pin function	PG1 input	$\overline{\text{CS3}}$ output	PG1 input	PG1 output
	$\overline{\text{IRQ7}}$ input*			

Note: * When this port is used as an external interrupt pin, do not specify other functions.

- PG0/$\overline{\text{IRQ6}}$

  The pin function is switched as shown below according to the PG0DDR bit.

PG0DDR	0	1
Pin function	PG0 input	PG0 output
	$\overline{\text{IRQ6}}$ interrupt input*	

Note: * When this port is used as an external interrupt pin, do not specify other functions.

9.15 Port H

Port H is an 8-bit I/O port and has the following registers.

- Port H data direction register (PHDDR)
- Port H data register (PHDR)
- Port H register (PORTH)

9.15.1 Port H Data Direction Register (PHDDR)

PHDDR specifies input or output the port H pins using the individual bits. PHDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PH7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port H pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PH6DDR	0	W	
5	PH5DDR	0	W	
4	PH4DDR	0	W	
3	PH3DDR	0	W	
2	PH2DDR	0	W	
1	PH1DDR	0	W	
0	PH0DDR	0	W	

9.15.2 Port H Data Register (PHDR)

PHDR stores output data for port H pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PH7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PH6DR	0	R/W	
5	PH5DR	0	R/W	
4	PH4DR	0	R/W	
3	PH3DR	0	R/W	
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

9.15.3 Port H Register (PORTH)

PORTH shows port H pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PH7	—*	R	If a port H read is performed while PHDDR bits are set to 1, the PHDR values are read. If a port H read is performed while PHDDR bits are cleared to 0, the pin states are read.
6	PH6	—*	R	
5	PH5	—*	R	
4	PH4	—*	R	
3	PH3	—*	R	
2	PH2	—*	R	
1	PH1	—*	R	
0	PH0	—*	R	

Note: * Determined by the states of pins PH7 to PH0.

9.15.4 Pin Functions

Port H pins also function as general purpose I/O pins. Port H pin functions are shown below.

- PH7, PH6, PH5, PH4, PH3, PH2, PH1, PH0
  The pin function is switched as shown below according to the PHnDDR bit.

PHnDDR	0	1
Pin function	PHn input	PHn output

Note: n = 7 to 0

9.16 Port J

Port J is an 8-bit I/O port and has the following registers.

- Port J data direction register (PJDDR)
- Port J data register (PJDR)
- Port J register (PORTJ)

9.16.1 Port J Data Direction Register (PJDDR)

PJDDR specifies input or output the port J pins using the individual bits. PJDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port J pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PJ6DDR	0	W	
5	PJ5DDR	0	W	
4	PJ4DDR	0	W	
3	PJ3DDR	0	W	
2	PJ2DDR	0	W	
1	PJ1DDR	0	W	
0	PJ0DDR	0	W	

9.16.2 Port J Data Register (PJDR)

PJDR stores output data for port J pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose I/O port.
6	PJ6DR	0	R/W	
5	PJ5DR	0	R/W	
4	PJ4DR	0	R/W	
3	PJ3DR	0	R/W	
2	PJ2DR	0	R/W	
1	PJ1DR	0	R/W	
0	PJ0DR	0	R/W	

9.16.3 Port J Register (PORTJ)

PORTJ shows port J pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7	—*	R	If a port J read is performed while PJDDR bits are set to 1, the PJDR values are read. If a port J read is performed while PJDDR bits are cleared to 0, the pin states are read.
6	PJ6	—*	R	
5	PJ5	—*	R	
4	PJ4	—*	R	
3	PJ3	—*	R	
2	PJ2	—*	R	
1	PJ1	—*	R	
0	PJ0	—*	R	

Note: * Determined by the states of pins PJ7 to PJ0.

9.16.4 Pin Functions

Port J pins also function as general purpose I/O pins. Port J pin functions are shown below.

- PJ7, PJ6, PJ5, PJ4, PJ3, PJ2, PJ1, PJ0

  The pin function is switched as shown below according to the PJnDDR bit.

PJnDDR	0	1
Pin function	PJn input	PJn output

Note: n = 7 to 0

Section 10 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) comprised of six 16-bit timer channels.

The function list of the 16-bit timer unit and its block diagram are shown in table 10.1 and figure 10.1, respectively.

10.1 Features

- Maximum 16-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - — Waveform output at compare match
  - — Input capture function
  - — Counter clear operation
  - — Synchronous operation:
    Multiple timer counters (TCNT) can be written to simultaneously
    Simultaneous clearing by compare match and input capture is possible
    Register simultaneous input/output is possible by synchronous counter operation
  - — A maximum 15-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

Table 10.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		φ/1 φ/4 φ/16 φ/64 TCLKA TCLKB TCLKC TCLKD	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKB	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKB TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 φ/1024 φ/4096 TCLKA	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKC TCLKD
General registers (TGR)		TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRA_5 TGRB_5
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare Match output	0 output	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
A/D converter trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture	TGRA_5 compare match or input capture
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Legend

○: Possible

—: Not possible

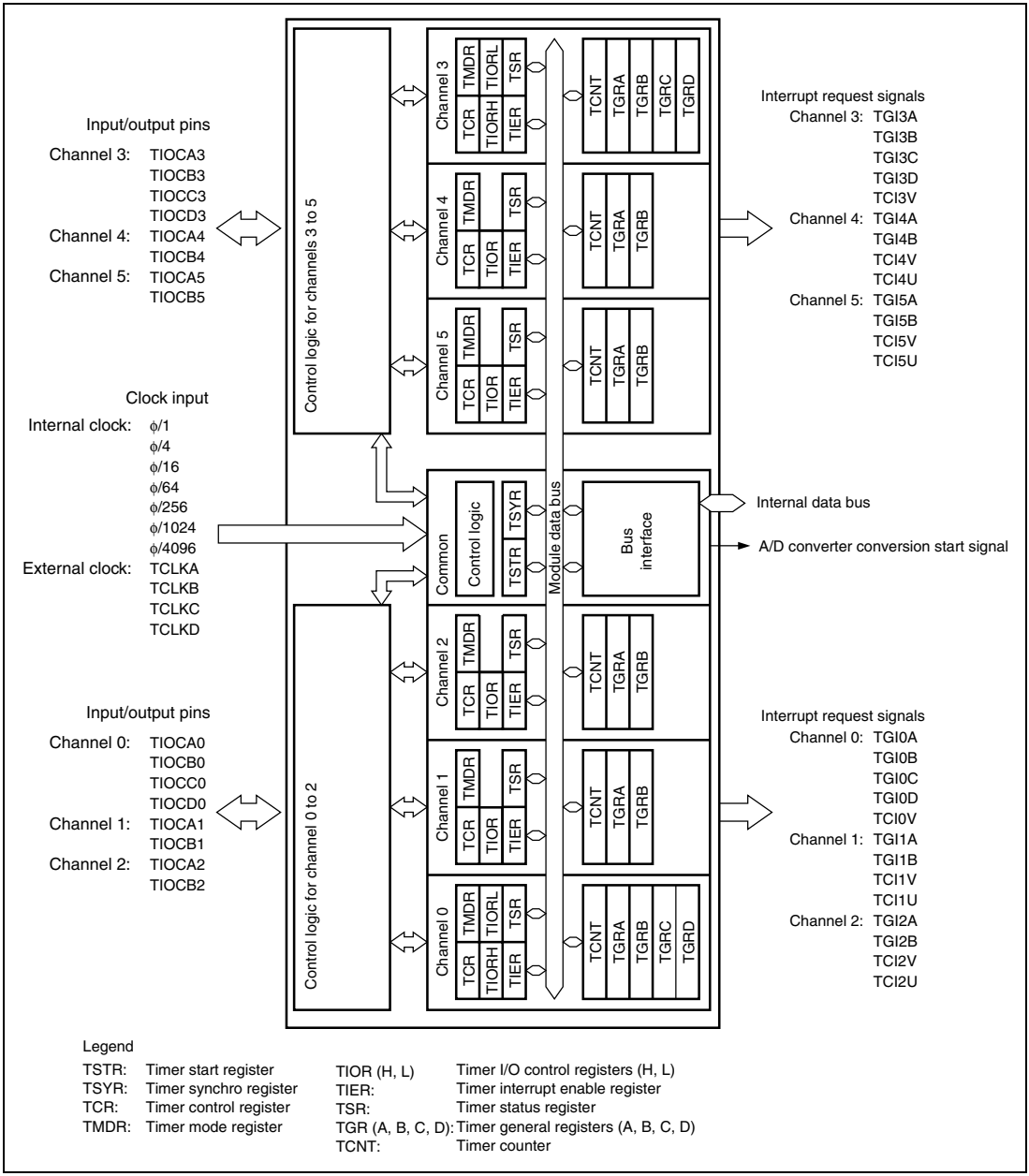


Figure 10.1 Block Diagram of TPU

10.2 Input/Output Pins

Table 10.2 TPU Pins

Channel	Symbol	I/O	Function
Common	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM output pin
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM output pin

10.3 Register Descriptions

The TPU has the following registers.

Channel 0

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

Channel 1

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)

Channel 2

- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)

Channel 3

- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)

- Timer I/O control register L_3 (TIORL_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)

Channel 4

- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register _4 (TIOR_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)

Channel 5

- Timer control register_5 (TCR_5)
- Timer mode register_5 (TMDR_5)
- Timer I/O control register_5 (TIOR_5)
- Timer interrupt enable register_5 (TIER_5)
- Timer status register_5 (TSR_5)
- Timer counter_5 (TCNT_5)
- Timer general register A_5 (TGRA_5)
- Timer general register B_5 (TGRB_5)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

10.3.1 Timer Control Register (TCR)

The TCR registers control the TCNT operation for each channel. The TPU has a total of six TCR registers, one for each channel (channels 0 to 5). TCR register settings should be conducted only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 10.3 and 10.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. $\phi/4$ both edges = $\phi/2$ rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is $\phi/4$ or slower. This setting is ignored if the input clock is $\phi/1$, or when overflow/underflow of another channel is selected. 00: Count at rising edge 01: Count at falling edge 1X: Count at both edges Legend X: Don't care
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.5 to 10.10 for details.
0	TPSC0	0	R/W	

Table 10.3 CCLR0 to CCLR2 (channels 0 and 3)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*[2]
		1	0	TCNT cleared by TGRD compare match/input capture*[2]
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]

Notes: 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Table 10.4 CCLR0 to CCLR2 (channels 1, 2, 4, and 5)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*[1]

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Table 10.5 TPSC0 to TPSC2 (channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 10.6 TPSC0 to TPSC2 (channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on ϕ/256
			1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 10.7 TPSC0 to TPSC2 (channels 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on ϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 10.8 TPSC0 to TPSC2 (channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	Internal clock: counts on ϕ/1024
		1	0	Internal clock: counts on ϕ/256
			1	Internal clock: counts on ϕ/4096

Table 10.9 TPSC0 to TPSC2 (channel 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on ϕ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 10.10 TPSC0 to TPSC2 (channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on ϕ/256
			1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

10.3.2 Timer Mode Register (TMDR)

The TMDR registers are used to set the operating mode of each channel. The TPU has six TMDR registers, one for each channel. TMDR register settings should be changed only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD are used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1:TGRA and TGRC are used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0 These bits are used to set the timer operating mode. MD3 is a reserved bit. In a write, it should always be written with 0. See table 10.11 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 10.11 MD0 to MD3

Bit 3 MD3*[1]	Bit 2 MD2*[2]	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	X	X	X	—

Legend

X: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. Care is required as TIOR is affected by the TMDR setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

Bit	Bit Name	Initial value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB. Refer to tables 10.12, 10.14, 10.15, 10.16, 10.18, and 10.19 for details.
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA. Refer to tables 10.20, 10,22, 10.23, 10.24, 10.26, and 10.27 for details.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- TIORL_0, TIORL_3

Bit	Bit Name	Initial value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD. Refer to tables 10.13, and 10.17 for details.
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C3 to C0
2	IOC2	0	R/W	Specify the function of TGRC. Refer to tables 10.21, and 10.25 for details.
1	IOC1	0	R/W	
0	IOC0	0	R/W	

Table 10.12 TIORH_0

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOCB0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	X		Capture input source is TIOCB0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*

Legend

X: Don’t care

Note: * When the TPSC0 to TPSC2 bits in TCR_1 are set to B'000 and ϕ/1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.

Table 10.13 TIORL_0

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOCD0 Pin Function**
0	0	0	0	Output compare register*2	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	X		Capture input source is TIOCD0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down*1

Legend

X: Don't care

Notes: 1. When the TPSC0 to TPSC2 bits in TCR_1 are set to B'000 and ϕ/1 is used as the TCNT_1 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.14 TIOR_1

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	X		Capture input source is TIOCB1 pin Input capture at both edges
	1	X	X		TGRC_0 compare match/ input capture Input capture at generation of TGRC_0 compare match/input capture

Legend

X: Don’t care

Table 10.15 TIOR_2

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	X		Capture input source is TIOCB2 pin Input capture at both edges

Legend

X: Don't care

Table 10.16 TIORH_3

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_3 Function**	**TIOCB3 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge
			1		Capture input source is TIOCB3 pin Input capture at falling edge
		1	X		Capture input source is TIOCB3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*

Legend

X: Don't care

Note: * When the TPSC0 to TPSC2 bits in TCR_4 are set to B'000 and ϕ/1 is used as the TCNT_4 count clock, this setting is invalid and input capture is not generated.

Table 10.17 TIORL_3

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_3 Function**	**TIOCD3 Pin Function**
0	0	0	0	Output compare register*2	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 pin Input capture at rising edge
			1		Capture input source is TIOCD3 pin Input capture at falling edge
		1	X		Capture input source is TIOCD3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*1

Legend

X: Don't care

Notes: 1. When the TPSC0 to TPSC2 bits in TCR_4 are set to B'000 and ϕ/1 is used as the TCNT_4 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.18 TIOR_4

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOCB4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 pin Input capture at rising edge
			1		Capture input source is TIOCB4 pin Input capture at falling edge
		1	X		Capture input source is TIOCB4 pin Input capture at both edges
	1	X	X		Capture input source is TGRC_3 compare match/input capture Input capture at generation of TGRC_3 compare match/input capture

Legend

X: Don’t care

Table 10.19 TIOR_5

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_5 Function**	**TIOCB5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCB5 pin Input capture at rising edge
			1		Capture input source is TIOCB5 pin Input capture at falling edge
		1	X		Capture input source is TIOCB5 pin Input capture at both edges

Legend

X: Don't care

Table 10.20 TIORH_0

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOCA0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	X		Capture input source is TIOCA0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

Legend

X: Don’t care

Table 10.21 TIORL_0

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOCC0 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC0 pin Input capture at rising edge
			1		Capture input source is TIOCC0 pin Input capture at falling edge
		1	X		Capture input source is TIOCC0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

Legend

X: Don't care

Note: * When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.22 TIOR_1

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOCA1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	X		Capture input source is TIOCA1 pin Input capture at both edges
	1	X	X		Capture input source is TGRA_0 compare match/input capture Input capture at generation of channel 0/TGRA_0 compare match/input capture

Legend

X: Don’t care

Table 10.23 TIOR_2

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	X		Capture input source is TIOCA2 pin Input capture at both edges

Legend

X: Don't care

Table 10.24 TIORH_3

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOCA3 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
			1		Capture input source is TIOCA3 pin Input capture at falling edge
		1	X		Capture input source is TIOCA3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

Legend

X: Don’t care

Table 10.25 TIORL_3

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOCC3 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC3 pin Input capture at rising edge
			1		Capture input source is TIOCC3 pin Input capture at falling edge
		1	X		Capture input source is TIOCC3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

Legend

X: Don't care

Note: * When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.26 TIOR_4

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOCA4 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
			1		Capture input source is TIOCA4 pin Input capture at falling edge
		1	X		Capture input source is TIOCA4 pin Input capture at both edges
	1	X	X		Capture input source is TGRA_3 compare match/input capture Input capture at generation of TGRA_3 compare match/input capture

Legend

X: Don’t care

Table 10.27 TIOR_5

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_5 Function**	**TIOCA5 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCA5 pin Input capture at rising edge
			1		Capture input source is TIOCA5 pin Input capture at falling edge
		1	X		Capture input source is TIOCA5 pin Input capture at both edges

Legend

X: Don't care

10.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1, 2, 4, and 5. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

Bit	Bit Name	Initial value	R/W	Description
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

10.3.5 Timer Status Register (TSR)

The TSR registers indicate the status of each channel. The TPU has six TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3. Only 0 can be written, for flag clearing. In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD and TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register [Clearing conditions] • When DTC is activated by TGID interrupt and the DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3. Only 0 can be written, for flag clearing. In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRC and TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIC interrupt and the DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1

Bit	Bit Name	Initial value	R/W	Description
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRB and TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIB interrupt and the DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing. [Setting conditions] • When TCNT = TGRA and TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIA interrupt and the DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1

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

10.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit readable/writable counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.3.7 Timer General Register (TGR)

The TGR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

10.3.8 Timer Start Register (TSTR)

TSTR specifies whether to operate or stop TCNT for channels 0 to 5. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	—	Reserved Only 0 should be written to these bits.
5	CST5	0	R/W	Counter Start 5 to 0 These bits specify whether to operate or stop TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_0 to TCNT_5 count operation is stopped 1: TCNT_0 to TCNT_5 performs count operation
4	CST4	0	R/W	
3	CST3	0	R/W	
2	CST2	0	R/W	
1	CST1	0	R/W	
0	CST0	0	R/W	

10.3.9 Timer Synchro Register (TSYR)

TSYR selects the independent operation or synchronous operation of TCNT for channels 0 to 5. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved Only 0 should be written to these bits.
5	SYNC5	0	R/W	Timer Synchro 5 to 0 These bits are used to select the independent or synchronized operation with other channels. When synchronous operation is selected, the TCNT synchronous presetting of multiple channels, and synchronous clearing by counter clearing on another channel, are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit , the TCNT clearing source must also be set by means of the CCLR0 to CCLR2 bits in TCR. 0: TCNT_0 to TCNT_5 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_0 to TCNT_5 performs synchronous operation. TCNT synchronous presetting/synchronous clearing is possible.
4	SYNC4	0	R/W	
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

10.4 Operation

10.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Counter Operation: When one of the CST0 to CST5 bits is set to 1 in TSTR, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

1. Example of count operation setting procedure
   Figure 10.2 shows an example of the count operation setting procedure.

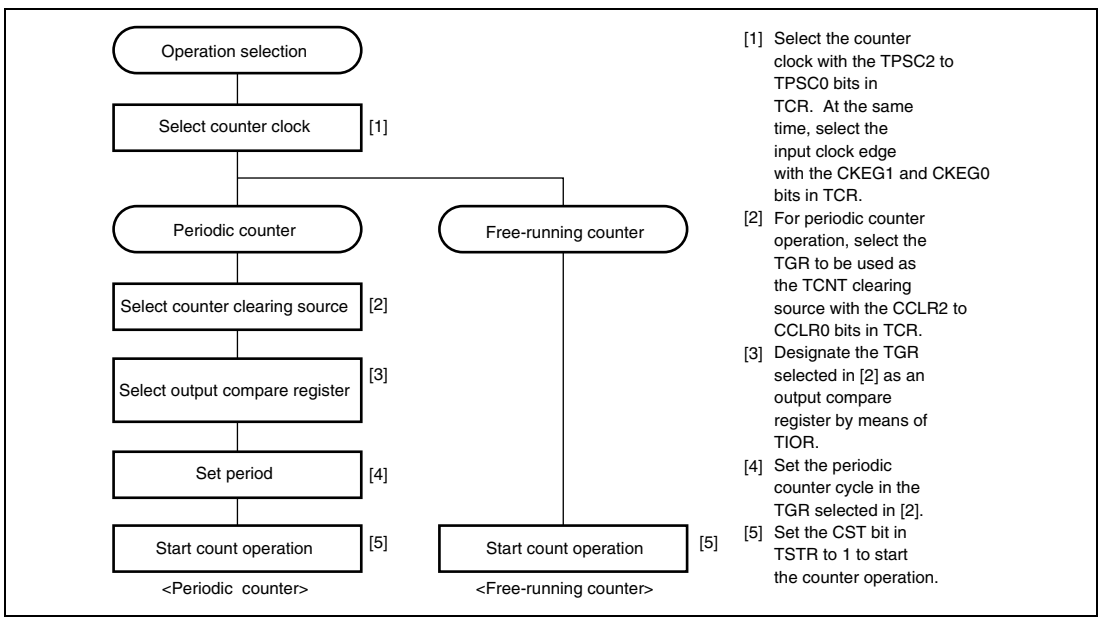


Figure 10.2 Example of Counter Operation Setting Procedure

2. Free-running count operation and periodic count operation
   Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 10.3 illustrates free-running counter operation.

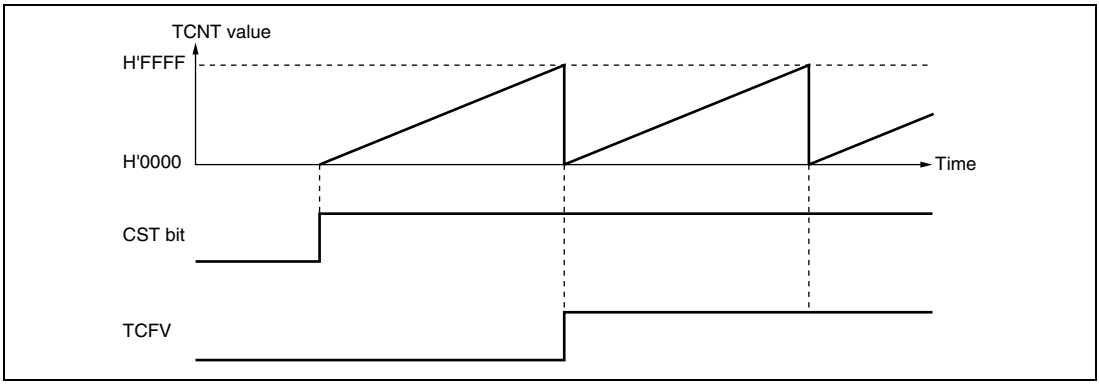


Figure 10.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of the CCLR0 to CCLR2 bits in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.4 illustrates periodic counter operation.

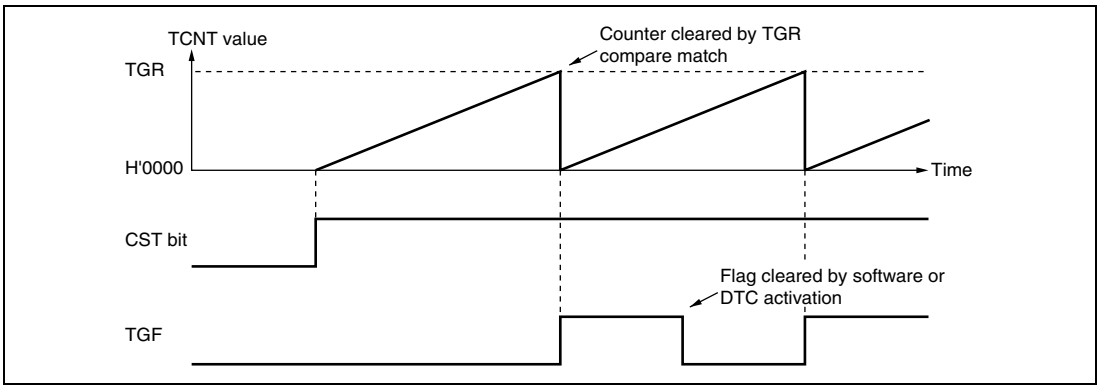


Figure 10.4 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

1. Example of setting procedure for waveform output by compare match

   Figure 10.5 shows an example of the setting procedure for waveform output by compare match.

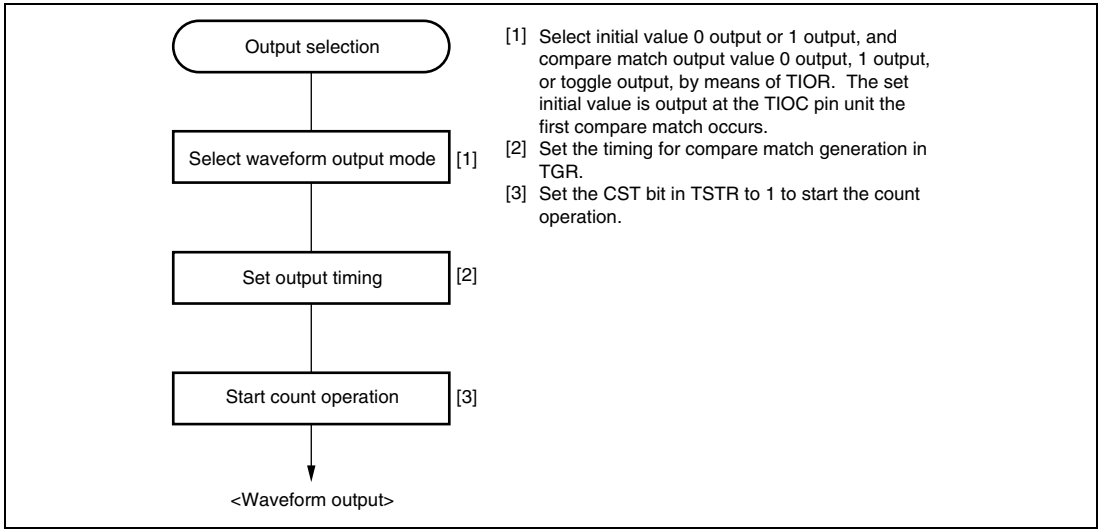


Figure 10.5 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of waveform output operation

   Figure 10.6 shows an example of 0 output/1 output.

   In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

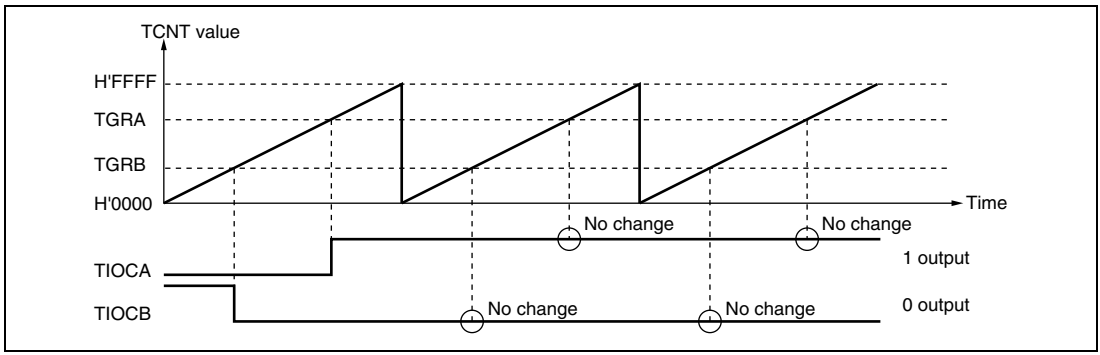


Figure 10.6 Example of 0 Output/1 Output Operation

Figure 10.7 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

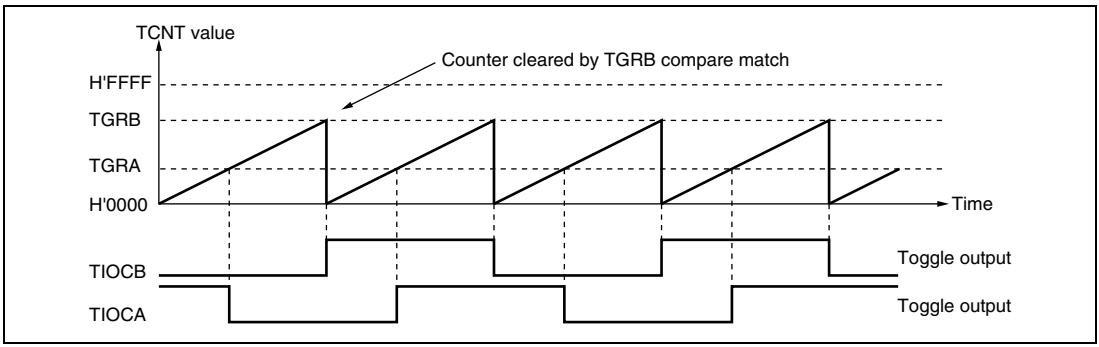


Figure 10.7 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 3, ϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if ϕ/1 is selected.

1. Example of input capture operation setting procedure

   Figure 10.8 shows an example of the input capture operation setting procedure.

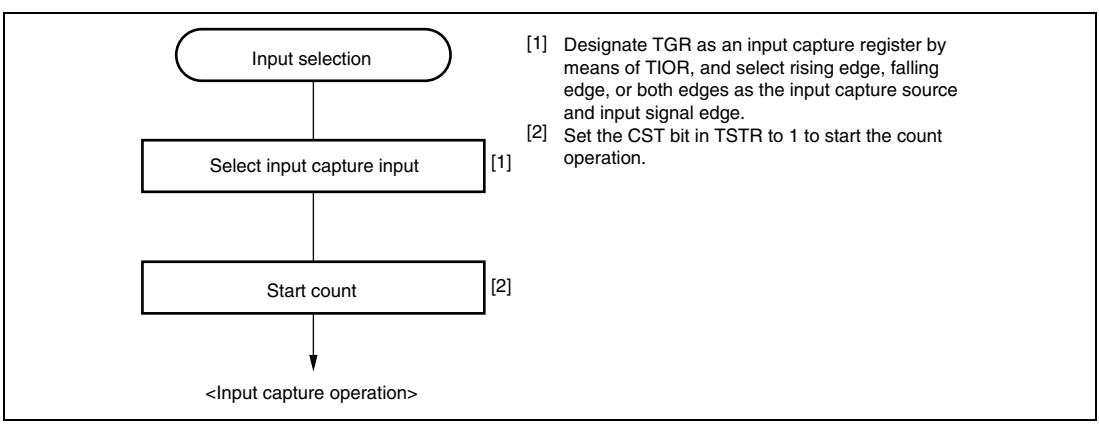


Figure 10.8 Example of Input Capture Operation Setting Procedure

2. Example of input capture operation

   Figure 10.9 shows an example of input capture operation.

   In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, the falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

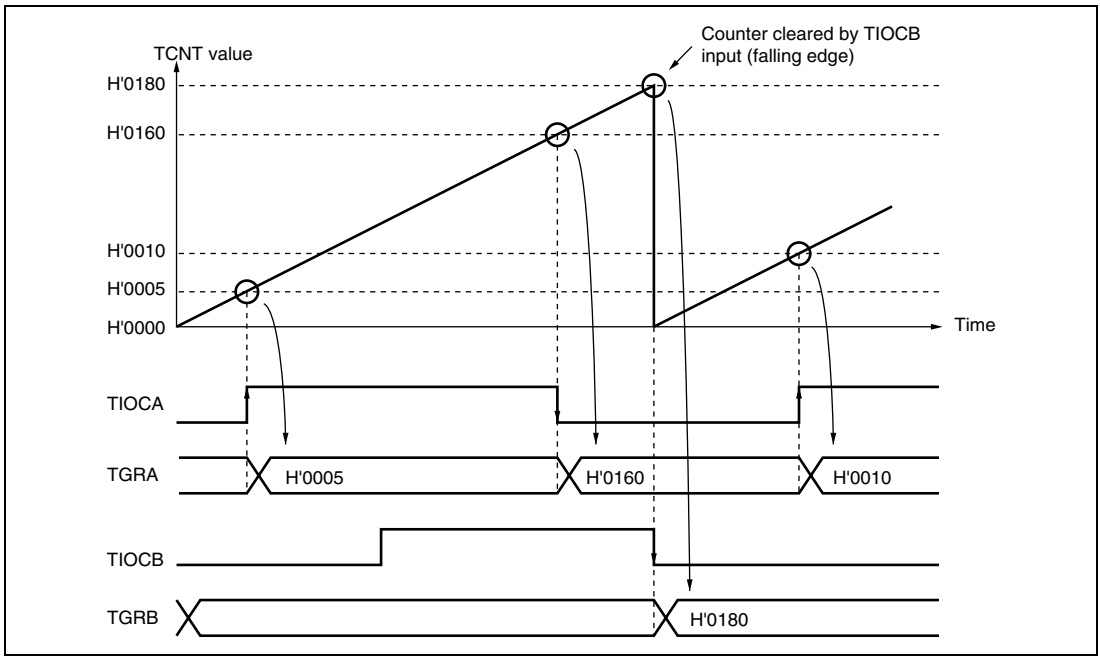


Figure 10.9 Example of Input Capture Operation

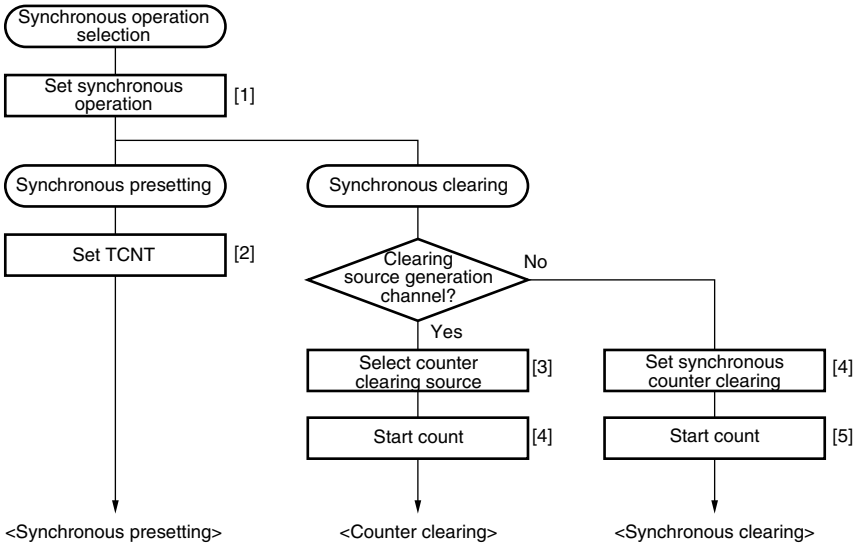
10.4.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 10.10 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use the CCLR2 to CCLR0 bits in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use the CCLR2 to CCLR0 bits in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.10 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 10.11 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 10.4.5, PWM Modes.

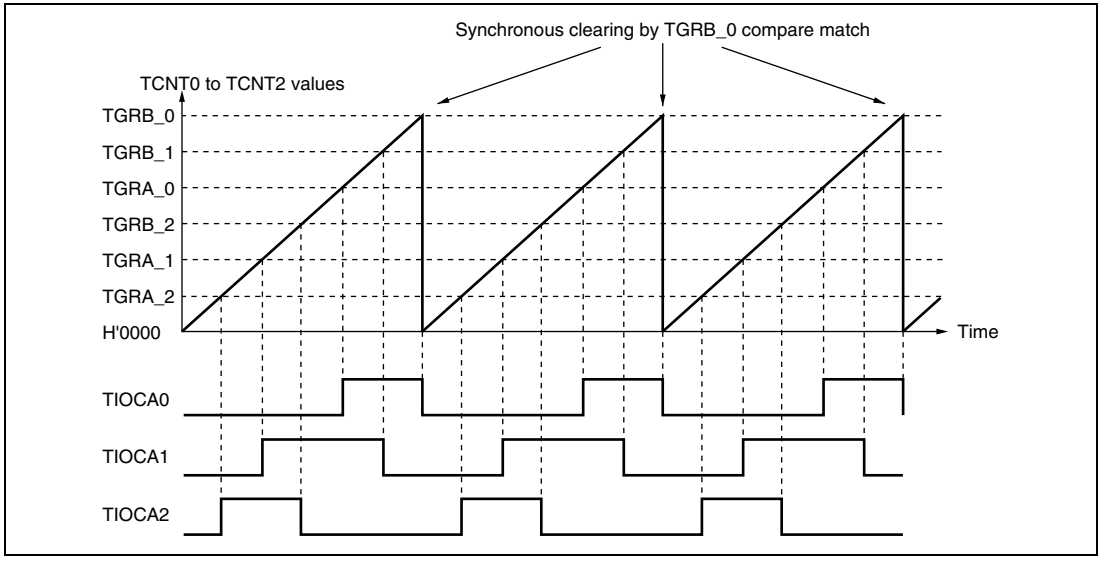


Figure 10.11 Example of Synchronous Operation

10.4.3 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Table 10.28 shows the register combinations used in buffer operation.

Table 10.28 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 10.12.

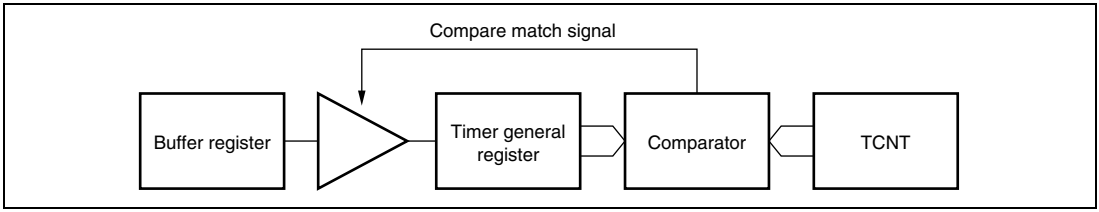


Figure 10.12 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 10.13.

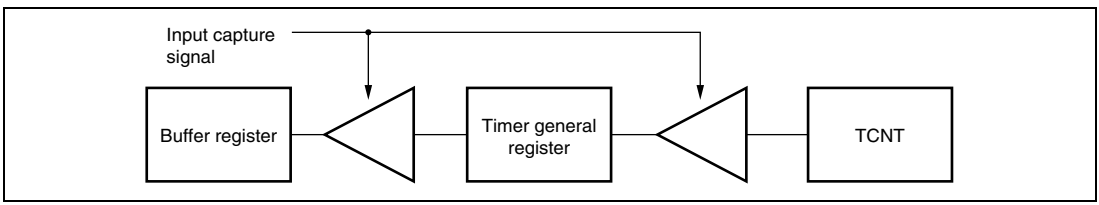


Figure 10.13 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.14 shows an example of the buffer operation setting procedure.

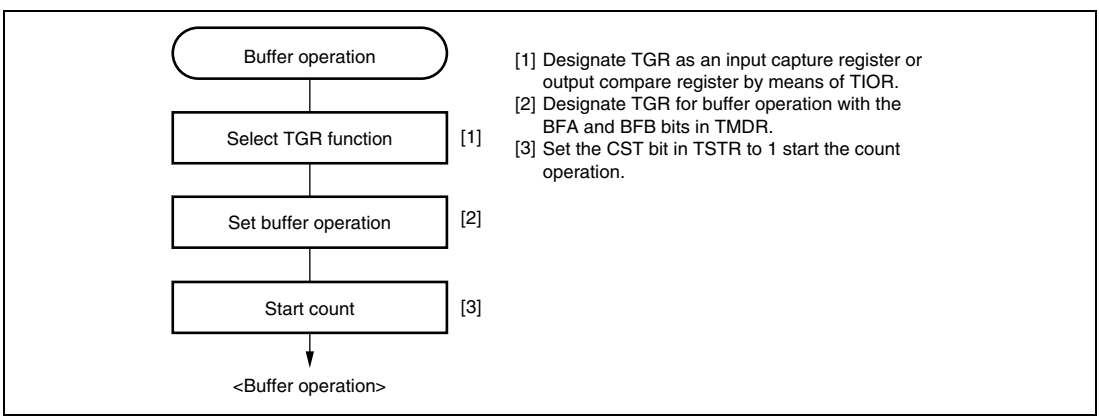


Figure 10.14 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation

1. When TGR is an output compare register

   Figure 10.15 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

   As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time that compare match A occurs.

   For details of PWM modes, see section 10.4.5, PWM Modes.

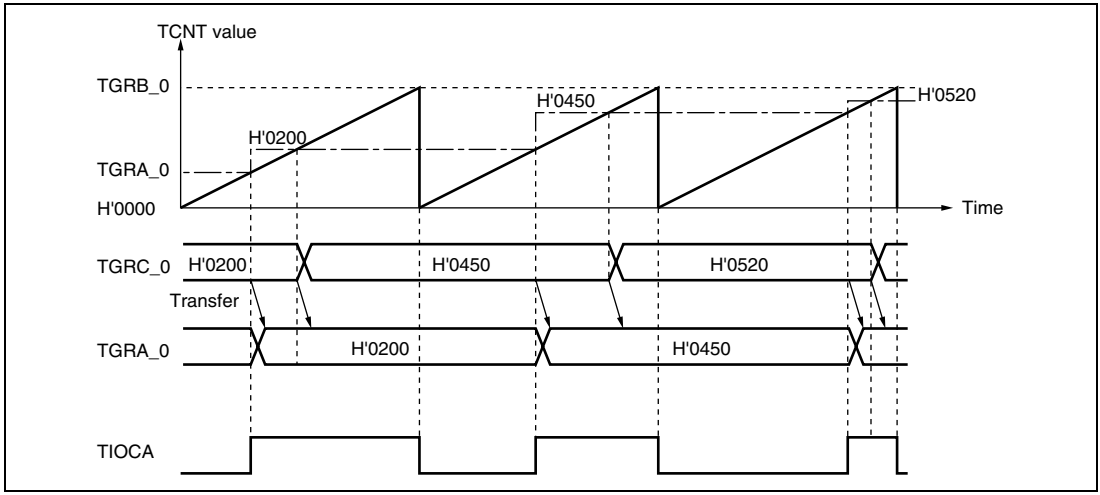


Figure 10.15 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 10.16 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

   Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

   As buffer operation has been set, when the TCNT value is stored in TGRA upon the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

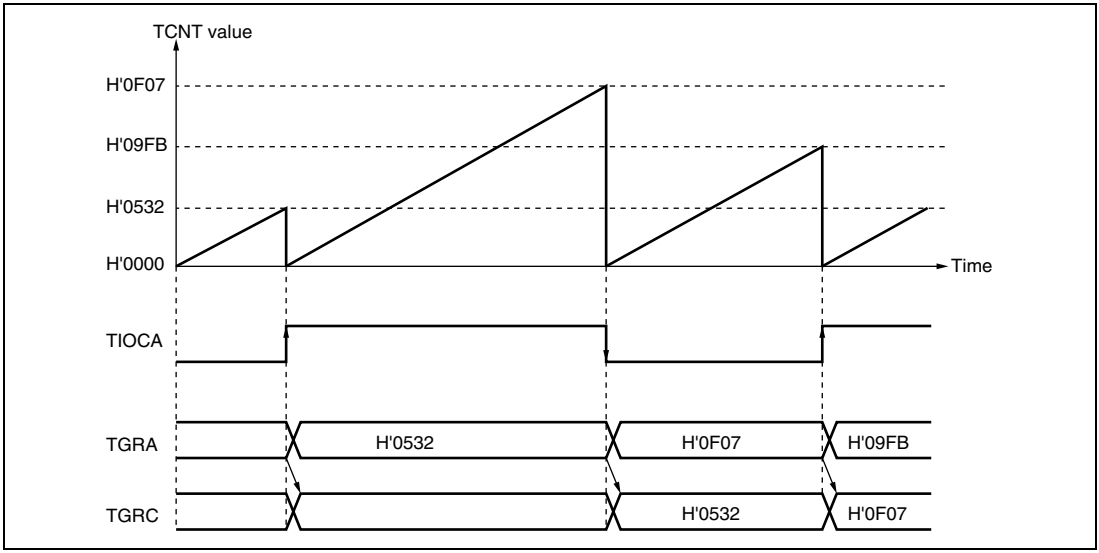


Figure 10.16 Example of Buffer Operation (2)

10.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4) counter clock upon overflow/underflow of TCNT_2 (TCNT_5) as set in the TPSC0 to TPSC2 bits in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10.29 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counters operates independently in phase counting mode.

Table 10.29 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

Example of Cascaded Operation Setting Procedure: Figure 10.17 shows an example of the setting procedure for cascaded operation.

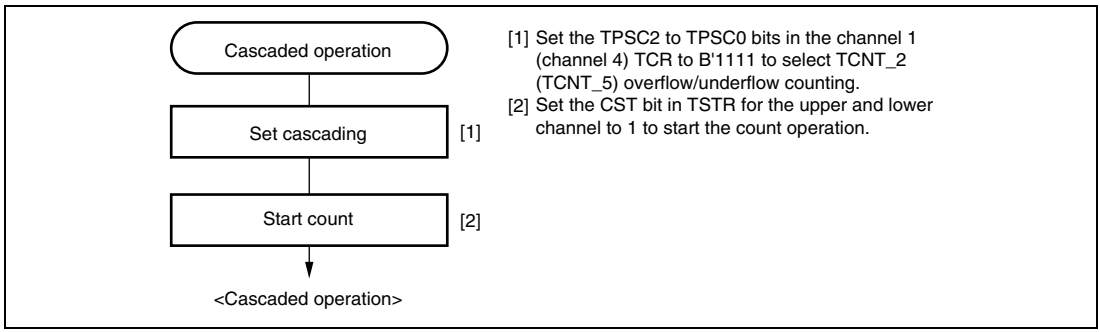


Figure 10.17 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 10.18 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1, when TGRA_1 and TGRA_2 have been designated as input capture registers, and when TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGRA_1, and the lower 16 bits to TGRA_2.

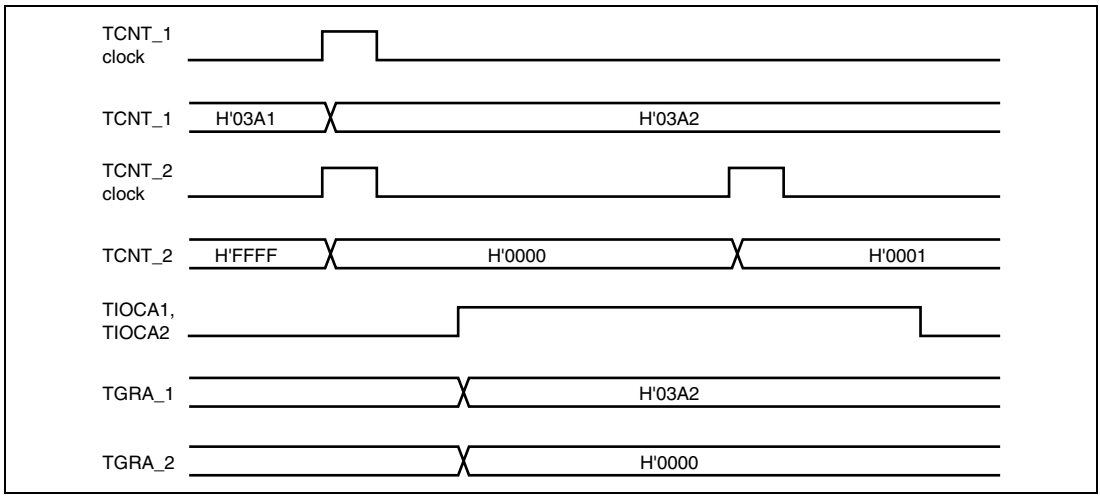


Figure 10.18 Example of Cascaded Operation (1)

Figure 10.19 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1 and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

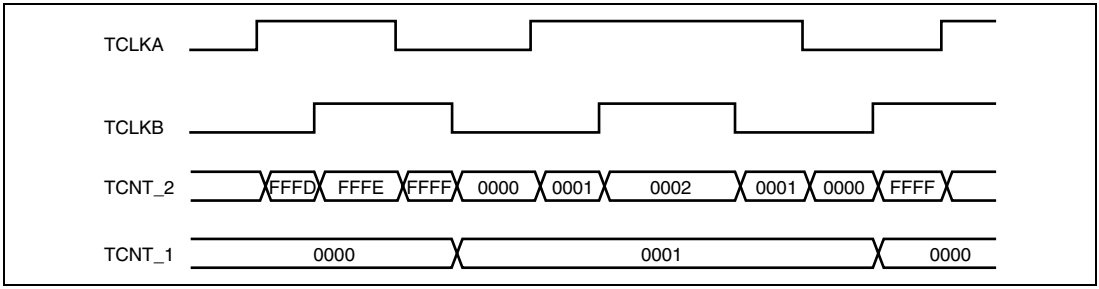


Figure 10.19 Example of Cascaded Operation (2)

10.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by the IOA0 to IOA3 bits and IOC0 to IOC3 bits in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by the IOB0 to IOB3 bits and IOD0 to IOD3 bits in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.30.

Table 10.30 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGR4A_4	TIOCA4	TIOCA4
	TGR4B_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 10.20 shows an example of the PWM mode setting procedure.

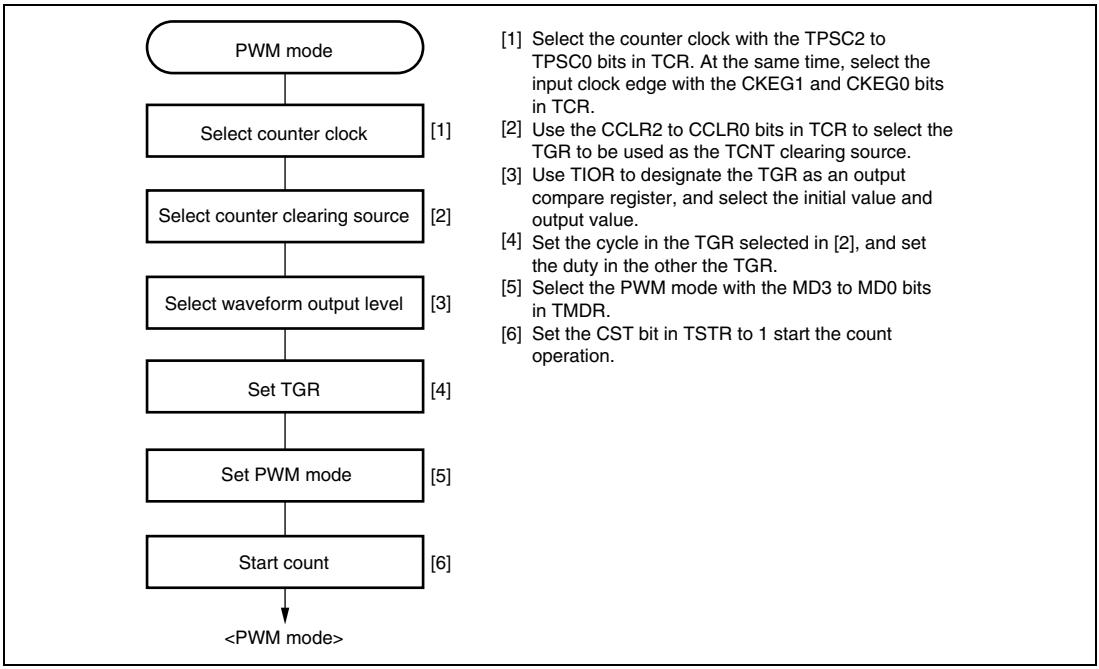


Figure 10.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty levels.

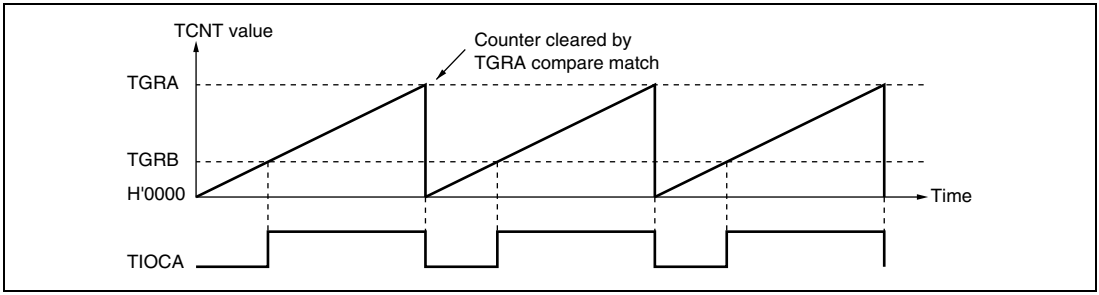


Figure 10.21 Example of PWM Mode Operation (1)

Figure 10.22 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs are used as the duty levels.

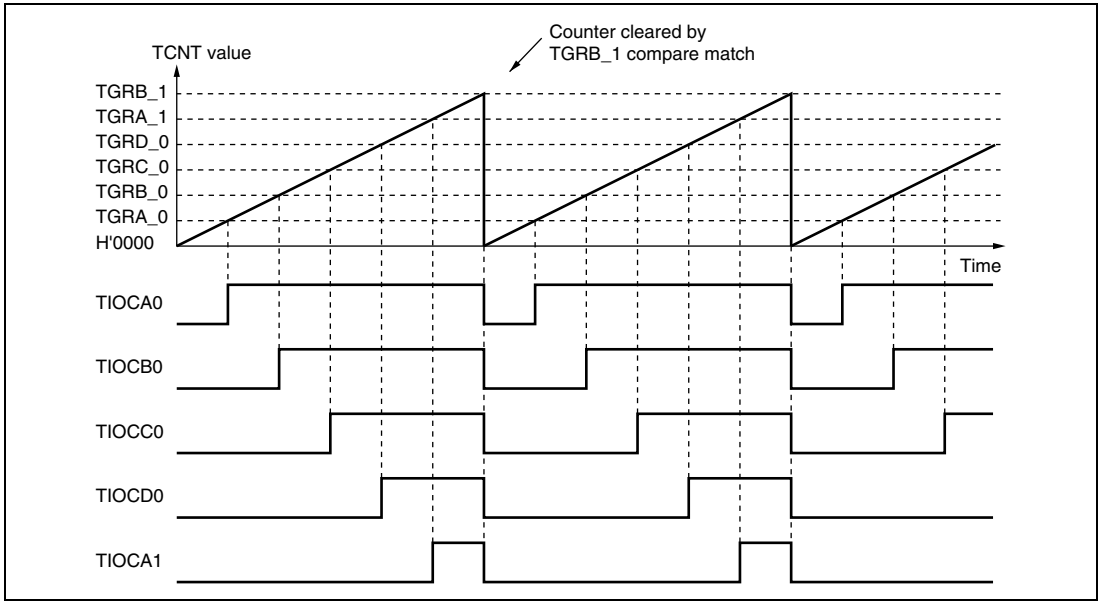


Figure 10.22 Example of PWM Mode Operation (2)

Figure 10.23 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

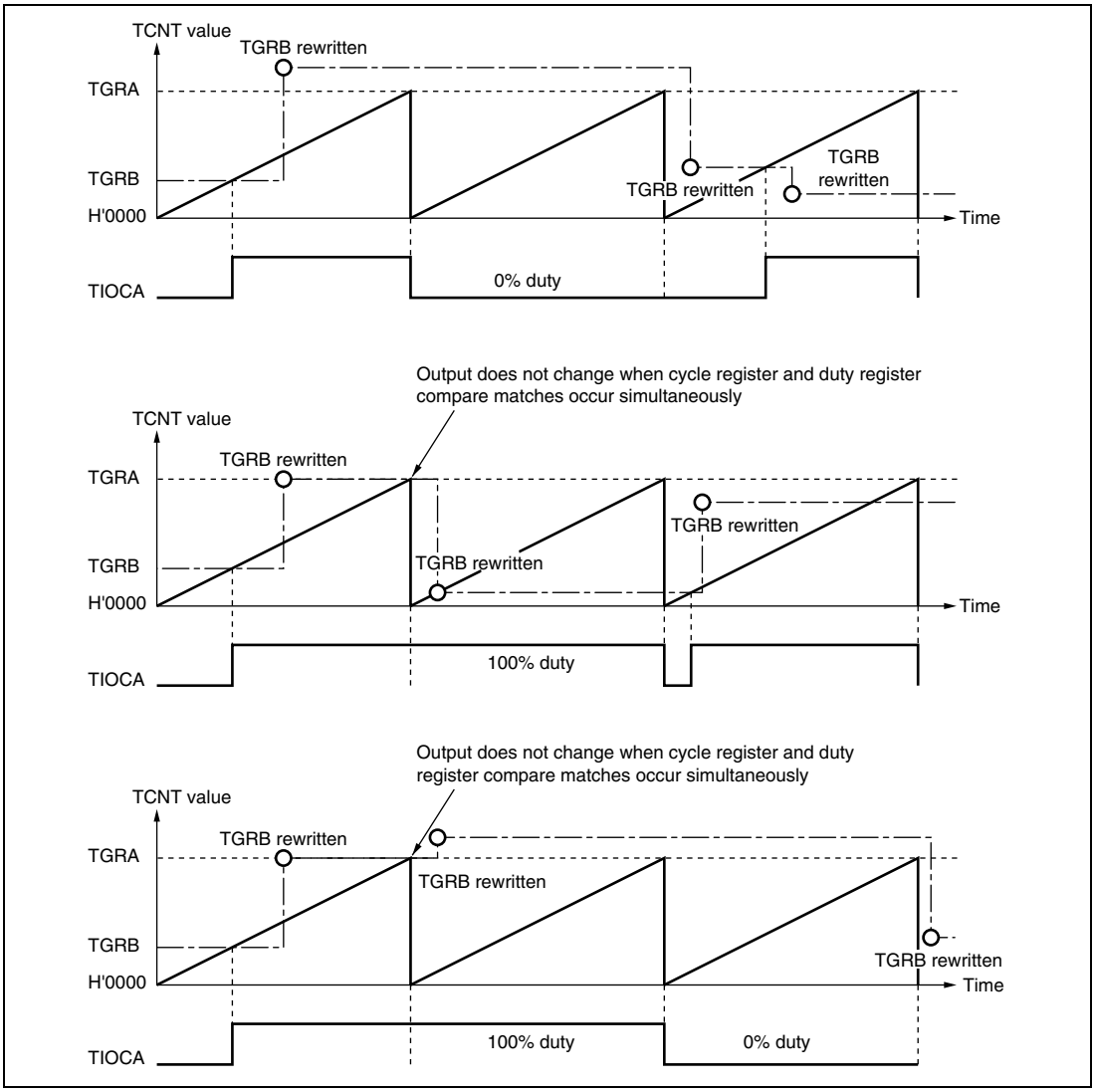


Figure 10.23 Example of PWM Mode Operation (3)

10.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of the TPSC0 to TPSC2 bits and CKEG0 and CKEG1 bits in TCR. However, the functions of the CCLR0 and CCLR1 bits in TCR, and of TIOR, TIER, and TGR, are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 10.31 shows the correspondence between external clock pins and channels.

Table 10.31 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10.24 shows an example of the phase counting mode setting procedure.

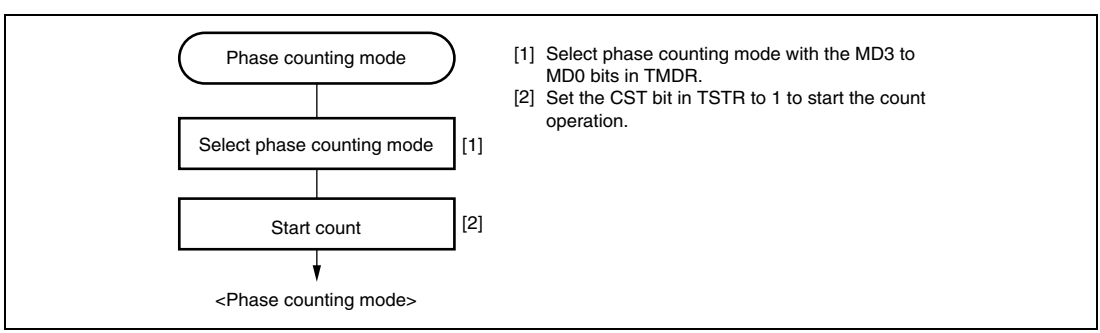


Figure 10.24 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

1. Phase counting mode 1

   Figure 10.25 shows an example of phase counting mode 1 operation, and table 10.32 summarizes the TCNT up/down-count conditions.

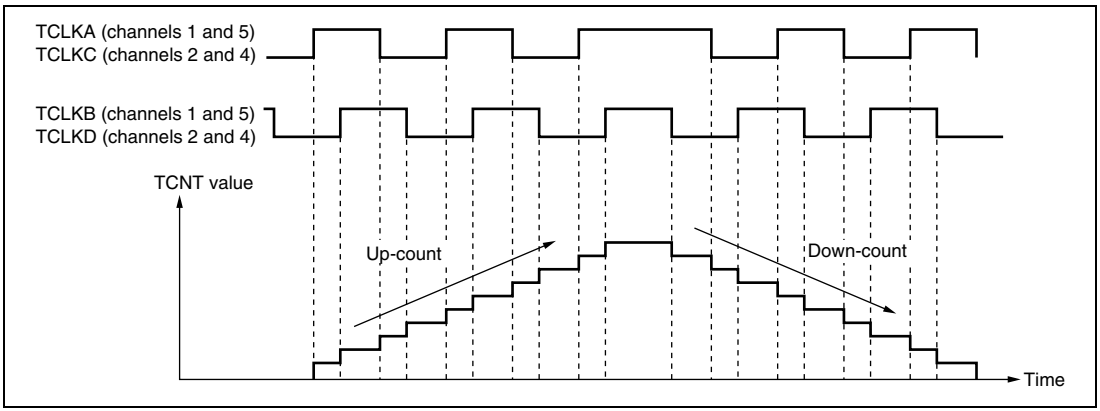


Figure 10.25 Example of Phase Counting Mode 1 Operation

Table 10.32 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

Legend

↑ : Rising edge

↓ : Falling edge

2. Phase counting mode 2

   Figure 10.26 shows an example of phase counting mode 2 operation, and table 10.33 summarizes the TCNT up/down-count conditions.

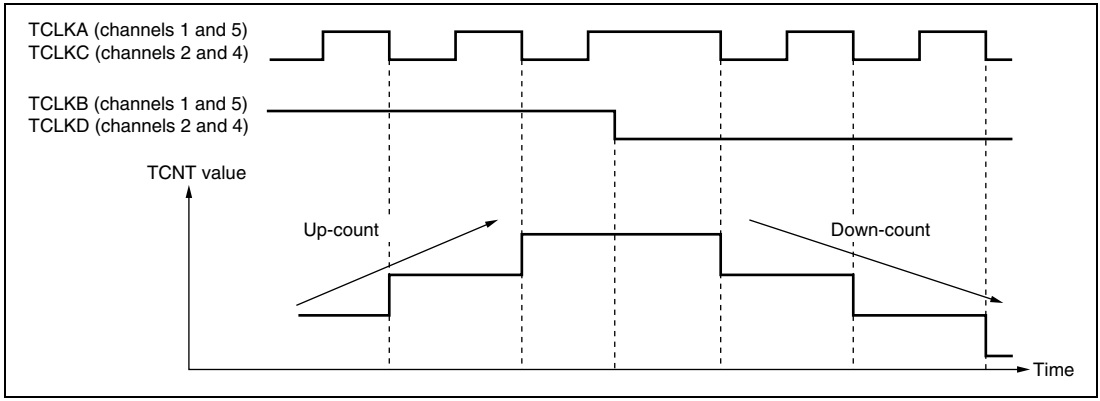


Figure 10.26 Example of Phase Counting Mode 2 Operation

Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

Legend

↑ : Rising edge

↓ : Falling edge

3. Phase counting mode 3

   Figure 10.27 shows an example of phase counting mode 3 operation, and table 10.34 summarizes the TCNT up/down-count conditions.

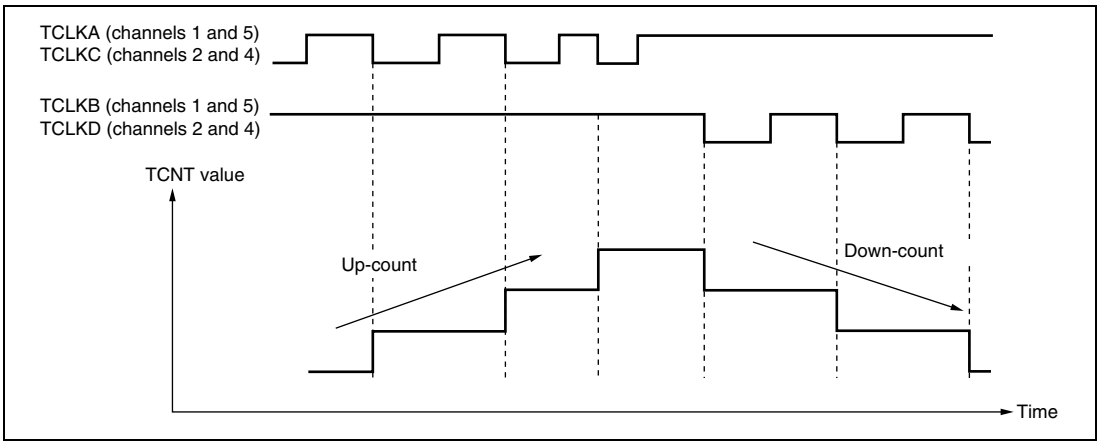


Figure 10.27 Example of Phase Counting Mode 3 Operation

Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	⌐↑	Don't care
Low level	⌐↓	Don't care
⌐↑	Low level	Don't care
⌐↓	High level	Up-count
High level	⌐↓	Down-count
Low level	⌐↑	Don't care
⌐↑	High level	Don't care
⌐↓	Low level	Don't care

Legend

⌐↑ : Rising edge

⌐↓ : Falling edge

4. Phase counting mode 4

   Figure 10.28 shows an example of phase counting mode 4 operation, and table 10.35 summarizes the TCNT up/down-count conditions.

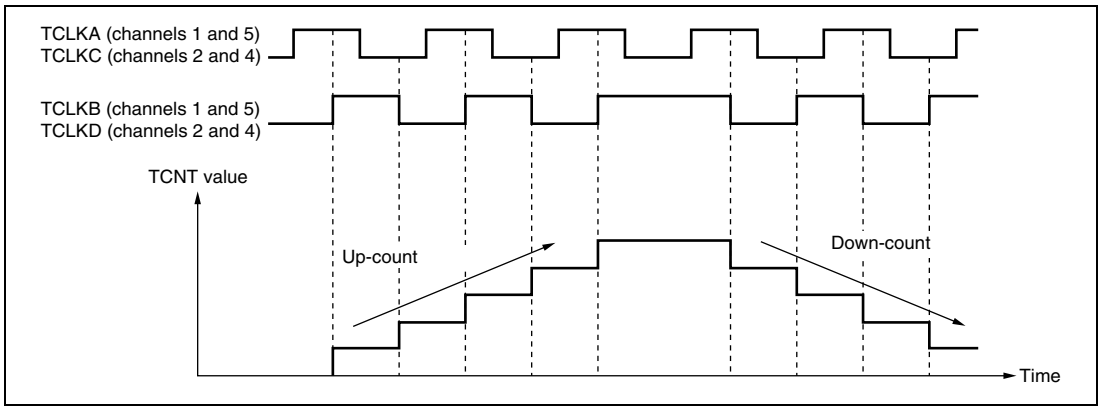


Figure 10.28 Example of Phase Counting Mode 4 Operation

Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Legend

↑ : Rising edge

↓ : Falling edge

Phase Counting Mode Application Example: Figure 10.29 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function and are set with the speed control period and position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

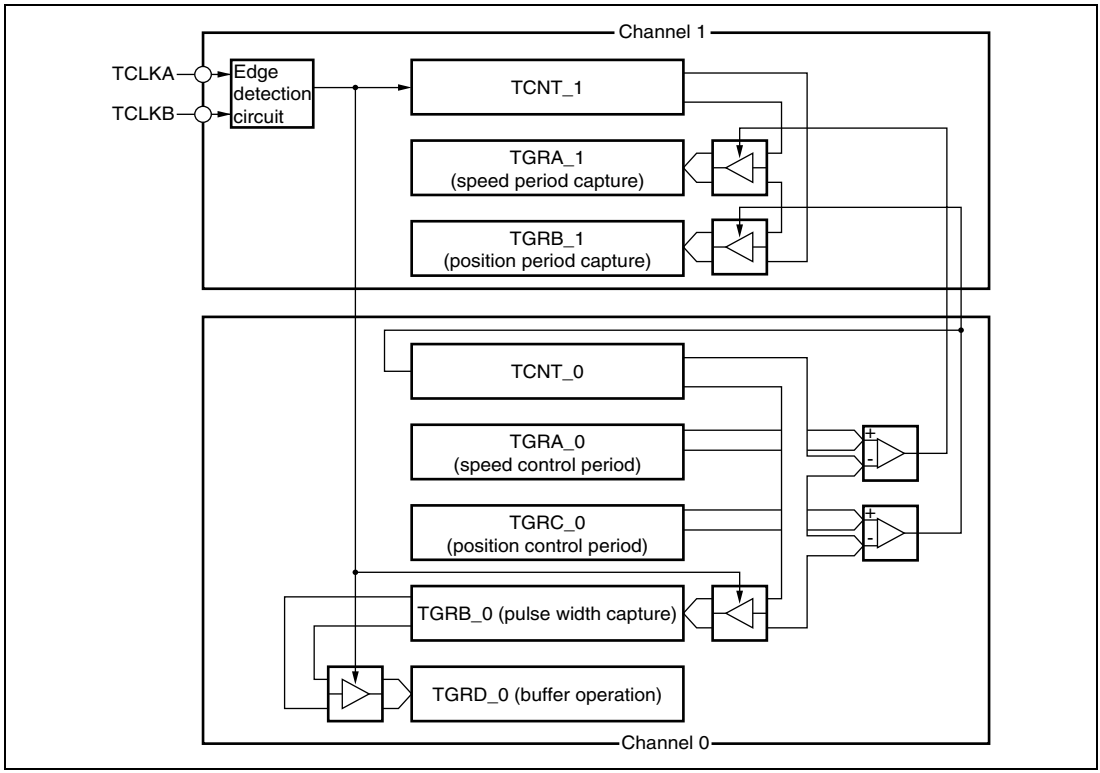


Figure 10.29 Phase Counting Mode Application Example

10.5 Interrupts

There are three kinds of TPU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.36 lists the TPU interrupt sources.

Table 10.36 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Possible
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Possible
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Possible
	TCI0V	TCNT_0 overflow	TCFV_0	Not possible
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Possible
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Possible
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Possible
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Possible
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	Possible
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	Possible
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	Possible
	TCI3V	TCNT_3 overflow	TCFV_3	Not possible
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Possible
	TGI4B	TGRB_4 input capture/compare match	TGFB_4	Possible
	TCI4V	TCNT_4 overflow	TCFV_4	Not possible
	TCI4U	TCNT_4 underflow	TCFU_4	Not possible
5	TGI5A	TGRA_5 input capture/compare match	TGFA_5	Possible
	TGI5B	TGRB_5 input capture/compare match	TGFB_5	Possible
	TCI5V	TCNT_5 overflow	TCFV_5	Not possible
	TCI5U	TCNT_5 underflow	TCFU_5	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2, 4, and 5.

10.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 8, Data Transfer Controller (DTC).

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

10.7 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to begin A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is begun.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.8 Operation Timing

10.8.1 Input/Output Timing

TCNT Count Timing: Figure 10.30 shows TCNT count timing in internal clock operation, and figure 10.31 shows TCNT count timing in external clock operation.

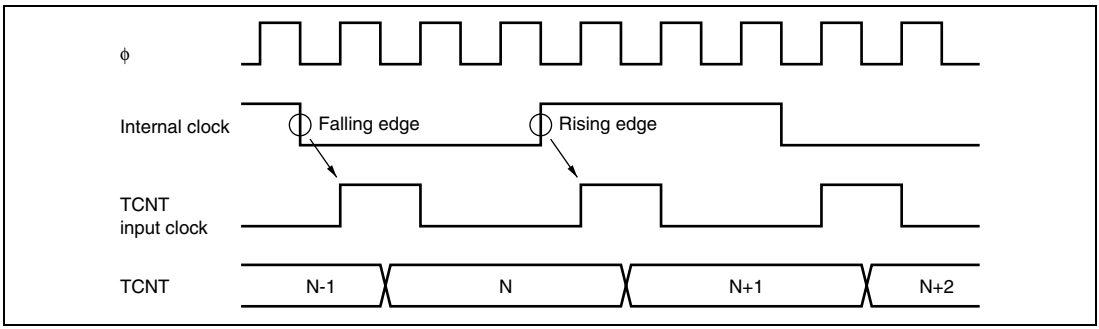


Figure 10.30 Count Timing in Internal Clock Operation

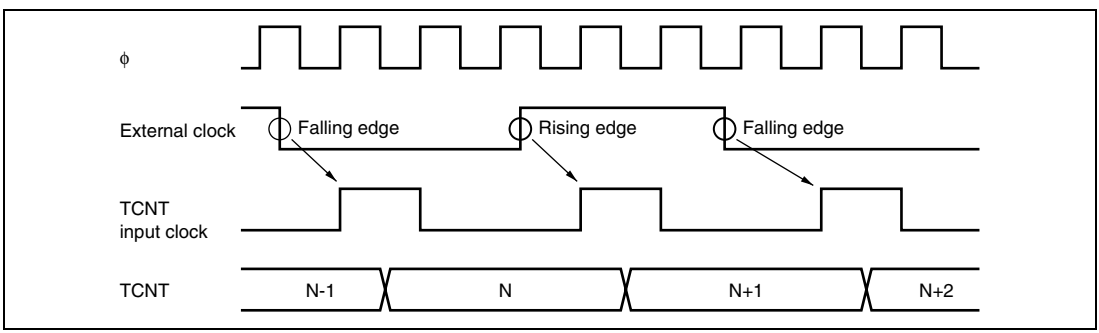


Figure 10.31 Count Timing in External Clock Operation

Output Compare Output Timing: A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10.32 shows output compare output timing.

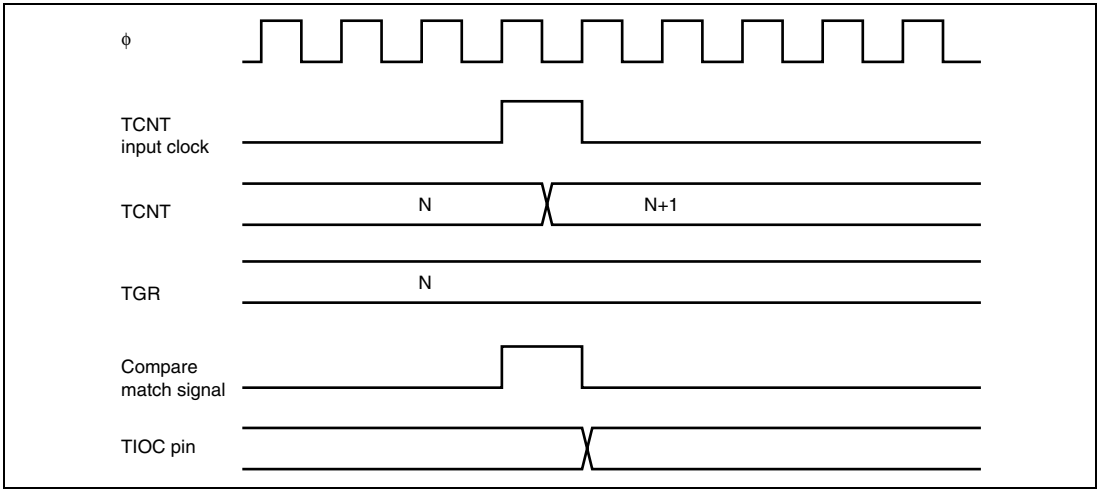


Figure 10.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.33 shows input capture signal timing.

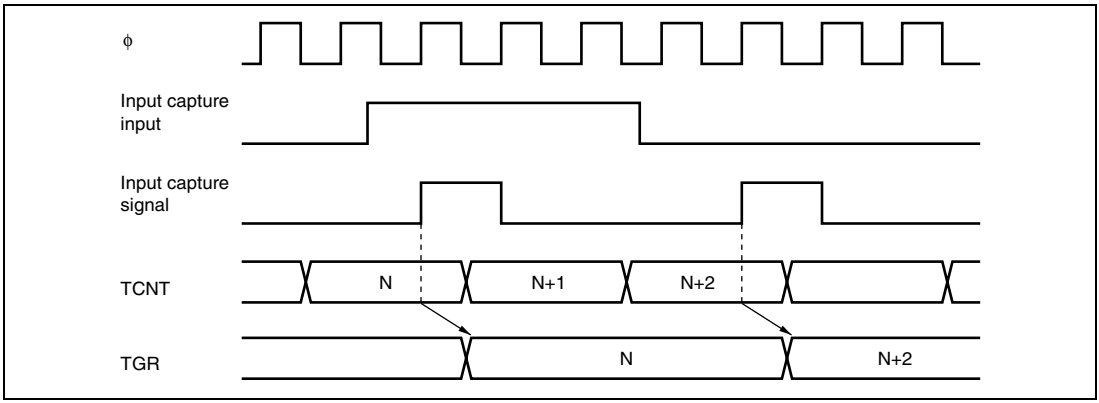


Figure 10.33 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 10.34 shows the timing when counter clearing on compare match is specified, and figure 10.35 shows the timing when counter clearing on input capture is specified.

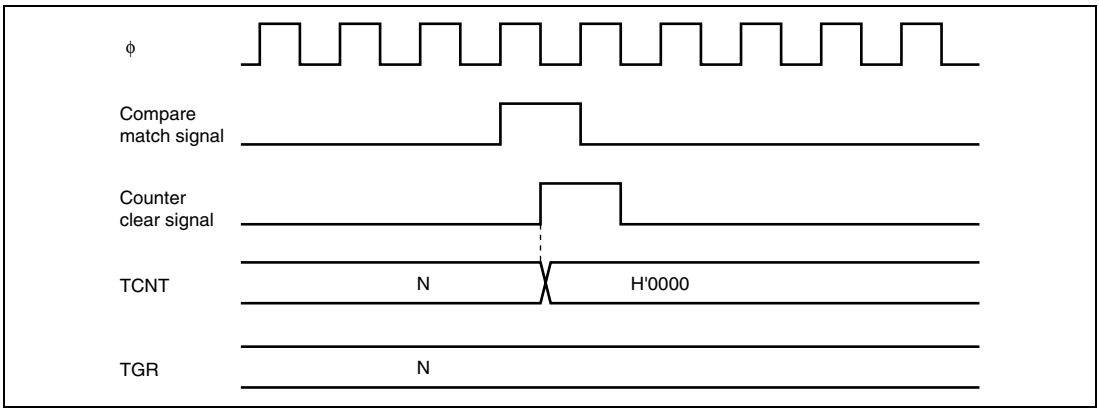


Figure 10.34 Counter Clear Timing (Compare Match)

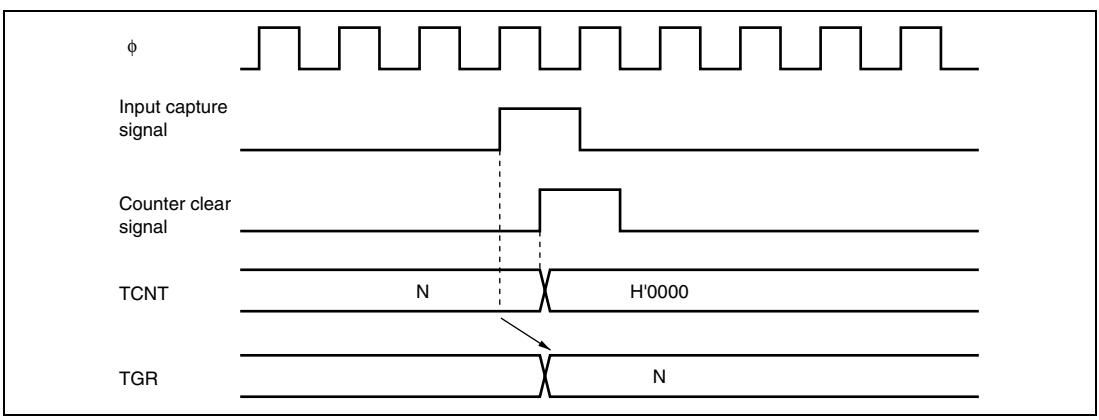


Figure 10.35 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 10.36 and 10.37 show the timing in buffer operation.

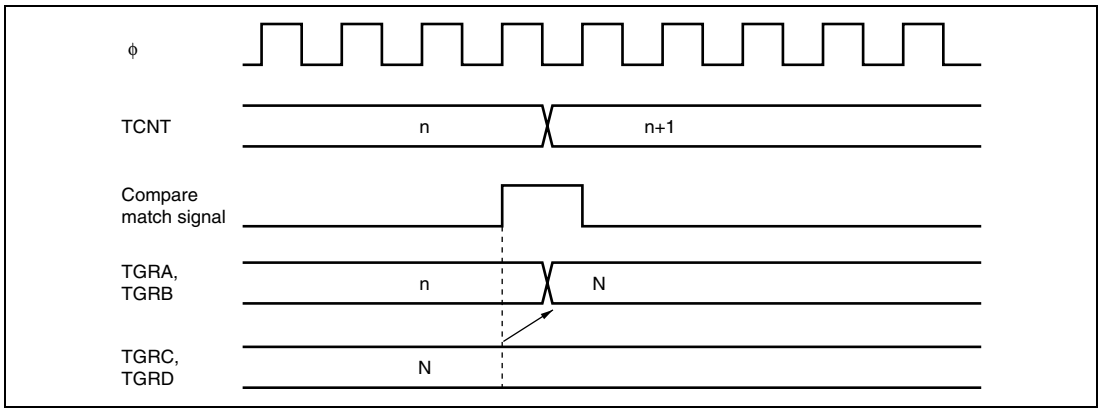


Figure 10.36 Buffer Operation Timing (Compare Match)

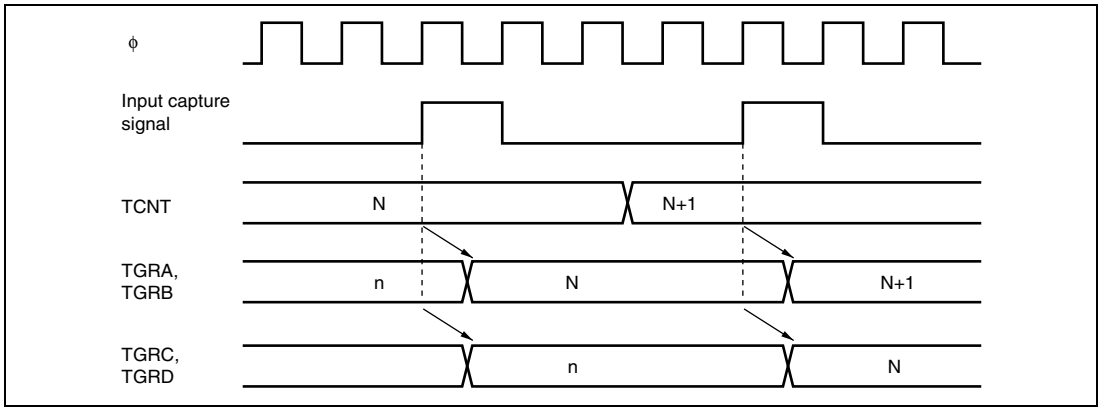


Figure 10.37 Buffer Operation Timing (Input Capture)

10.8.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 10.38 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

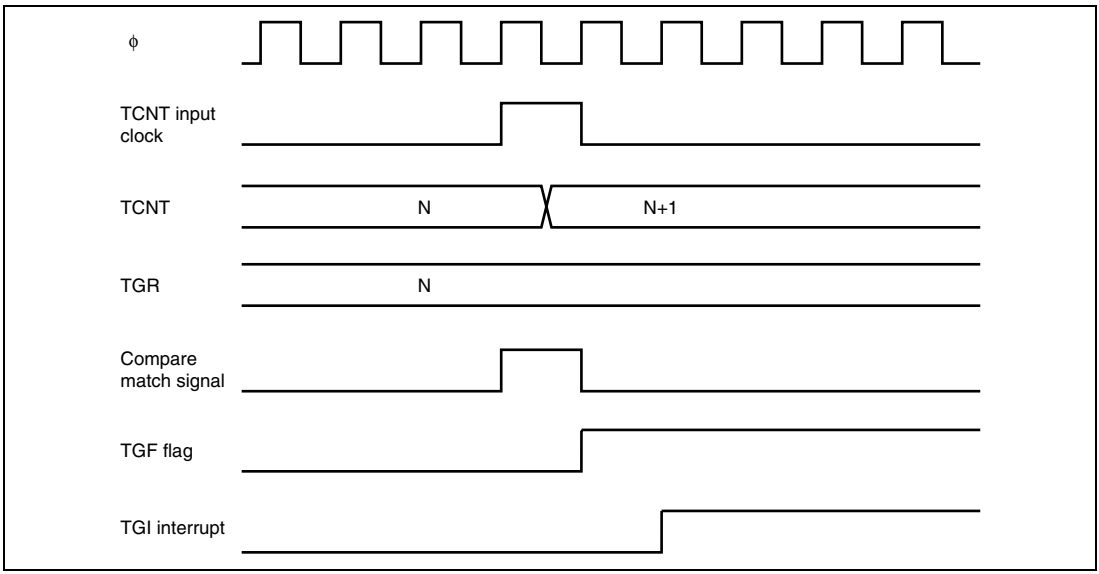


Figure 10.38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 10.39 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

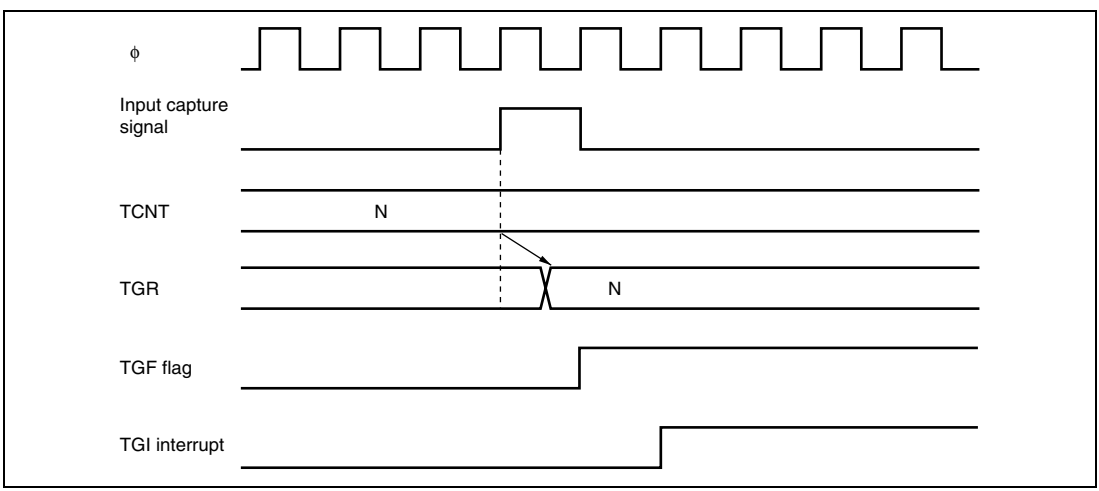


Figure 10.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 10.40 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.41 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

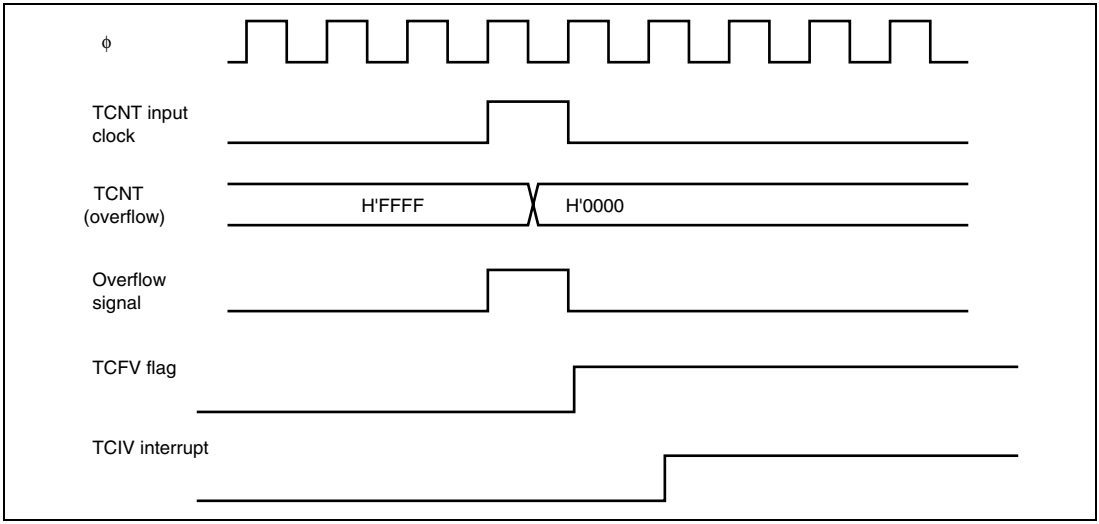


Figure 10.40 TCIV Interrupt Setting Timing

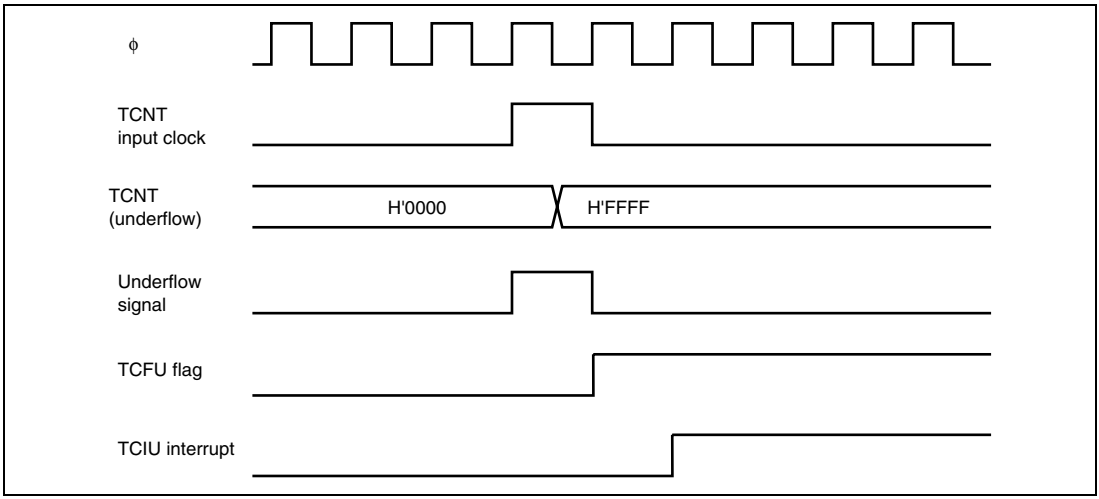


Figure 10.41 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC is activated, the flag is cleared automatically. Figure 10.42 shows the timing for status flag clearing by the CPU, and figure 10.43 shows the timing for status flag clearing by the DTC.

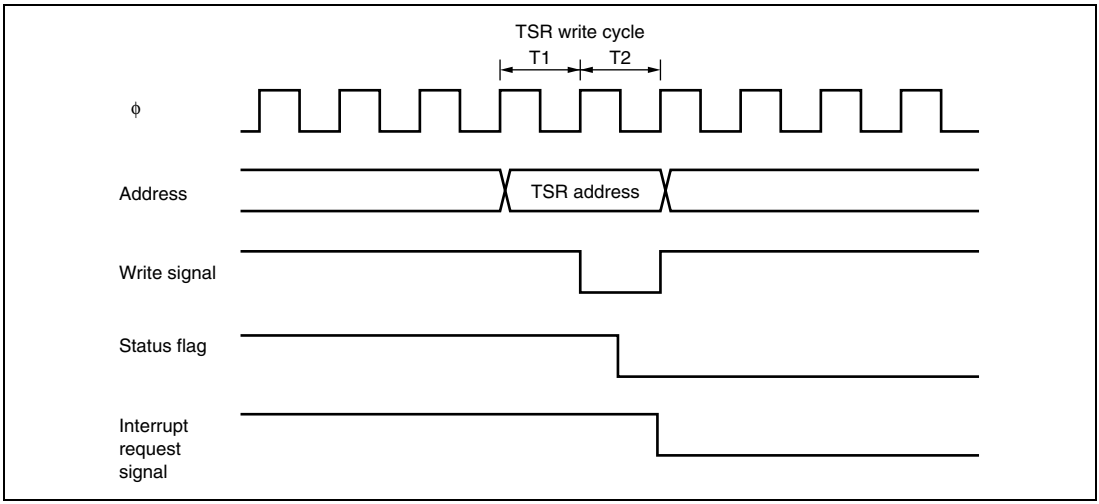


Figure 10.42 Timing for Status Flag Clearing by CPU

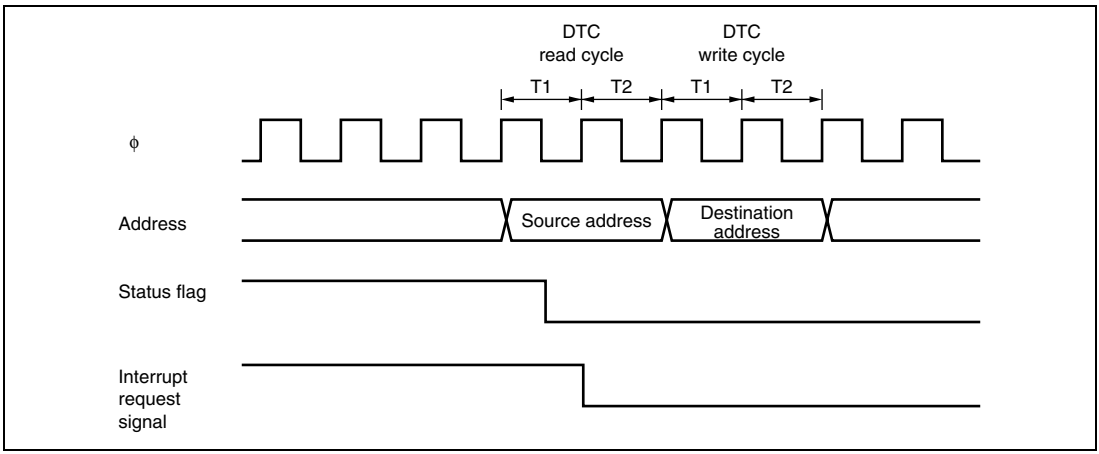


Figure 10.43 Timing for Status Flag Clearing by DTC Activation

10.9 Usage Notes

10.9.1 Module Stop Mode Setting

TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

10.9.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.44 shows the input clock conditions in phase counting mode.

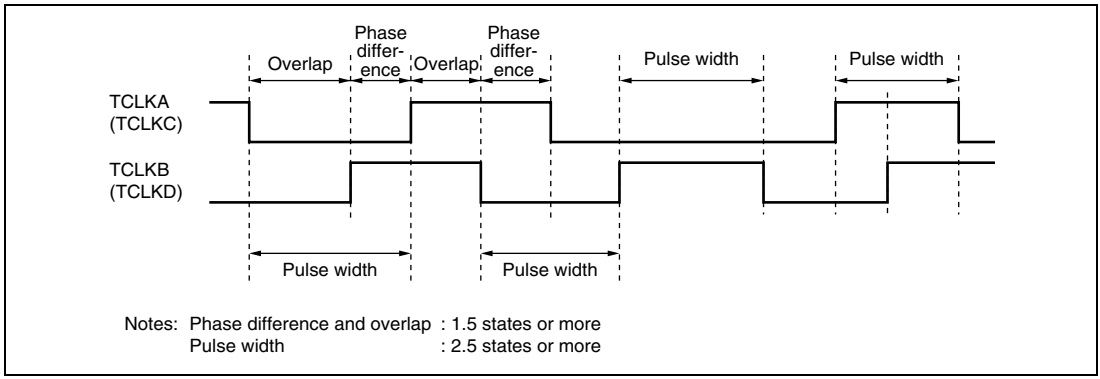


Figure 10.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

10.9.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f : Counter frequency
ϕ : Operating frequency
N : TGR set value

10.9.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10.45 shows the timing in this case.

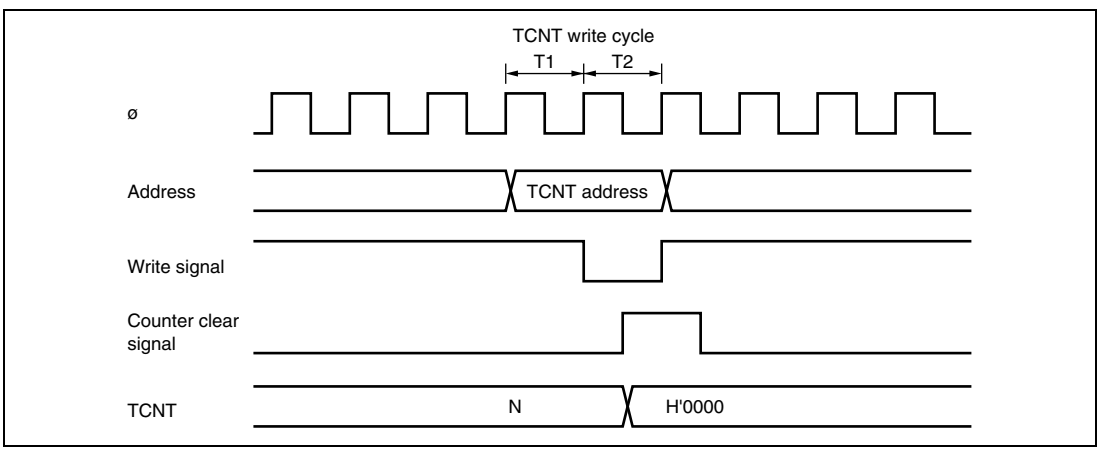


Figure 10.45 Contention between TCNT Write and Clear Operations

10.9.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10.46 shows the timing in this case.

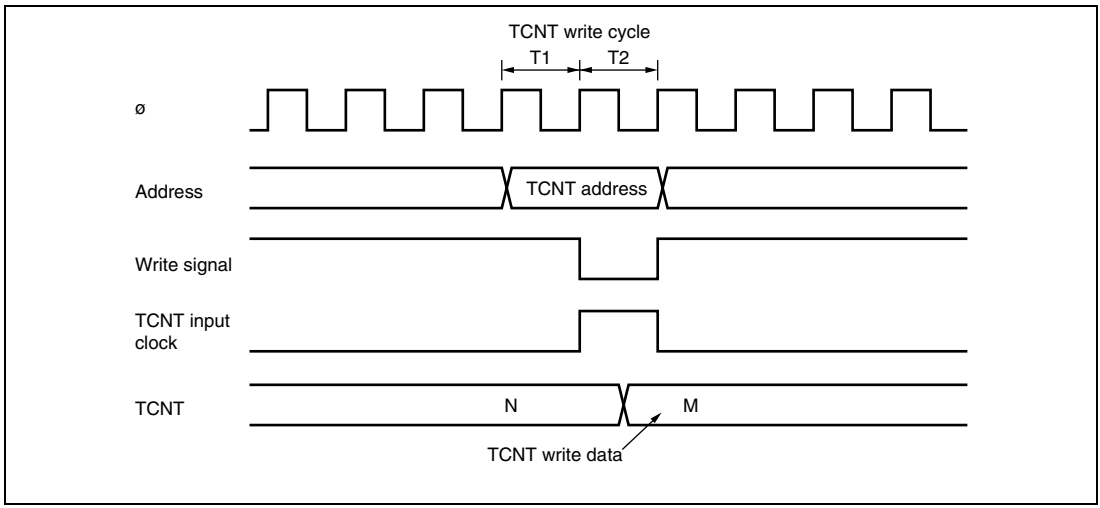


Figure 10.46 Contention between TCNT Write and Increment Operations

10.9.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the previous value is written.

Figure 10.47 shows the timing in this case.

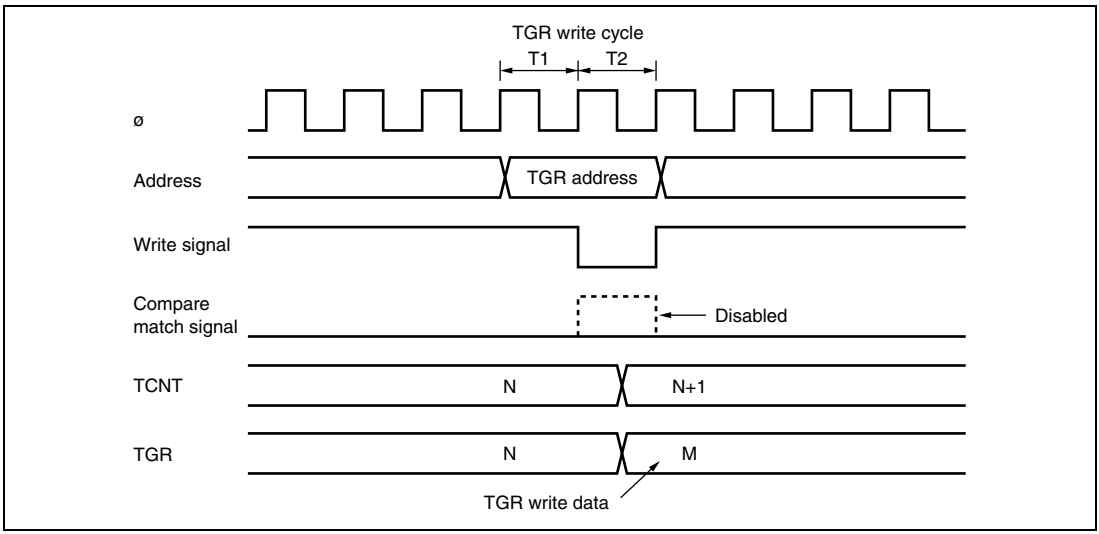


Figure 10.47 Contention between TGR Write and Compare Match

10.9.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation will be that in the buffer prior to the write.

Figure 10.48 shows the timing in this case.

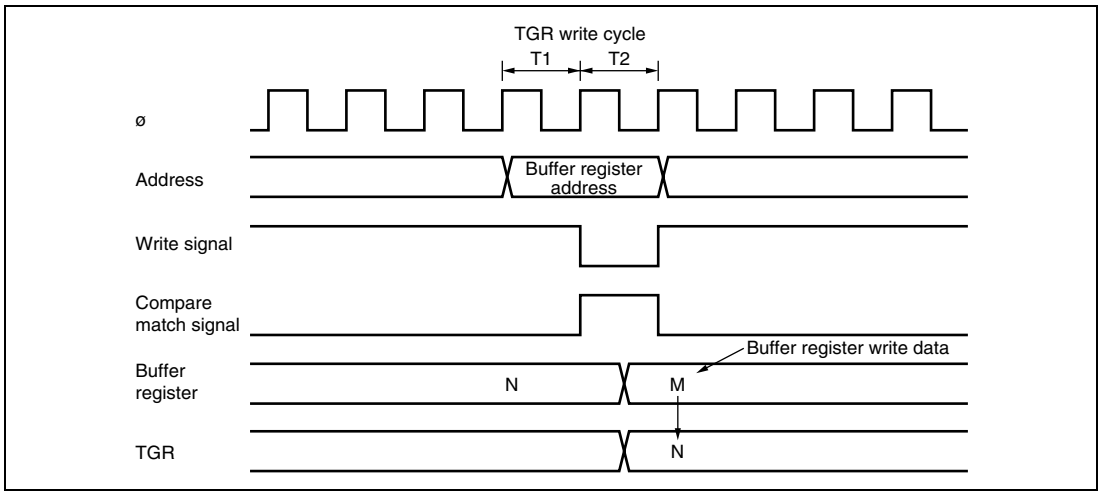


Figure 10.48 Contention between Buffer Register Write and Compare Match

10.9.8 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be that in the buffer after input capture transfer.

Figure 10.49 shows the timing in this case.

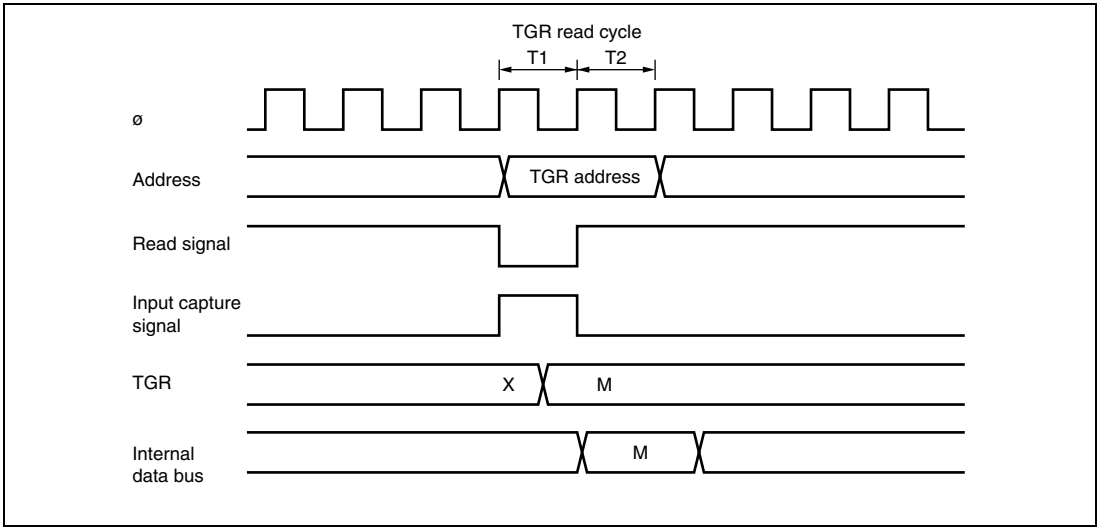


Figure 10.49 Contention between TGR Read and Input Capture

10.9.9 Contention between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10.50 shows the timing in this case.

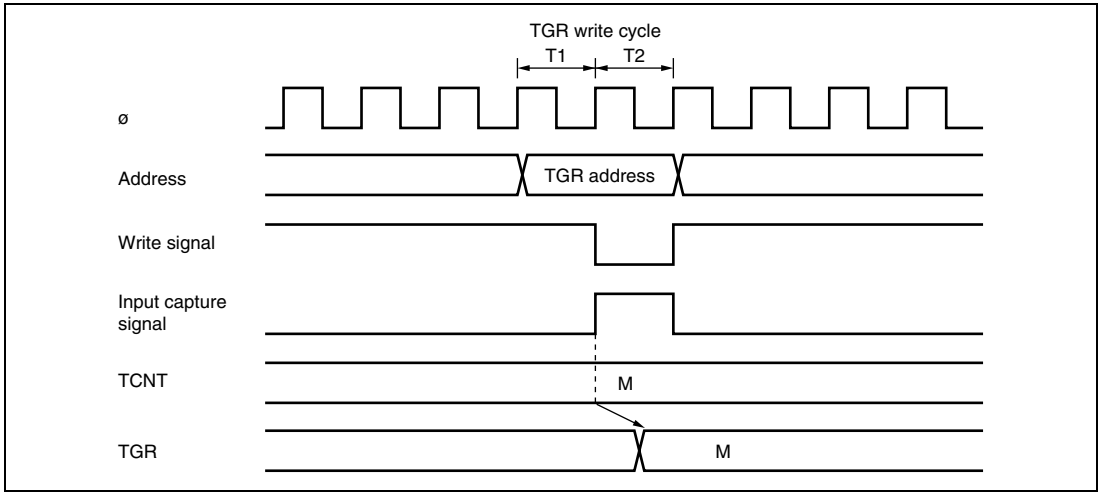


Figure 10.50 Contention between TGR Write and Input Capture

10.9.10 Contention between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.51 shows the timing in this case.

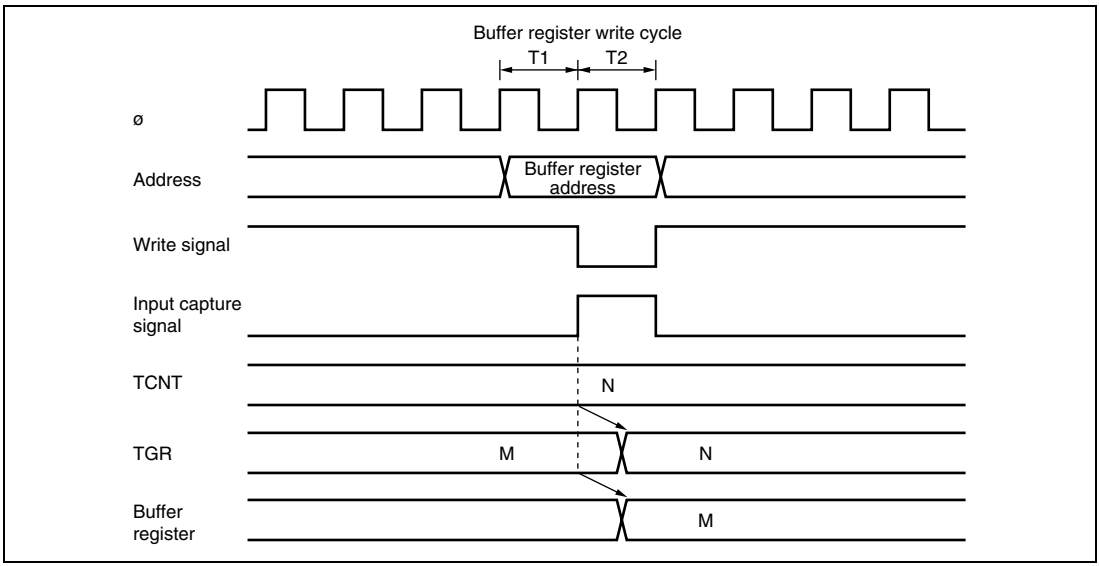


Figure 10.51 Contention between Buffer Register Write and Input Capture

10.9.11 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 10.52 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

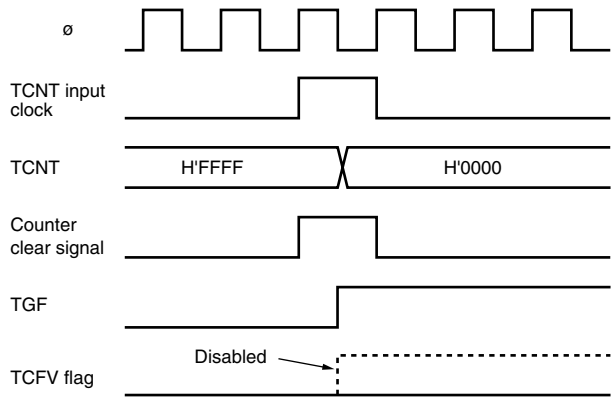


Figure 10.52 Contention between Overflow and Counter Clearing

10.9.12 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.53 shows the operation timing when there is contention between TCNT write and overflow.

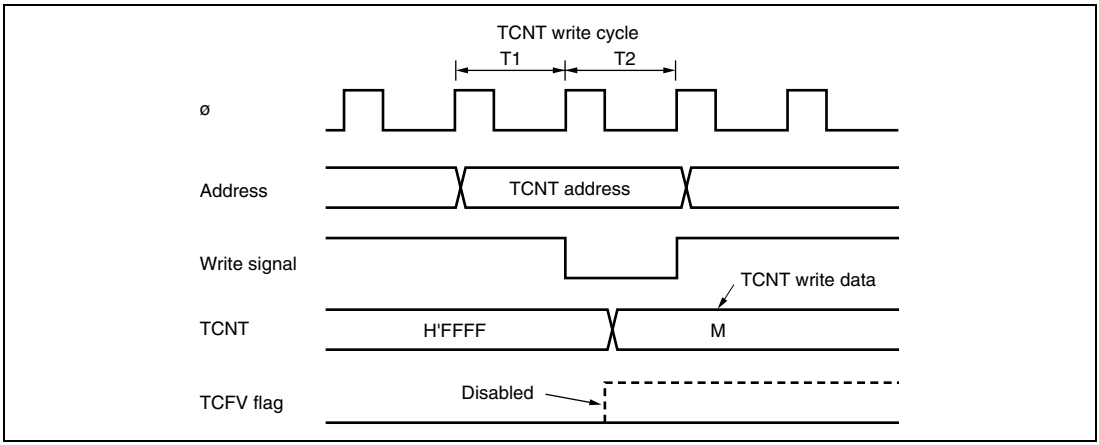


Figure 10.53 Contention between TCNT Write and Overflow

10.9.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

10.9.14 Interrupts in Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 11 8-Bit Timers (TMR)

This LSI has an on-chip 8-bit timer module with four channels. The 8-bit timer module can be used to count external events and be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with an arbitrary duty cycle using a compare-match signal with two registers.

11.1 Features

- Selection of four clock sources

  Selected from three internal clocks (ϕ/8, ϕ/64, and ϕ/8192) and an external clock
- Selection of three ways to clear the counters

  The counters can be cleared on compare-match A or B, or by an external reset signal
- 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 be used for various applications, such as the generation of pulse output or PWM output with an arbitrary duty cycle
- Cascading of the two channels

  (Cascading of TMR_0, TMR_1)

  The module can operate as a 16-bit timer using TMR_0 as the upper half and TMR_1 as the lower half (16-bit count mode)

  TMR_1 can be used to count TMR_0 compare-match occurrences (compare-match count mode)

  (Cascading of TMR_2, TMR_3)

  The module can operate as a 16-bit timer using TMR_2 as the upper half and TMR_3 as the lower half (16-bit count mode)

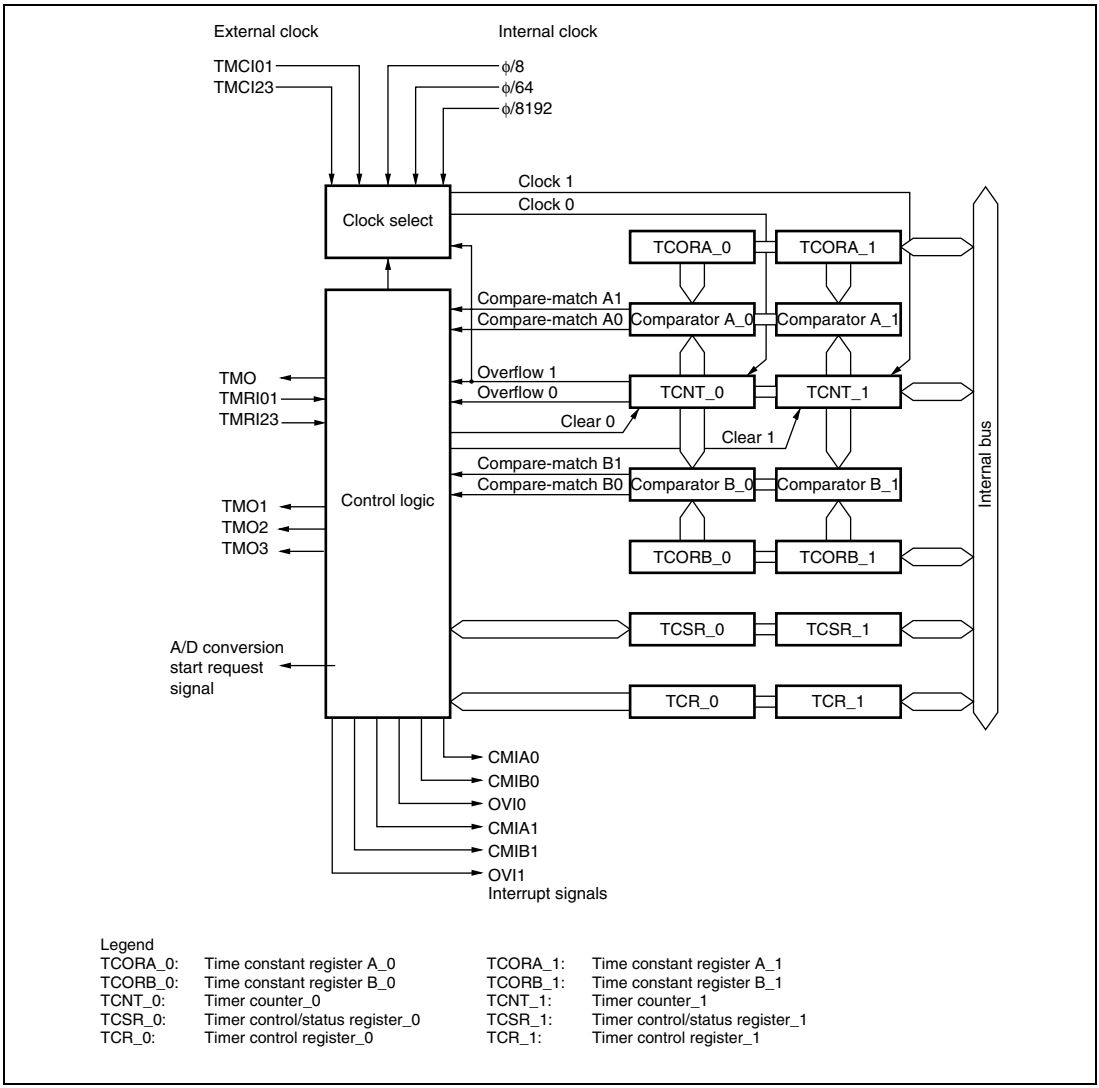
  TMR_3 can be used to count TMR_2 compare-match occurrences (compare-match count mode)
- Multiple interrupt sources for each channel

  Two compare-match interrupts and one overflow interrupt can be requested independently
- Generation of A/D converter conversion start trigger

  Channel 0 compare-match signal can be used as the A/D converter conversion start trigger
- Module stop mode can be set

  As the initial setting, the 8-bit timer operation is halted. Register access is enabled by canceling the module stop mode.

Figure 11.1 shows a block diagram of the 8-bit timer module (TMR_0 and TMR_1).



Legend

TCORA_0:	Time constant register A_0	TCORA_1:	Time constant register A_1
TCORB_0:	Time constant register B_0	TCORB_1:	Time constant register B_1
TCNT_0:	Timer counter_0	TCNT_1:	Timer counter_1
TCSR_0:	Timer control/status register_0	TCSR_1:	Timer control/status register_1
TCR_0:	Timer control register_0	TCR_1:	Timer control register_1

Figure 11.1 Block Diagram of 8-Bit Timer Module

11.2 Input/Output Pins

Table 11.1 summarizes the input and output pins of the 8-bit timer module.

Table 11.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Timer output	TMO0	Output	Output controlled by compare-match
1	Timer output	TMO1	Output	Output controlled by compare-match
Common to 0 and 1	Timer clock input	TMCI01	Input	External clock input for the counter
	Timer reset input	TMRI01	Input	External reset input for the counter
2	Timer output	TMO2	Output	Output controlled by compare-match
3	Timer output	TMO3	Output	Output controlled by compare-match
Common to 2 and 3	Timer clock input	TMCI23	Input	External clock input for the counter
	Timer reset input	TMRI23	Input	External reset input for the counter

11.3 Register Descriptions

The 8-bit timer has the following registers. For details on the module stop register, refer to section 22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

Channel 0

- Timer counter_0 (TCNT_0)
- Time constant register A_0 (TCORA_0)
- Time constant register B_0 (TCORB_0)
- Timer control register_0 (TCR_0)
- Timer control/status register_0 (TCSR_0)

Channel 1

- Timer counter_1 (TCNT_1)
- Time constant register A_1 (TCORA_1)
- Time constant register B_1 (TCORB_1)
- Timer control register_1 (TCR_1)
- Timer control/status register_1 (TCSR_1)

Channel 2

- Timer counter_2 (TCNT_2)
- Time constant register A_2 (TCORA_2)
- Time constant register B_2 (TCORB_2)
- Timer control register_2 (TCR_2)

- Timer control/status register_2 (TCSR_2)

Channel 3

- Timer counter_3 (TCNT_3)
- Time constant register A_3 (TCORA_3)
- Time constant register B_3 (TCORB_3)
- Timer control register_3 (TCR_3)
- Timer control/status register_3 (TCSR_3)

11.3.1 Timer Counter (TCNT)

Each TCNT is an 8-bit up-counter. TCNT_0 and TCNT_1 (or TCNT_2 and TCNT_3) comprise a single 16-bit register, so they can be accessed together by word access.

This clock source is selected by the clock select bits, CKS2 to CKS0, in TCR. TCNT can be cleared by an external reset input signal or compare-match signals A and B. The CCLR1 and CCLR0 bits in TCR select the method of TCNT clearing.

When TCNT overflows from H'FF to H'00, the overflow flag (OVF) in TCSR is set to 1. The initial value of TCNT is H'00.

11.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 (or TCORA_2 and TCORA_3) comprise a single 16-bit register, so they can be accessed together by word access.

TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag A (CMFA) in TCSR is set. Note, however, that comparison is disabled during the T_2 state of a TCORA write cycle.

The timer output from the TMO pin can be freely controlled by the compare-match signal A and the settings of the output select bits, OS1 and OS0, in TCSR.

The initial value of TCORA is H'FF.

11.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_1 (or TCORB_2 and TCORB_3) comprise a single 16-bit register, so they can be accessed together by word access.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag B (CMFB) in TCSR is set. Note, however, that comparison is disabled during the T_2 state of a TCORB write cycle.

The timer output from the TMO pin can be freely controlled by the compare-match signal B and the settings of the output select bits, OS1 and OS0, in TCSR.

The initial value of TCORB is H'FF.

11.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the time at which TCNT is cleared, and controls interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare-Match Interrupt Enable B Selects whether the CMFB interrupt request (CMIB) is enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt request (CMIB) is disabled 1: CMFB interrupt request (CMIB) is enabled
6	CMIEA	0	R/W	Compare-Match Interrupt Enable A Selects whether the CMFA interrupt request (CMIA) is enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt request (CMIA) is disabled 1: CMFA interrupt request (CMIA) is enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether the OVF interrupt request (OVI) is enabled or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	These bits select the method by which TCNT is cleared 00: Clearing is disabled 01: Cleared on compare-match A 10: Cleared on compare-match B 11: Cleared on rising edge of external reset input

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	The input clock can be selected from three clocks divided from the system clock (φ). 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.
0	CKS0	0	R/W	000: Clock input disabled 001: φ/8 internal clock source, counted on the falling edge 010: φ/64 internal clock source, counted on the falling edge 011: φ/8192 internal clock source, counted on the falling edge 100: For channel 0: Counted on TCNT1 overflow signal* For channel 1: Counted on TCNT0 compare-match A signal* For channel 2: Counted on TCNT3 overflow signal* For channel 3: Counted on TCNT2 compare-match A signal* 101: External clock source, counted at rising edge 110: External clock source, counted at falling edge 111: External clock source, counted at both rising and falling edges

Note: * If the count input of channel 0 (channel 2) is the TCNT1 (TCNT3) overflow signal and that of channel 1 (channel 3) is the TCNT0 (TCNT2) compare-match signal, no incrementing clock will be generated. Do not use this setting.

11.3.5 Timer Control/Status Register (TCSR)

TCSR indicates status flags and controls compare-match output.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*1	Compare-Match Flag B [Setting condition] When TCNT = TCORB [Clearing condition] • Read CMFB when CMFB = 1, then write 0 in CMFB. • The DTC is activated by the CMIB interrupt and the DISEL bit = 0 in MRB of the DTC.
6	CMFA	0	R/(W)*1	Compare-Match Flag A [Setting condition] When TCNT = TCORA [Clearing condition] • Read CMFA when CMFA = 1, then write 0 in CMFA. • The DTC is activated by the CMIA interrupt and DISEL bit = 0 in MRB of the DTC.
5	OVF	0	R/(W)*1	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1*2, then write 0 in OVF
4	ADTE	0	R/W	A/D Trigger Enable Enables or disables A/D converter start requests by compare-match A. 0: A/D converter start requests by compare-match A are disabled 1: A/D converter start requests by compare-match A are enabled

Bit	Bit Name	Initial Value	R/W	Description
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the timer output level is to be changed by a compare-match B of TCORB and TCNT. 00: No change when compare-match B occurs 01: 0 is output when compare-match B occurs 10: 1 is output when compare-match B occurs 11: Output is inverted when compare-match B occurs (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the timer output level is to be changed by a compare-match A of TCORA and TCNT. 00: No change when compare-match A occurs 01: 0 is output when compare-match A occurs 10: 1 is output when compare-match A occurs 11: Output is inverted when compare-match A occurs (toggle output)

Notes: 1. Only 0 can be written to this bit, to clear the flag.
2. When the interval timer interrupt is disabled and the OVF is polled, read state of OVF = 1 twice or more.

- TCSR_1 and TCSR_3

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When TCNT = TCORB [Clearing condition] • Read CMFB when CMFB = 1, then write 0 in CMFB. • The DTC is activated by the CMIB interrupt and the DISEL Bit = 0 in MRB of the DTC.
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When TCNT = TCORA [Clearing condition] • Read CMFA when CMFA = 1, then write 0 in CMFA. • The DTC is activated by the CMIA interrupt and the DISEL Bit = 0 in MRB of the DTC.
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
3	OS3	0	R/W	Output Select 3 and 2 These bits specify how the timer output level is to be changed by a compare-match B of TCORB and TCNT. 00: No change when compare-match B occurs 01: 0 is output when compare-match B occurs 10: 1 is output when compare-match B occurs 11: Output is inverted when compare-match B occurs (toggle output)
2	OS2	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the timer output level is to be changed by a compare-match A of TCORA and TCNT. 00: No change when compare-match A occurs 01: 0 is output when compare-match A occurs 10: 1 is output when compare-match A occurs 11: Output is inverted when compare-match A occurs (toggle output)

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

- TCSR_2

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When TCNT = TCORB [Clearing condition] • Read CMFB when CMFB = 1, then write 0 in CMFB. • The DTC is activated by the CMIB interrupt and the DISEL Bit = 0 in MRB of the DTC.
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When TCNT = TCORA [Clearing condition] • Read CMFA when CMFA = 1, then write 0 in CMFA. • The DTC is activated by the CMIA interrupt and the DISEL Bit = 0 in MRB of the DTC.
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF

Bit	Bit Name	Initial Value	R/W	Description
4	—	0	R/W	Reserved This bit is a readable/writable bit, but the write value should always be 0.
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the timer output level is to be changed by a compare-match B of TCORB and TCNT. 00: No change when compare-match B occurs 01: 0 is output when compare-match B occurs 10: 1 is output when compare-match B occurs 11: Output is inverted when compare-match B occurs (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the timer output level is to be changed by a compare-match A of TCORA and TCNT. 00: No change when compare-match A occurs 01: 0 is output when compare-match A occurs 10: 1 is output when compare-match A occurs 11: Output is inverted when compare-match A occurs (toggle output)

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

11.4 Operation

11.4.1 Pulse Output

Figure 11.2 shows an example of arbitrary duty pulse output.

1. Set the CCR1 bit in TCR to 0 and the CCLR0 bit to 1 to clear TCNT by a TCORA compare-match.
2. Set the OS3 to OS0 bits in TCSR to B'0110 to output 1 by a TCORA compare-match and 0 by a TCORB compare-match.

By the above settings, waveforms with the cycle of TCORA and the pulse width of TCORB can be output without software intervention.

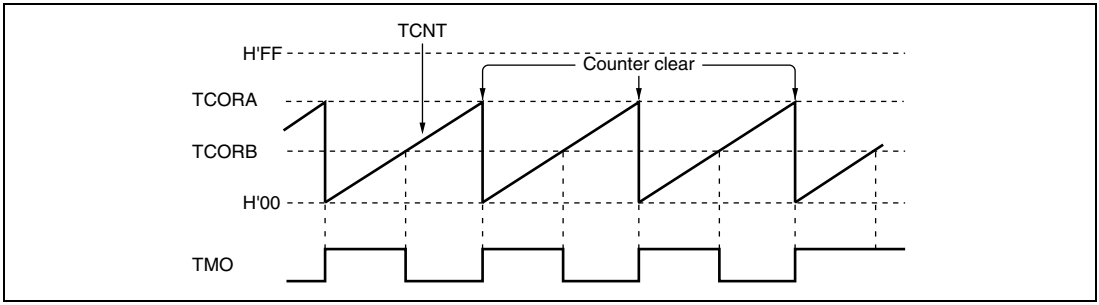


Figure 11.2 Example of Pulse Output

11.5 Operation Timing

11.5.1 TCNT Incrementation Timing

Figure 11.3 shows the TCNT count timing with internal clock source. Figure 11.4 shows the TCNT incrementation timing with external clock source. The pulse width of the external clock for incrementation at single edge must be at least 1.5 status, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

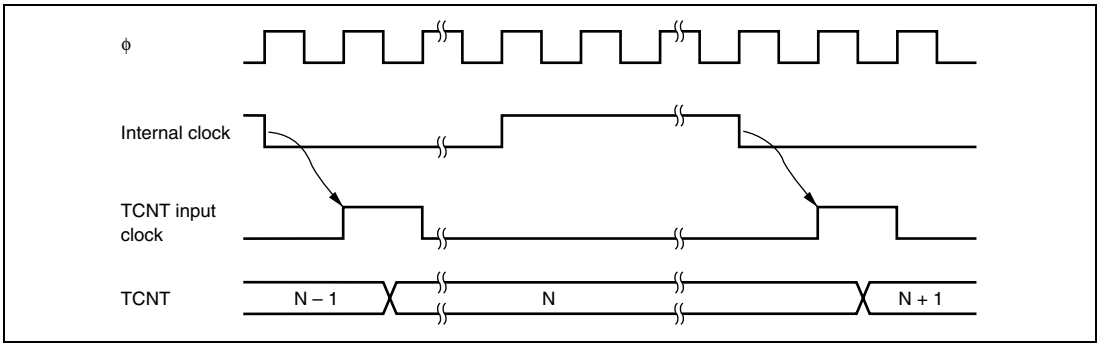


Figure 11.3 Count Timing for Internal Clock Input

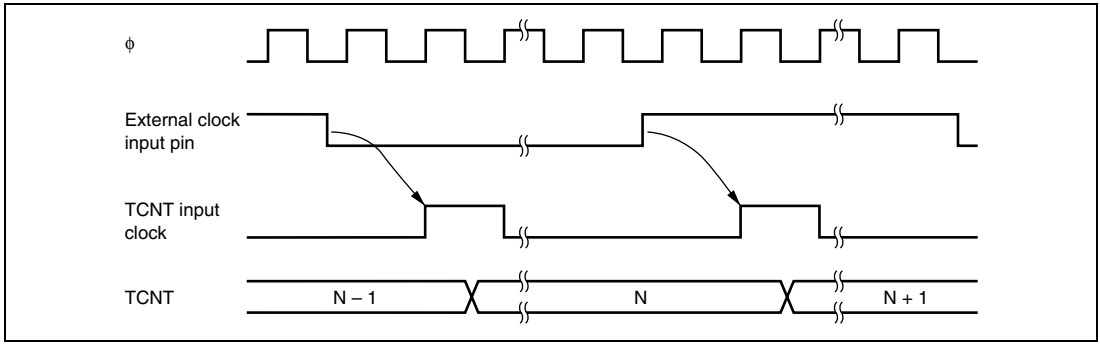


Figure 11.4 Count Timing for External Clock Input

11.5.2 Timing of CMFA and CMFB Setting When a Compare-Match Occurs

The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated when the TCOR and TCNT values match. The compare-match signal is generated at the last state in which the match is true, just before the timer counter is updated. Therefore, when TCOR and TCNT match, the compare-match signal is not generated until the next incrementation clock input. Figure 11.5 shows the timing of CMF flag setting.

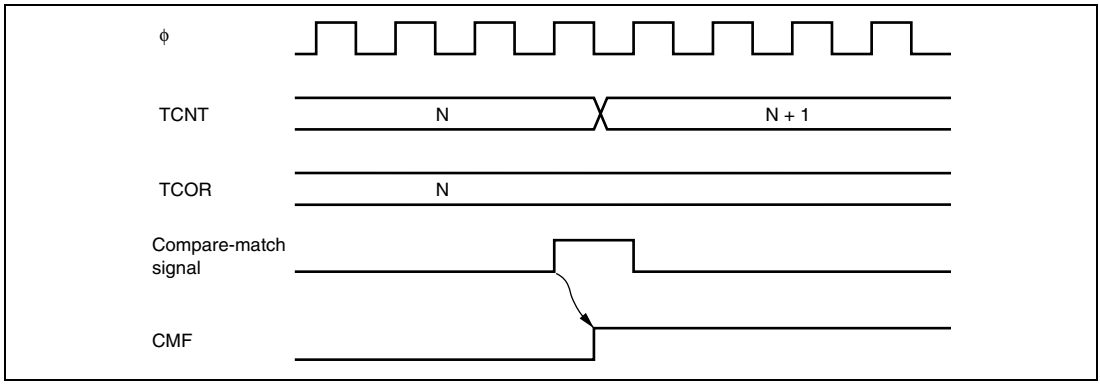


Figure 11.5 Timing of CMF Flag Setting

11.5.3 Timing of Timer Output When a Compare-Match Occurs

When a compare-match occurs, the timer output changes as specified by the output select bits, OS3 to OS0, in TCSR. Figure 11.6 shows the timing when the output is set to toggle at compare-match A.

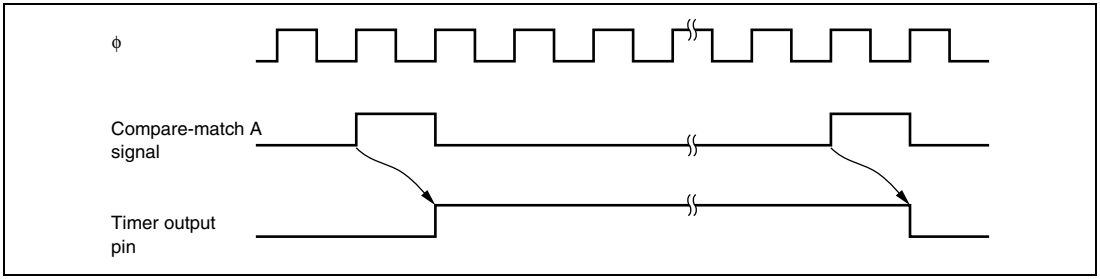


Figure 11.6 Timing of Timer Output

11.5.4 Timing of Compare-Match Clear When a Compare-Match Occurs

TCNT is cleared when compare-match A or B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 11.7 shows the timing of this operation.

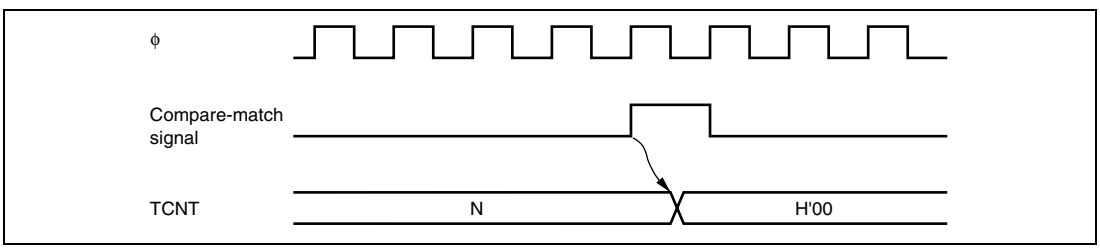


Figure 11.7 Timing of Compare-Match Clear

11.5.5 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 states. Figure 11.8 shows the timing of this operation.

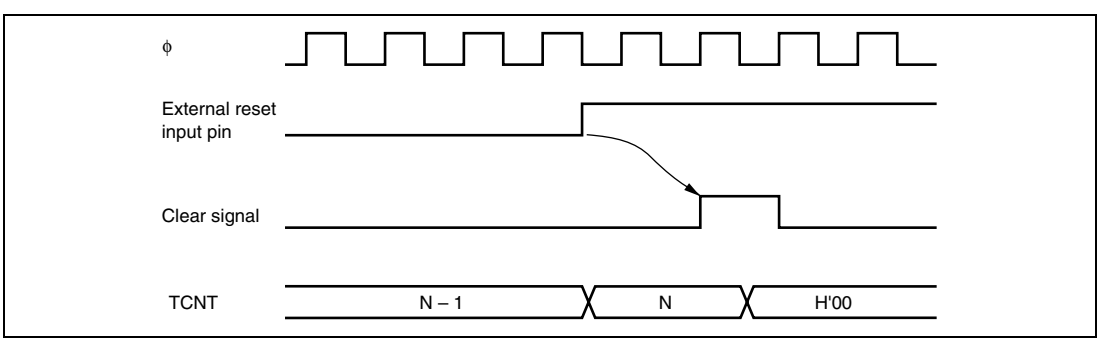


Figure 11.8 Timing of Clearing by External Reset Input

11.5.6 Timing of Overflow Flag (OVF) Setting

OVF in TCSR is set to 1 when the timer count overflows (changes from H'FF to H'00). Figure 11.9 shows the timing of this operation.

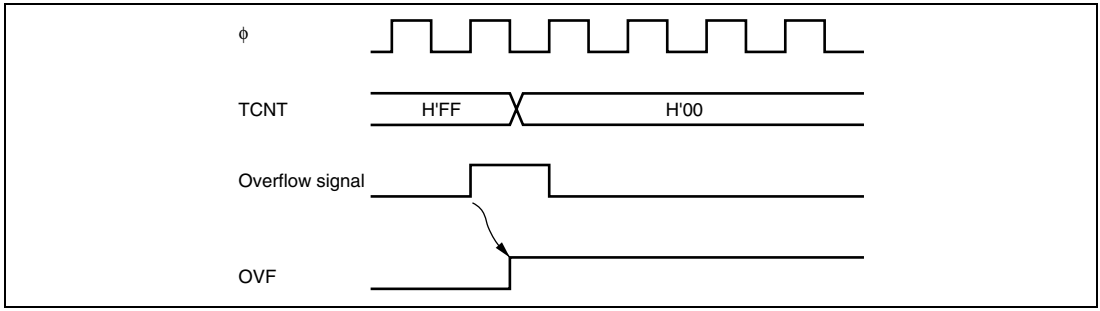


Figure 11.9 Timing of OVF Setting

11.6 Operation with Cascaded Connection

If the CKS2 to CKS0 bits in one of TCR_0 and TCR_1 (or TCR_2 and TCR_3) are set to B'100, the 8-bit timers (TMR) of the two channels are cascaded. With this configuration, a single 16-bit timer can be used (16-bit timer mode) or compare-matches of 8-bit channel 0 (channel 2) can be counted by the timer of channel 1 (channel 3) (compare-match count mode). In the case that channel 0 is connected to channel 1 in cascade, the timer operates as described below.

11.6.1 16-Bit Count Mode

When the CKS2 to CKS0 bits in TCR_0 are set to B'100, 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 of compare-match flags
  - The CMF flag in TCSR_0 is set to 1 when a 16-bit compare-match occurs.
  - The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare-match, the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare-match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is cleared even if counter clear by the TMRI01 pin has also been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cannot be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by the OS3 to OS0 bits in TCSR_0 is in accordance with the 16-bit compare-match conditions.

— Control of output from the TMO1 pin by the OS3 to OS0 bits in TCSR_1 is in accordance with the lower 8-bit compare-match conditions.

11.6.2 Compare-Match Count Mode

When the CKS2 to CKS0 bits in TCR_1 are B'100, TCNT_1 counts compare-match A for channel 0. Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each channel.

11.7 Interrupt Sources

11.7.1 Interrupt Sources and DTC Activation

The 8-bit timer can generate three types of interrupt: CMIA, CMIB, and OVI. Table 11.2 shows the interrupt sources and priority. Each interrupt source can be enabled or disabled independently by the interrupt enable bits in TCR. Independent signals are sent to the interrupt controller for each interrupt. It is also possible to activate the DTC by means of CMIA or CMIB interrupt.

Table 11.2 8-Bit Timer Interrupt Sources

Interrupt source	Description	Flag	DTC Activation*	Interrupt Priority
CMIA0	TCORA_0 compare-match	CMFA	Possible	High
CMIB0	TCORB_0 compare-match	CMFB	Possible	
OVI0	TCNT_0 overflow	OVF	Not possible	
CMIA1	TCORA_1 compare-match	CMFA	Possible	
CMIB1	TCORB_1 compare-match	CMFB	Possible	
OVI1	TCNT_1 overflow	OVF	Not possible	
CMIA2	TCORA_2 compare-match	CMFA	Possible	
CMIB2	TCORB_2 compare-match	CMFB	Possible	
OVI2	TCNT_2 overflow	OVF	Not possible	
CMIA3	TCORA_3 compare-match	CMFA	Possible	
CMIB3	TCORB_3 compare-match	CMFB	Possible	
OVI3	TCNT_3 overflow	OVF	Not possible	Low

Note: This list shows the initial state directly after the reset. Relative channel priorities can be changed by the interrupt controller.

11.7.2 A/D Converter Activation

The A/D converter can be activated only by channel 0 compare match A.

If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrence of channel 0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

11.8 Usage Notes

11.8.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed. Figure 11.10 shows this operation.

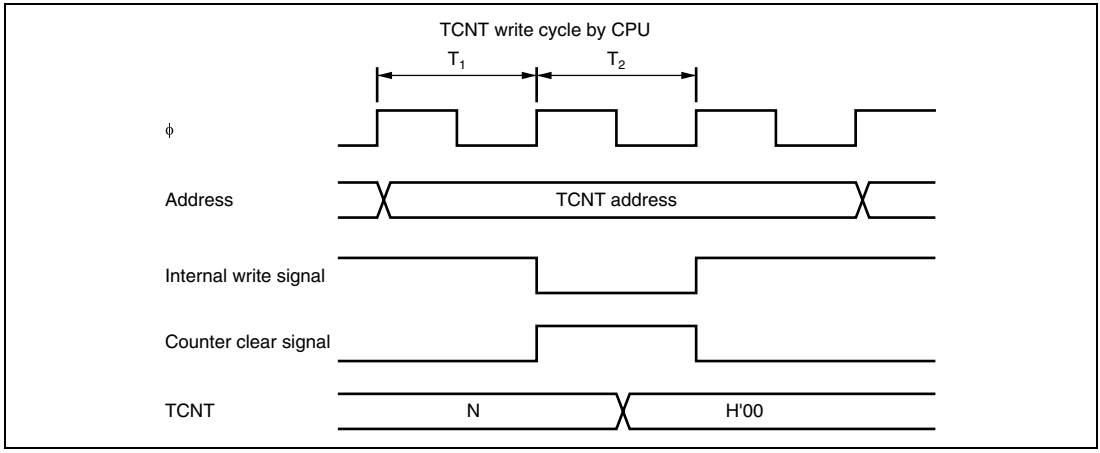


Figure 11.10 Contention between TCNT Write and Clear

11.8.2 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the counter is not incremented. Figure 11.11 shows this operation.

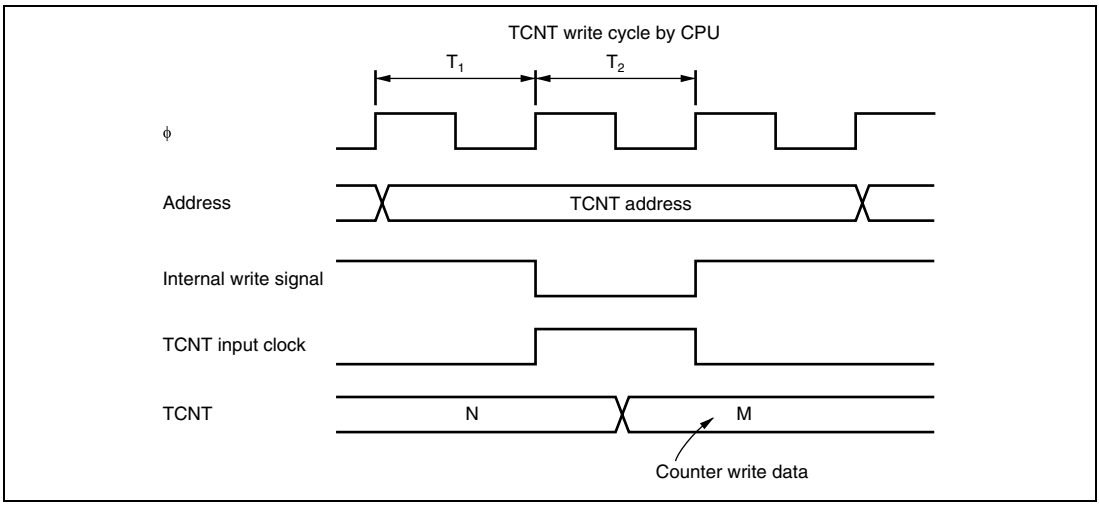


Figure 11.11 Contention between TCNT Write and Increment

11.8.3 Contention between TCOR Write and Compare-Match

During the T_2 state of a TCOR write cycle, the TCOR write has priority even if a compare-match occurs and the compare-match signal is disabled. Figure 11.12 shows this operation.

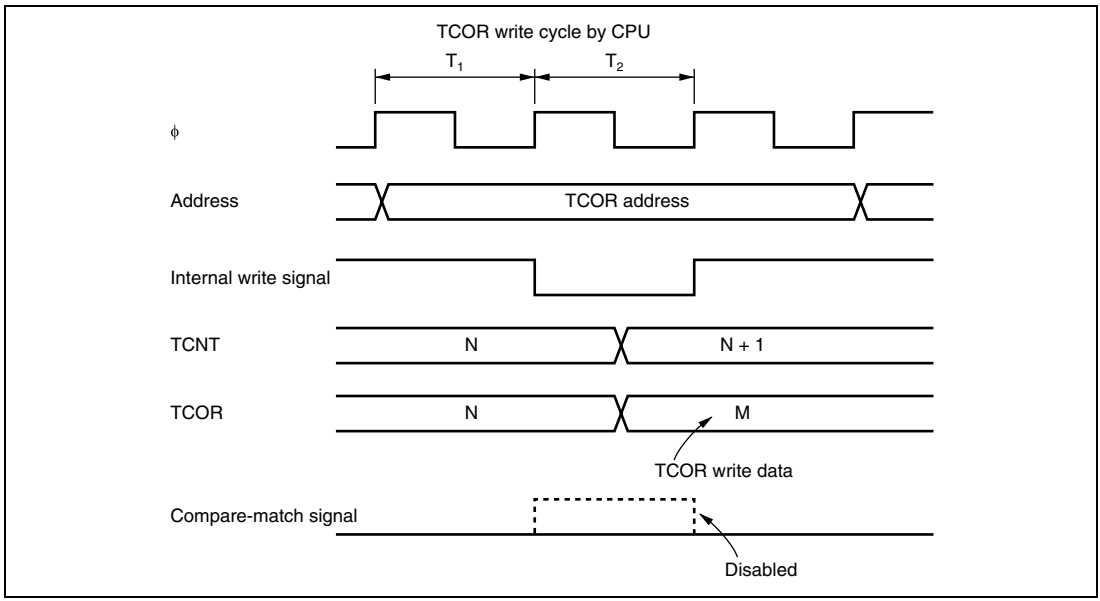


Figure 11.12 Contention between TCOR Write and Compare-Match

11.8.4 Contention between Compare-Matches A and B

If compare-matches A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output states set for compare-match A and compare-match B, as shown in table 11.3.

Table 11.3 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	↑
No change	Low

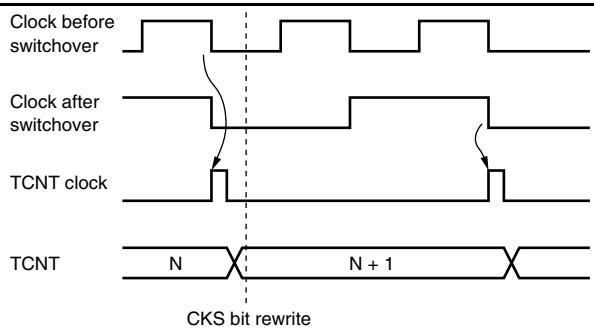
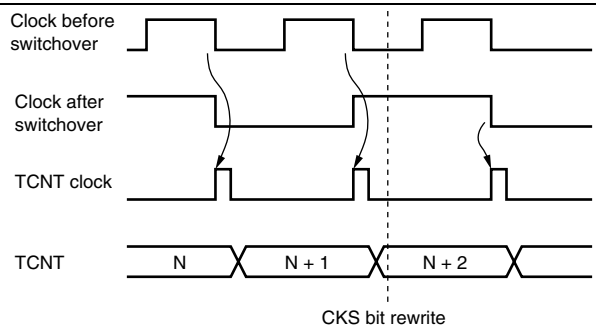
11.8.5 Switching of Internal Clocks and TCNT Operation

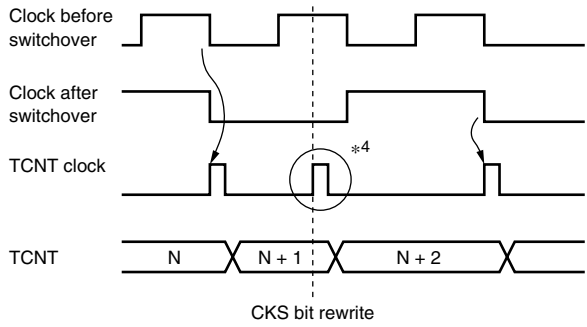
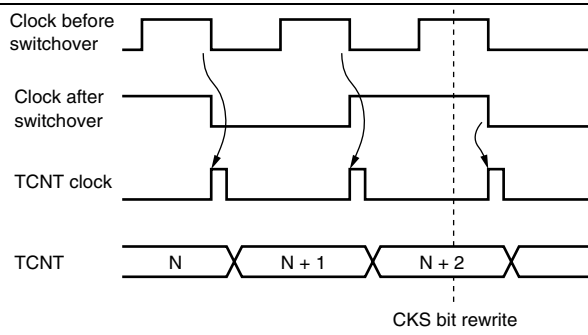
TCNT may increment erroneously when the internal clock is switched over. Table 11.4 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in no. 3 in table 11.4, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

Erroneous incrementation can also happen when switching between internal and external clocks.

Table 11.4 Switching of Internal Clock and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*[1]	
2	Switching from low to high*[2]	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
3	Switching from high to low*3	
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

11.8.6 Contention between Interrupts and Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

11.8.7 Mode Setting in Cascading

When 16-bit counter mode and compare much counter mode are set simultaneously, input clock of TCNT_0 and TCNT_1 (or TCNT_2 and TCNT_3) are not generated, causing the counter to stop operating. This mode should not be set.

Section 12 Watchdog Timer (WDT)

The watchdog timer (WDT) is an 8-bit timer in two channels that can reset this LSI internally or generate the internal NMI interrupt, if a system crash prevents the CPU from writing to the timer.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

The block diagram of the WDT is shown in figure 12.1.

12.1 Features

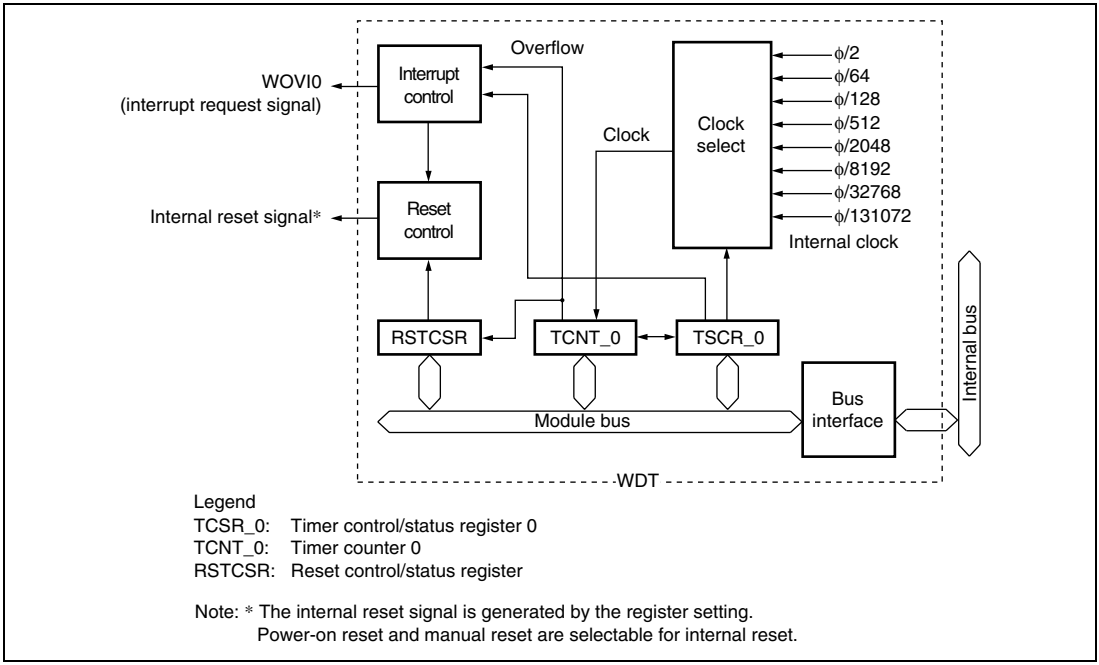
- Selectable from eight counter input clocks for WDT_0
  Selectable from 16 counter input clocks for WDT_1
- Switchable between watchdog timer mode and interval timer mode

Watchdog timer mode

- If the counter in WDT_0 overflows, it is possible to select whether this LSI is internally reset or not
- Power-on reset and manual reset are selectable for internal reset
- If the counter in WDT_1 overflows, it is possible to select whether this LSI is internally reset at a power-on timing or the internal NMI interrupt is generated

Interval timer mode

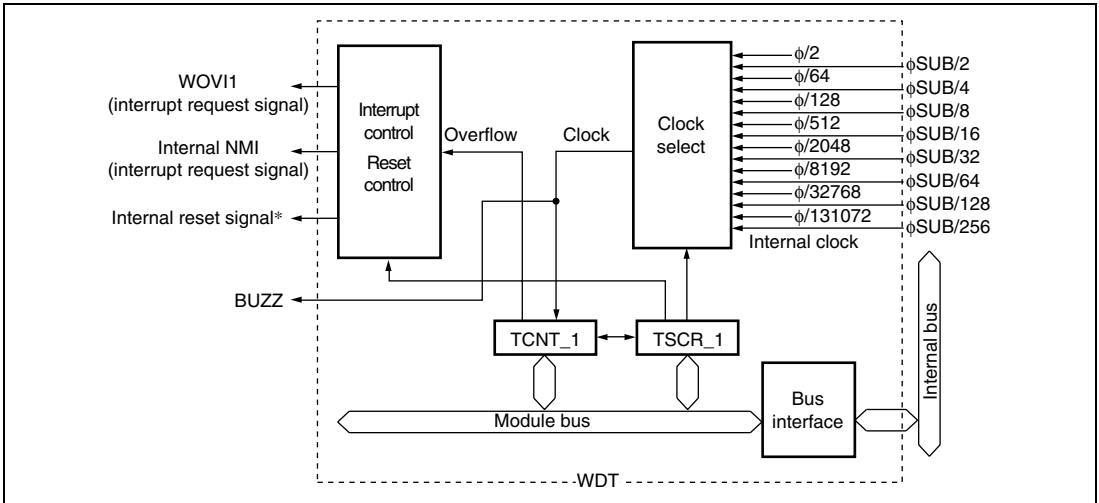
- If the counter overflows, the WDT generates an interval timer interrupt (WOVI)
- Selected clock can be output from BUZZ output pin (WDT_1)



Legend
TCSR_0: Timer control/status register 0
TCNT_0: Timer counter 0
RSTCSR: Reset control/status register

Note: * The internal reset signal is generated by the register setting.
Power-on reset and manual reset are selectable for internal reset.

Figure 12.1 Block Diagram of WDT_0 (1)



Legend
TCSR_1: Timer control/status register 1
TCNT_1: Timer counter 1

Note: * The internal reset signal is generated by the register setting.
Power-on reset is set for internal reset.

Figure 12.1 Block Diagram of WDT_1 (2)

12.2 Input/Output Pin

Table 12.1 shows the WDT pin.

Table 12.1 Pin Configuration

Name	Symbol	Input/Output	Function
Buzz output	BUZZ	Output	Clock output selected at WDT_1

12.3 Register Descriptions

The WDT has the following three registers. To prevent accidental overwriting, TCSR and TCNT have to be written to by a different method from normal registers. For details, refer to section 12.6.1, Notes on Register Access. For detailed description on the system control register, refer to section 3.2.2, System Control Register (SYSCR). For details on the pin function control register, refer to section 7.3.6, Pin Function Control Register (PFCR).

- Timer counter (TCNT)
- Timer control/status register (TCSR)
- Reset control/status register (RSTCSR)

12.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TME bit in TCSR is cleared to 0.

12.3.2 Timer Control/Status Register (TCSR)

TCSR functions include selecting the clock source to be input to TCNT and the timer mode.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*1	Overflow Flag Indicates that TCNT has overflowed. Only a 0 can be written to this bit, to clear the flag. [Setting condition] When TCNT overflows (changes from H'FF to H'00). However, when internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing condition] Cleared by reading TCSR*2 when OVF = 1, then writing 0 to OVF
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4, 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	CKS2	0	R/W	Clock Select 0 to 2 These bits select the clock source to be input to TCNT. The overflow frequency*3 for ϕ = 20 MHz is enclosed in parentheses. 000: Clock ϕ/2 (frequency: 25.6 μs) 001: Clock ϕ/64 (frequency: 819.2 μs) 010: Clock ϕ/128 (frequency: 1.6 ms) 011: Clock ϕ/512 (frequency: 6.6 ms) 100: Clock ϕ/2048 (frequency: 26.2 ms) 101: Clock ϕ/8192 (frequency: 104.9 ms) 110: Clock ϕ/32768 (frequency: 419.4 ms) 111: Clock ϕ/131072 (frequency: 1.68 s)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Notes: 1. Only 0 can be written, for flag clearing.
2. When the OVF flag is polled with the interval timer interrupt disabled, read the OVF bit while it is 1 at least twice.
3. The overflow period is the time from when TCNT starts counting up from H′00 until an overflow occurs.

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*[1]	Overflow Flag Indicates that TCNT has overflowed. Only a 0 can be written to this bit, to clear the flag. [Setting condition] When TCNT overflows (changes from H'FF to H'00). However, when internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing condition] Cleared by reading TCSR*[2] when OVF = 1, then writing 0 to OVF
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4	PSS	0	R/W	Prescaler Select Selects the clock source input to TCNT of WDT_1. Controls the operation in power-down mode transition. 0: TCNT counts divided clock of ϕ-base prescaler (PSM). When SLEEP instruction is executed in high-speed mode or medium-speed mode, transition to sleep mode or software standby mode is made. 1: TCNT counts divided clock of ϕ_{SUB} -base prescaler (PSS). When SLEEP instruction is executed in high-speed mode or medium-speed mode, transition to sleep mode, software standby mode, or watch mode* is made. Note: * When transition is made to watch mode, make sure that high-speed mode is set.
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI (REST/$\overline{NMI}$) 0: NMI interrupt is requested. 1: Power-on reset is requested.

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock source to be input to TCNT. The overflow frequency*[3] for ϕ = 20 MHz and ϕ_{SUB} = 32.768 MHz is enclosed in parentheses.
0	CKS0	0	R/W	
				When PSS = 0:
				000: Clock ϕ/2 (frequency: 25.6 µs)
				001: Clock ϕ/64 (frequency: 819.2 µs)
				010: Clock ϕ/128 (frequency: 1.6 ms)
				011: Clock ϕ/512 (frequency: 6.6 ms)
				100: Clock ϕ/2048 (frequency: 26.2 ms)
				101: Clock ϕ/8192 (frequency: 104.9 ms)
				110: Clock ϕ/32768 (frequency: 419.4 ms)
				111: Clock ϕ/131072 (frequency: 1.68 s)
				When PSS = 1:
				000: Clock ϕ_{SUB}/2 (frequency: 15.6 ms)
				001: Clock ϕ_{SUB}/4 (frequency: 31.3 ms)
				010: Clock ϕ_{SUB}/8 (frequency: 62.5 ms)
				011: Clock ϕ_{SUB}/16 (frequency: 125 ms)
				100: Clock ϕ_{SUB}/32 (frequency: 250 ms)
				101: Clock ϕ_{SUB}/64 (frequency: 500 ms)
				110: Clock ϕ_{SUB}/128 (frequency: 1 s)
				111: Clock ϕ_{SUB}/256 (frequency: 2 s)

Notes: 1. Only 0 can be written, for flag clearing.
2. When the OVF flag is polled with the interval timer interrupt disabled, read the OVF bit while it is 1 at least twice.
3. The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs.

12.3.3 Reset Control/Status Register (RSTCSR) (WDT_0 only)

RSTCSR controls to generate the internal reset signal when TCNT overflows, and selects the type of internal reset signal. RSTCSR is initialized by a reset signal from the $\overline{\text{RES}}$ pin, and not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Overflow Flag This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode, and only 0 can be written, to clear the flag. [Setting condition] Set when TCNT overflows (changed from H'FF to H'00) in watchdog timer mode [Clearing condition] Cleared by reading RSTCSR when WOVF = 1, and then writing 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation. 0: Reset signal is not generated even if TCNT overflows. (Though this LSI is not reset, TCNT and TCSR in WDT are reset.) 1: Reset signal is generated if TCNT overflows.
5	RSTS	0	R/W	Reset Select Selects the type of internal reset, which is generated if TCNT overflows during watchdog timer operation. 0: Power-on reset 1: Manual reset
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

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

12.4 Operation

12.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ bit in TCSR and the TME bit to 1.

Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs. Thus, TCNT does not overflow while the system is operating normally.

When the WDT is used as a watchdog timer and the RSTE bit in RSTCSR of WDT_0 is set to 1, and if TCNT overflows without being rewritten because of a system malfunction or other error, an internal reset signal for this LSI is output for 518 system clocks.

When the RST/$\overline{NMI}$ bit in TCSR of WDT_1 is set to 1, and if TCNT overflows, the internal reset signal is output for 516 system clocks. When the RST/ $\overline{NMI}$ bit is cleared to 0, if TCNT overflows, an NMI interrupt request is generated (for 515 or 516 system clocks when the clock source is set to ϕ_{SUB} (PSS = 1)).

An internal reset request from the watchdog timer and a reset input from the $\overline{RES}$ pin are both treated as having the same vector. If a WDT internal reset request and the $\overline{RES}$ pin reset occur at the same time, the $\overline{RES}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

An NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin are both treated as having the same vector. So, avoid handling an NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin at the same time.

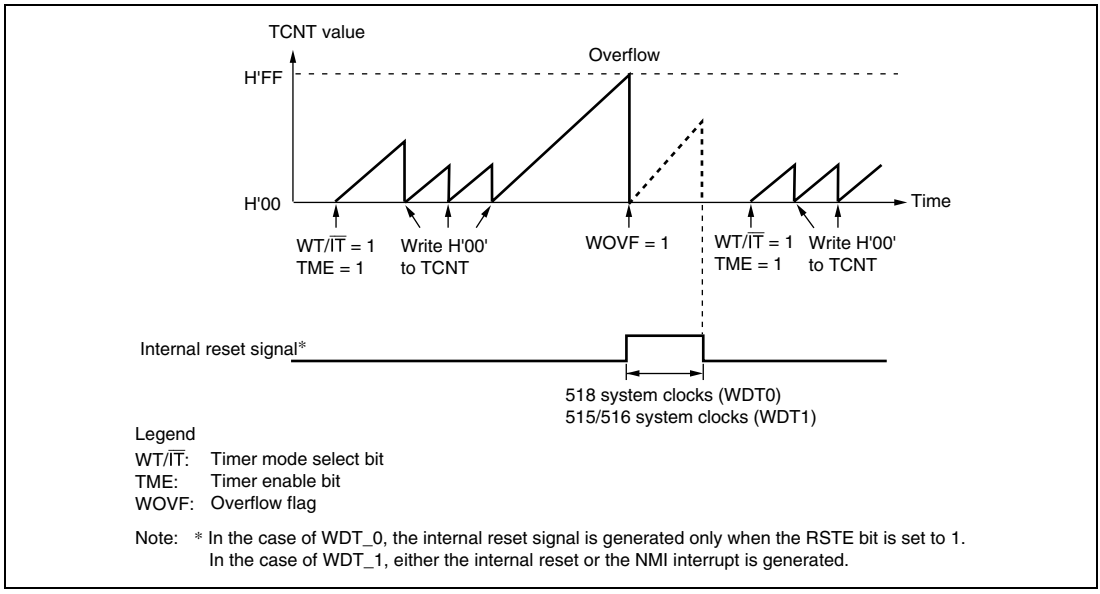


Figure 12.2 Watchdog Timer Mode Operation

12.4 2 Interval Timer Mode

To use the WDT as an internal timer, set the WT/$\overline{IT}$ bit in TCSR to 0 and the TME bit to 1.

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows. Therefore, an interrupt can be generated at specified times.

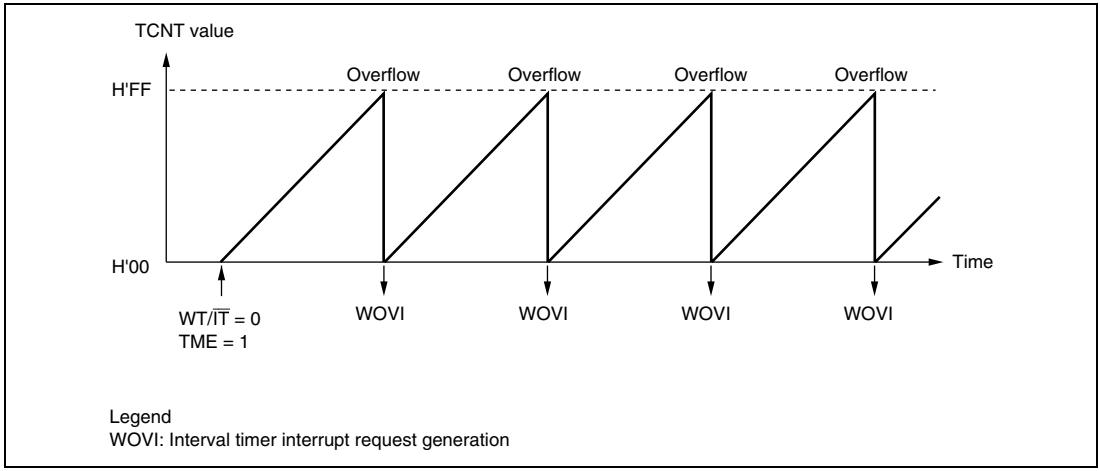


Figure 12.3 Interval Timer Mode Operation

12.4.3 Timing of Setting Overflow Flag (OVF)

The OVF bit in TCSR is set to 1 if TCNT overflows during interval timer operation. At the same time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure 12.4.

In the case of WDT_1, when an NMI request is selected in watchdog timer mode, if TCNT overflows, the OVF bit in TCSR is set to 1 and an NMI interrupt is requested simultaneously.

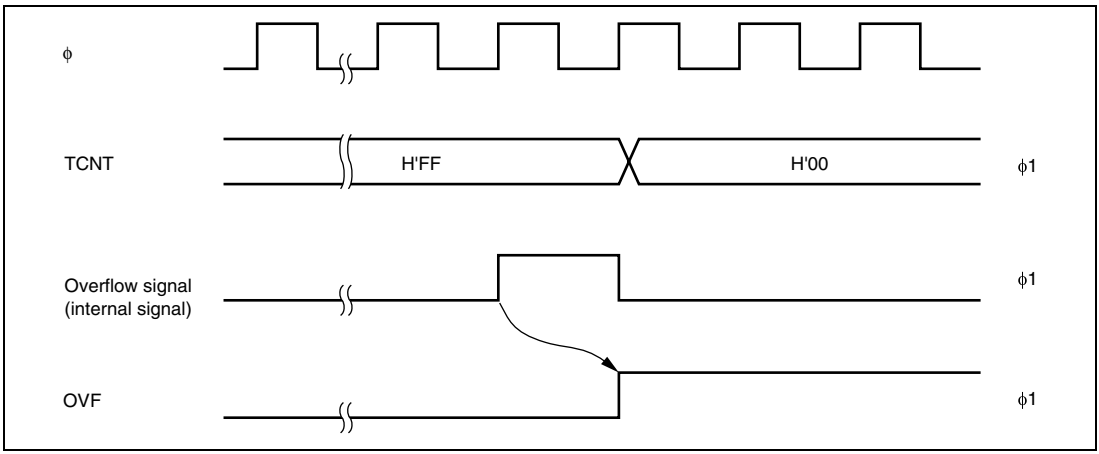


Figure 12.4 Timing of OVF Setting

12.4.4 Timing of Setting Watchdog Timer Overflow Flag (WOVF)

In the case of WDT_0, if TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. If TCNT overflows while the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated for the entire chip. This timing is shown in figure 12.5.

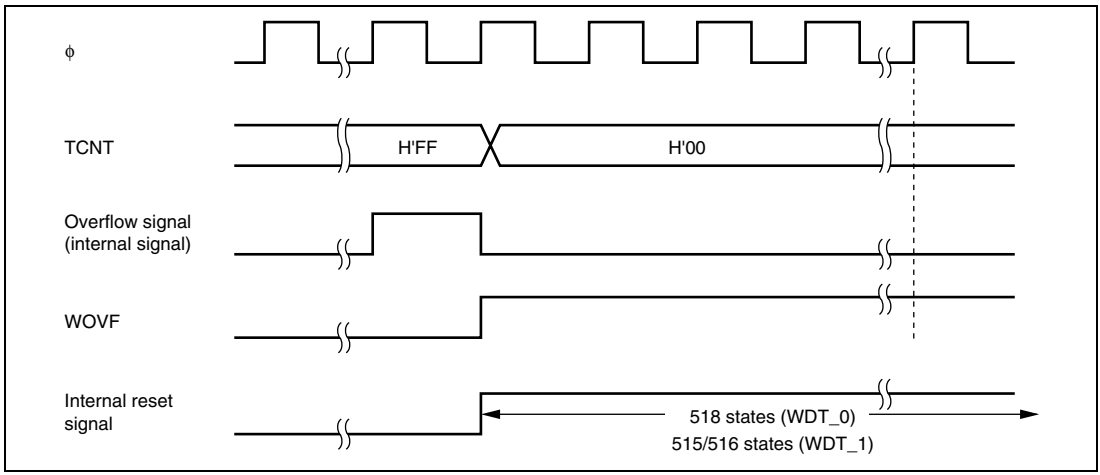


Figure 12.5 Timing of WOVF Setting

12.5 Interrupt Sources

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag in TCSR is set to 1. OVF must be cleared to 0 in the interrupt handling routine.

If an NMI request has been chosen in watchdog timer mode, an NMI request is generated when a TCNT overflow occurs.

Table 12.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag
WOVI	TCNT overflow (interval timer mode)	OVF
NMI	TCNT overflow (watchdog timer mode)	OVF

12.6 Usage Notes

12.6.1 Notes on Register Access

The watchdog timer's TCNT and TCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT, TCSR, and RSTCSR: These registers must be written to by a word transfer instruction. They cannot be written to by a byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, the relative condition shown in figure 12.6 needs to be satisfied in order to write to TCNT or TCSR. When writing to TCNT or TCSR, transfer the write date to the lower byte of TCNT or TCSR. The upper byte of TCNT or TCSR must be H'5A.

To write to RSTCSR, execute a word transfer instruction for address H'FF76. A byte transfer instruction cannot write to RSTCSR.

The method of writing 0 to the WOVF bit differs from that of writing to the RSTE and RSTS bits. To write 0 to the WOVF bit, satisfy the condition shown in figure 12.6. If satisfied, the transfer instruction clears the WOVF bit to 0, but has no effect on the RSTE and RSTS bits. To write to the RSTE and RSTS bits, satisfy the condition shown in figure 12.6. If satisfied, the transfer instruction writes the values in bits 5 and 6 of the lower byte into the RSTE and RSTS bits, but has no effect on the WOVF bit.

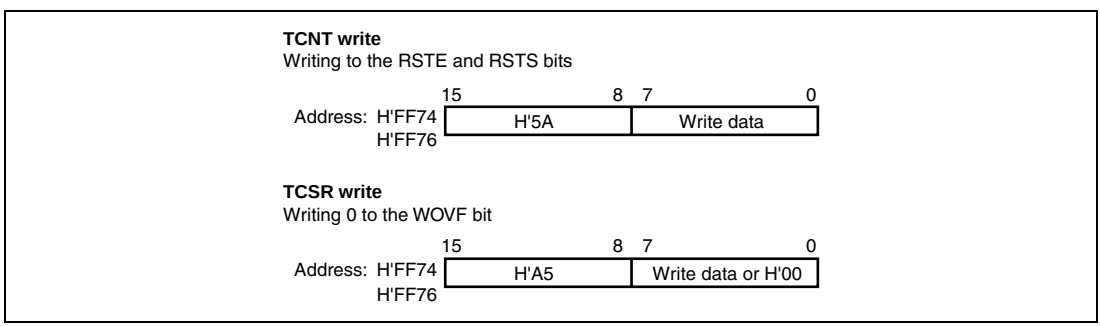


Figure 12.6 Writing to TCNT and TCSR (in the case of WDT_0)

Reading from TCNT, TCSR and RSTCSR (in the case of WDT_0): These registers are read in the same way as other registers. The read addresses are allocated in H'FF74 for TCSR, H'FF75 for TCNT, and H'FF77 for RSTCSR.

12.6.2 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 12.7 shows this operation.

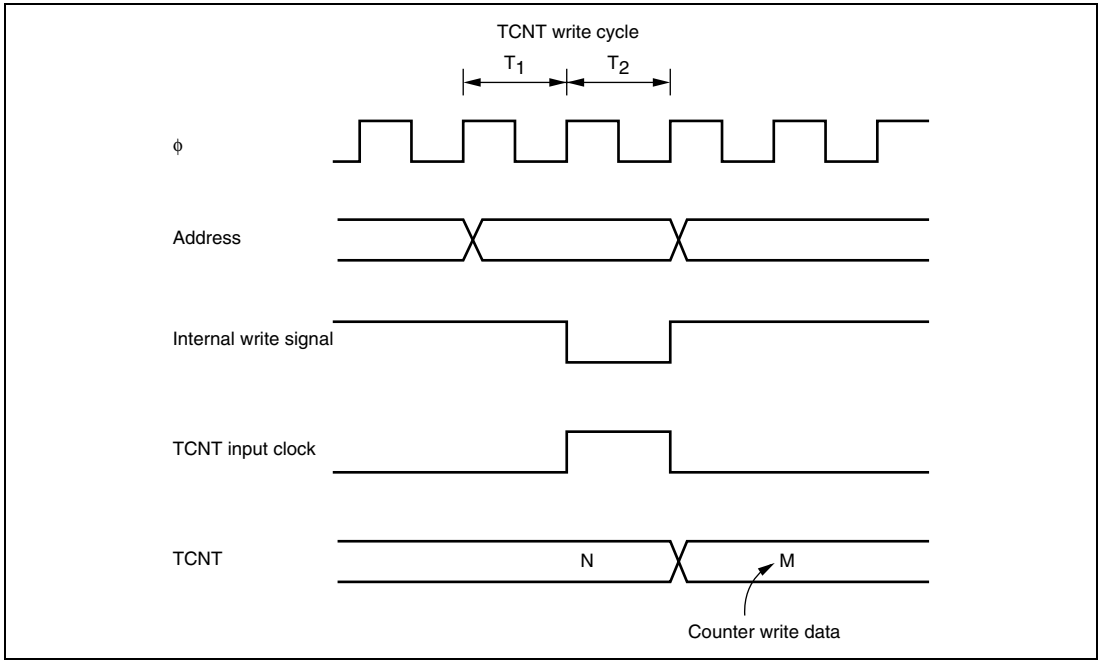


Figure 12.7 Contention between TCNT Write and Increment

12.6.3 Changing Value of CKS2 to CKS0

If the CKS0 to CKS2 bits in TCSR are modified while the WDT is operating, errors could occur in the incrementation. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before changing the value of the CKS0 to CKS2 bits.

12.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched between watchdog timer mode and interval timer mode while the WDT is operating, errors could occur. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before switching the timer mode.

12.6.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally when TCNT overflows, if the RSTE bit is cleared to 0 in watchdog timer mode, however TCNT_0 and TCSR_0 of the WDT_0 are reset.

TCNT, TCSR, or RSTCR cannot be written to for 132 states following an overflow. During this period, any attempt to read the WOVF flag is not acknowledged. Accordingly, wait 132 states after an overflow to write 0 to the WOVF flag.

12.6.6 OVF Flag Clearing in Interval Timer Mode

When the OVF flag setting conflicts with the OVF flag reading in interval timer mode, writing 0 to the OVF flag may not clear the flag even though the OVF flag has been read while it is 1. If there is a possibility that the OVF flag setting and reading will conflict, such as when the OVF flag is polled with the interval timer interrupt disabled, read the OVF flag while it is 1 at least twice before writing 0 to the OVF flag to clear the flag.

Section 13 Serial Communication Interface (SCI)

This LSI has five independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. In asynchronous mode, serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). In asynchronous mode, a function is also provided for serial communication between multiple processors (multiprocessor communication function). The SCI also supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as an extension function in clocked synchronous serial communication mode.

13.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode

  (Because the same pin is used as the clock input/output pin for channel 1 and channel 4, these clocks cannot be output at the same time.)
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously.

  The double-buffering configuration is adopted in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  External clock can be selected as a transfer clock source (except in Smart Card interface mode)
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests.

  The transmit-data-empty and receive-data-full interrupts can be used to activate the data transfer controller (DTC).
- Module stop mode can be set

Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in the case of a framing error
- Communications between multiple processors are possible.

Clocked synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

Smart Card interface

- Automatic transmission of an error signal when a parity error is detected in receive mode
- Automatic data retransmission when an error signal is received in transmit mode
- Direct convention and inverse convention both supported

Figure 13.1 shows a block diagram of the SCI.

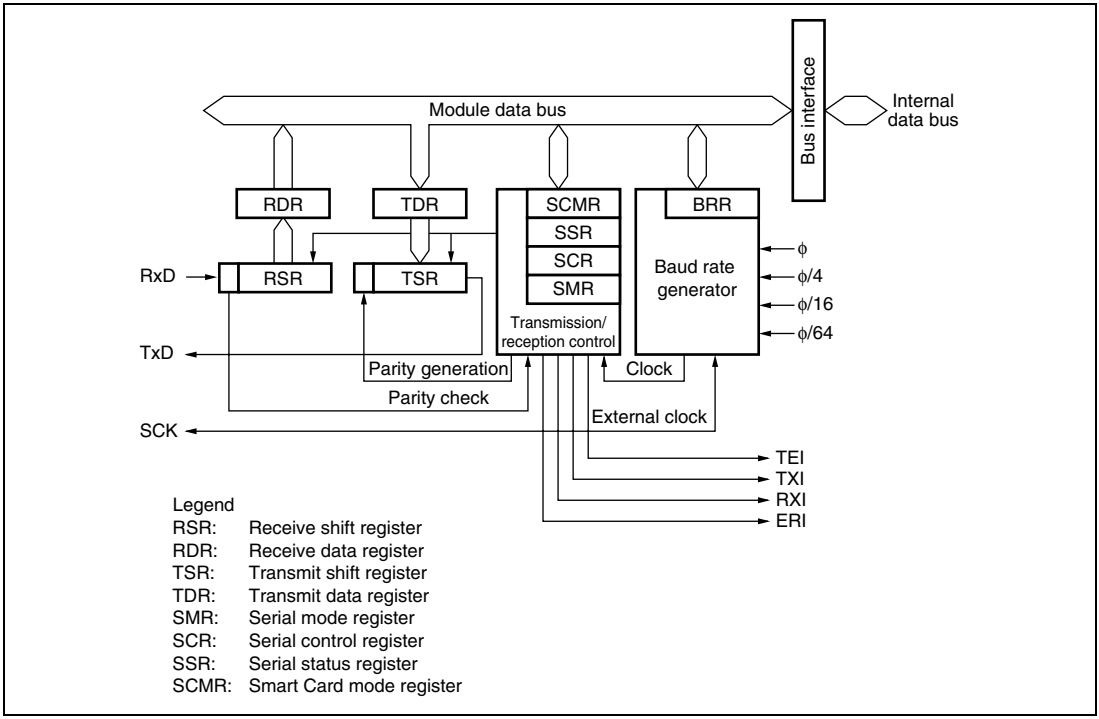


Figure 13.1 Block Diagram of SCI

13.2 Input/Output Pins

Table 13.1 shows the pin configuration for each SCI channel.

Table 13.1 Pin Configuration

Channel	Pin Name*1	I/O	Function
0	SCK0	I/O	Clock input/output in channel 0
	RxD0	Input	Receive data input in channel 0
	TxD0	Output	Transmit data output in channel 0
1	SCK1*2	I/O	Clock input/output in channel 1
	RxD1	Input	Receive data input in channel 1
	TxD1	Output	Transmit data output in channel 1
2	SCK2	I/O	Clock input/output in channel 2
	RxD2	Input	Receive data input in channel 2
	TxD2	Output	Transmit data output in channel 2
3	SCK3	I/O	Clock input/output in channel 3
	RxD3	Input	Receive data input in channel 3
	TxD3	Output	Transmit data output in channel 3
4	SCK4*2	I/O	Clock input/output in channel 4
	RxD4	Input	Receive data input in channel 4
	TxD4	Output	Transmit data output in channel 4

Notes: 1. Pin names SCK, RxD, and TxD are used in this manual for all channels, omitting the channel designation.
2. Because SCK1 and SCK4 are allocated to the same pin, these clocks cannot be output at the same time.

13.3 Register Descriptions

The SCI has the following registers for each channel. The serial mode register (SMR), serial status register (SSR), and serial control register (SCR) are described separately for normal serial communication interface mode and Smart Card interface mode, because some of their bit functions differ depending on the mode.

Channel 0

- Receive shift register_0 (RSR_0)
- Receive data register_0 (RDR_0)
- Transmit data register_0 (TDR_0)
- Transmit shift register_0 (TSR_0)
- Serial mode register_0 (SMR_0)
- Serial control register_0 (SCR_0)
- Serial status register_0 (SSR_0)
- Smart Card mode register_0 (SCMR_0)
- Bit rate register_0 (BRR_0)

Channel 1

- Receive shift register_1 (RSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_1)
- Transmit shift register_1 (TSR_1)
- Serial mode register_1 (SMR_1)
- Serial control register_1 (SCR_1)
- Serial status register_1 (SSR_1)
- Smart Card mode register_1 (SCMR_1)
- Bit rate register_1 (BRR_1)

Channel 2

- Receive shift register_2 (RSR_2)
- Receive data register_2 (RDR_2)
- Transmit data register_2 (TDR_2)
- Transmit shift register_2 (TSR_2)
- Serial mode register_2 (SMR_2)
- Serial control register_2 (SCR_2)
- Serial status register_2 (SSR_2)
- Smart Card mode register_2 (SCMR_2)
- Bit rate register_2 (BRR_2)

Channel 3

- Receive shift register_3 (RSR_3)
- Receive data register_3 (RDR_3)
- Transmit data register_3 (TDR_3)
- Transmit shift register_3 (TSR_3)
- Serial mode register_3 (SMR_3)
- Serial control register_3 (SCR_3)
- Serial status register_3 (SSR_3)
- Smart Card mode register_3 (SCMR_3)
- Bit rate register_3 (BRR_3)

Channel 4

- Receive shift register_4 (RSR_4)
- Receive data register_4 (RDR_4)
- Transmit data register_4 (TDR_4)
- Transmit shift register_4 (TSR_4)
- Serial mode register_4 (SMR_4)
- Serial control register_4 (SCR_4)
- Serial status register_4 (SSR_4)
- Smart Card mode register_4 (SCMR_4)
- Bit rate register_4 (BRR_4)

13.3.1 Receive Shift Register (RSR)

RSR is a shift register that is used to receive serial data input to the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically.

RSR cannot be directly accessed by the CPU.

13.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once.

RDR cannot be written to by the CPU.

RDR is initialized to H'00 by a reset, or in standby mode, watch mode, or module stop mode.

13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. As TDR and TSR function as a double buffer in this way, continuous transmit operations are possible. When the SCI transmits one byte of serial data, if the next transmit data has already been written to TDR, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1.

TDR is initialized to H'FF by a reset, or in standby mode, watch mode, or module stop mode.

13.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin. TSR cannot be directly accessed by the CPU.

13.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator clock source.

Some bit functions of SMR differ between normal serial communication interface mode and Smart Card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB (bit 7) of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.

Bit	Bit Name	Initial Value	R/W	Description
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even. In reception, a check is performed to see if the total number of 1 bits in the receive character plus parity bit is even. 1: Selects odd parity. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd. In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode. For details, see section 13.5, Multiprocessor Communication Function.
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relationship between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in bit rate register .

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode When this bit is set to 1, the SCI operates in GSM mode. In GSM mode, the timing of the TEND setting is advanced by 11.0 etu (Elementary Time Unit: the time for transfer of one bit), and clock output control mode addition is performed. For details, refer to section 13.7.8, Clock Output Control.
6	BLK	0	R/W	When this bit is set to 1, the SCI operates in block transfer mode. For details on block transfer mode, refer to section 13.7.3, Block Transfer Mode.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data in transmission, and the parity bit is checked in reception. In Smart Card interface mode, this bit must be set to 1.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity. For details on setting this bit in Smart Card interface mode, refer to section 13.7.2, Data Format (Except for Block Transfer Mode).
3	BCP1	0	R/W	Basic Clock Pulse 1 and 0
2	BCP0	0	R/W	These bits specify the number of basic clock periods in a 1-bit transfer interval on the Smart Card interface. 00: 32 clock (S = 32) 01: 64 clock (S = 64) 10: 372 clock (S = 372) 11: 256 clock (S = 256) For details, refer to section 13.7.4, Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode. S stands for the value of S in bit rate register.

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relationship between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR.

Note: etu (Elementary Time Unit): Abbreviation for the transfer period for one bit.

13.3.6 Serial Control Register (SCR)

SCR is a register that enables or disables SCI transfer operations and interrupt requests, and is also used to selection of the transfer clock source. For details on interrupt requests, refer to section 13.8, Interrupt Sources. Some bit functions of SCR differ between normal serial communication interface mode and Smart Card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt request is enabled. TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0, or clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled. RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF, FER, PER, or ORER flag in SSR, then clearing the flag to 0, or clearing the RIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0. SMR setting must be performed to decide the transfer format before setting the TE bit to 1. When this bit is cleared to 0, the transmission operation is disabled, and the TDRE flag is fixed at 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the reception format before setting the RE bit to 1. Clearing the RE bit to 0 does not affect the RDRF, FER,PER, and ORER flags, which retain their states.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 13.5, Multiprocessor Communication Function. When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RERF, FER, and ORER flags in SSR, are not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting are enabled.
2	TEIE	0	R/W	Transmit End Interrupt Enable This bit is set to 1, TEI interrupt request is enabled. TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode 00: On-chip baud rate generator SCK pin functions as I/O port 01: On-chip baud rate generator Outputs a clock of the same frequency as the bit rate from the SCK pin. 1X: External clock Inputs a clock with a frequency 16 times the bit rate from the SCK pin. Clocked synchronous mode 0X: Internal clock (SCK pin functions as clock output) 1X: External clock (SCK pin functions as clock input)

Legend

X: Don't care

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled. TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0, or clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled. RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF, FER, PER, or ORER flag in SSR, then clearing the flag to 0, or clearing the RIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0. SMR setting must be performed to decide the transfer format before setting the TE bit to 1. When this bit is cleared to 0, the transmission operation is disabled, and the TDRE flag is fixed at 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the reception format before setting the RE bit to 1. Clearing the RE bit to 0 does not affect the RDRF, FER,PER, and ORER flags, which retain their states.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) Write 0 to this bit in Smart Card interface mode. When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RERF, FER, and ORER flags in SSR, are not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting are enabled.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in Smart Card interface mode. TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Enable or disable clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, refer to section 13.7.8, Clock Output Control. When the GM bit in SMR is 0: 00: Output disabled (SCK pin can be used as an I/O port pin) 01: Clock output 1X: Reserved When the GM bit in SMR is 1: 00: Output fixed low 01: Clock output 10: Output fixed high 11: Clock output

Legend
X: Don't care

13.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared. Some bit functions of SSR differ between normal serial communication interface mode and Smart Card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and transfers data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. [Setting condition] • When the next serial reception is completed while RDRF = 1 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued either. [Clearing condition] • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
4	FER	0	R/(W)*	Framing Error Indicates that a framing error occurred during reception in asynchronous mode, causing abnormal termination. [Setting condition] • When the stop bit is 0 In 2 stop bit mode, only the first stop bit is checked for a value to 1; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to FER after reading FER = 1 In 2-stop-bit mode, only the first stop bit is checked. The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
3	PER	0	R/(W)*	Parity Error Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination. [Setting condition.] • When a parity error is detected during reception If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading PER = 1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End Indicates that transmission has been ended. [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and transfers transmit data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit data.

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

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and transfers data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. [Setting condition] • When the next serial reception is completed while RDRF = 1 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
4	ERS	0	R/(W)*	Error Signal Status Indicates that the status of an error, signal 1 returned from the reception side at reception [Setting condition] • When the low level of the error signal is sampled [Clearing condition] • When 0 is written to ERS after reading ERS = 1 The ERS flag is not affected and retains its previous state when the TE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*	Parity Error Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination. [Setting condition] • When a parity error is detected during reception If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading PER = 1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
2	TEND	1	R	Transmit End This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When the TE bit in SCR is 0 and the ERS bit is also 0 • When the ESR bit is 0 and the TDRE bit is 1 after the specified interval following transmission of 1-byte data. The timing of bit setting differs according to the register setting as follows: When GM = 0 and BLK = 0, 2.5 etu after transmission starts When GM = 0 and BLK = 1, 1.0 etu after transmission starts When GM = 1 and BLK = 0, 1.5 etu after transmission starts When GM = 1 and BLK = 1, 1.0 etu after transmission starts [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and transfers transmit data to TDR

Bit	Bit Name	Initial Value	R/W	Description
1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

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

13.3.8 Smart Card Mode Register (SCMR)

SCMR is a register that selects Smart Card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1, and cannot be modified.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer data format is 8 bits. For 7-bit data, LSB-first is fixed.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR
1	—	1	—	Reserved This bit is always read as 1, and cannot be modified.
0	SMIF	0	R/W	Smart Card Interface Mode Select This bit is set to 1 to make the SCI operate in Smart Card interface mode. 0: Normal asynchronous mode or clocked synchronous mode 1: Smart Card interface mode

13.3.9 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 13.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode, clocked synchronous mode, and Smart Card interface mode. The initial value of BRR is H'FF, and it can be read or written to by the CPU at all times.

Table 13.2 Relationships between N Setting in BRR and Bit Rate B

Communication Mode	Bit Rate	Error
Asynchronous Mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	$Error (\%) = \{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \} \times 100$
Clocked Synchronous Mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	—
Smart Card Interface Mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N + 1)}$	$Error (\%) = \{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} - 1 \} \times 100$

Legend

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

ϕ: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following tables.

SMR Setting		Clock	
CKS1	CKS0	Source	n
0	0	ϕ	0
0	1	ϕ/4	1
1	0	ϕ/16	2
1	1	ϕ/64	3

SMR Setting		
BCP1	BCP0	S
0	0	32
0	1	64
1	0	372
1	1	256

Table 13.3 shows sample N settings in BRR in normal asynchronous mode. Table 13.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 13.6 shows sample N settings in BRR in clocked synchronous mode. Table 13.8 shows sample N settings in BRR in Smart Card interface mode. In Smart Card interface mode, S (the number of basic clock periods in a 1-bit transfer interval) can be selected. For details, refer to section 13.7.4, Receive Data Sampling Timing and Reception Margin. Tables 13.5 and 13.7 show the maximum bit rates with external clock input.

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency ϕ (MHz)														
	8			9.8304			10			12			12.288		
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	Operating Frequency ϕ (MHz)											
	14			14.7456			16			17.2032		
Bit Rate (bps)	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	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	—	11	0.00	0	12	0.16	0	13	0.00

	Operating Frequency ϕ (MHz)											
	18			19.6608			20			25		
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	79	–0.12	3	86	0.31	3	88	–0.25	3	110	–0.02
150	2	233	0.16	2	255	0.00	2	64	0.16	2	80	–0.47
300	2	116	0.16	2	127	0.00	2	129	0.16	2	162	0.15
600	1	233	0.16	1	255	0.00	1	64	0.16	1	80	–0.47
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	162	0.15
2400	0	233	0.16	0	255	0.00	0	64	0.16	0	80	–0.47
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162	0.15
9600	0	58	–0.69	0	63	0.00	0	64	0.16	0	80	–0.47
19200	0	28	1.02	0	31	0.00	0	32	–1.36	0	40	–0.76
31250	0	17	0.00	0	19	–1.70	0	19	0.00	0	24	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73	0	19	1.73

Table 13.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0

Table 13.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500
25	6.2500	390625

Table 13.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bps)	Operating Frequency ϕ (MHz)									
	8		10		16		20		25	
	n	N	n	N	n	N	n	N	n	N
110										
250	3	124	—	—	3	249				
500	2	249	—	—	3	124	—	—		
1k	2	124	—	—	2	249	—	—	3	97
2.5k	1	199	1	249	2	99	2	124	2	155
5k	1	99	1	124	1	199	1	249	2	77
10k	0	199	0	249	1	99	1	124	1	155
25k	0	79	0	99	0	159	0	199	0	249
50k	0	39	0	49	0	79	0	99	0	124
100k	0	19	0	24	0	39	0	49	0	62
250k	0	7	0	9	0	15	0	19	0	24
500k	0	3	0	4	0	7	0	9	—	—
1M	0	1			0	3	0	4	—	—
2.5M			0	0*			0	1	—	—
5M							0	0*	—	—

Legend

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

Table 13.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
25	4.1667	4166666.7

Table 13.8 Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode) (When n = 0 and S = 372)

Bit Rate (bps)	Operating Frequency ϕ (MHz)							
	10.00		10.7136		13.00		14.2848	
	N	Error (%)	N	Error (%)	N	Error (%)	N	Error (%)
9600	1	30	1	25	1	8.99	1	0.00

Bit Rate (bps)	Operating Frequency ϕ (MHz)							
	16.00		18.00		20.00		25.00	
	N	Error (%)	N	Error (%)	N	Error (%)	N	Error (%)
9600	1	12.01	2	15.99	2	6.66	3	12.49

Table 13.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

ϕ (MHz)	Maximum Bit Rate (bps)	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

13.4 Operation in Asynchronous Mode

Figure 13.5 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer. In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

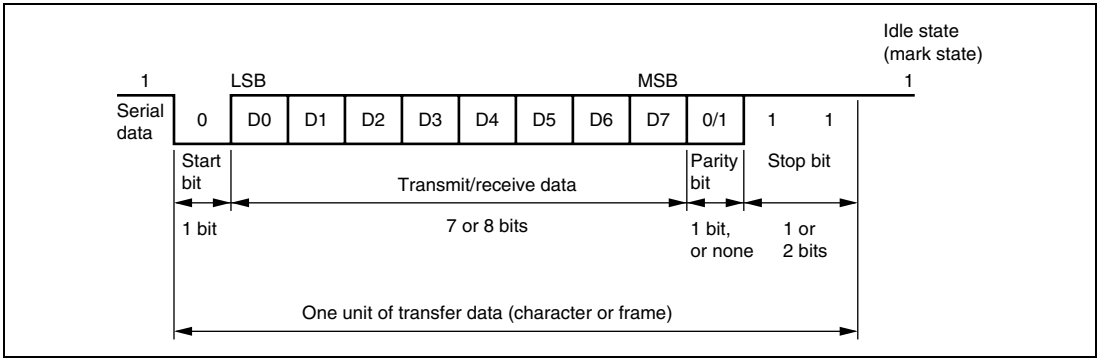


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

13.4.1 Data Transfer Format

Table 13.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 13.5, Multiprocessor Communication Function.

Table 13.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	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

13.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the transfer rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 13.6. Thus, the reception margin in asynchronous mode is given by formula (1) below.

$$M = \{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5) F\} \times 100 \ [\%]$$

... Formula (1)

Where M : Reception margin (%)
N : Ratio of bit rate to clock (N = 16)
D : Clock duty (D = 0.5 to 1.0)
L : Frame length (L = 9 to 12)
F : Absolute value of clock rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0, D (clock duty) = 0.5, and N (ratio of bit rate to clock) = 16 in formula (1), the reception margin can be given by the formula.

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

However, this is only the computed value, and a margin of 20% to 30% should be allowed for in system design.

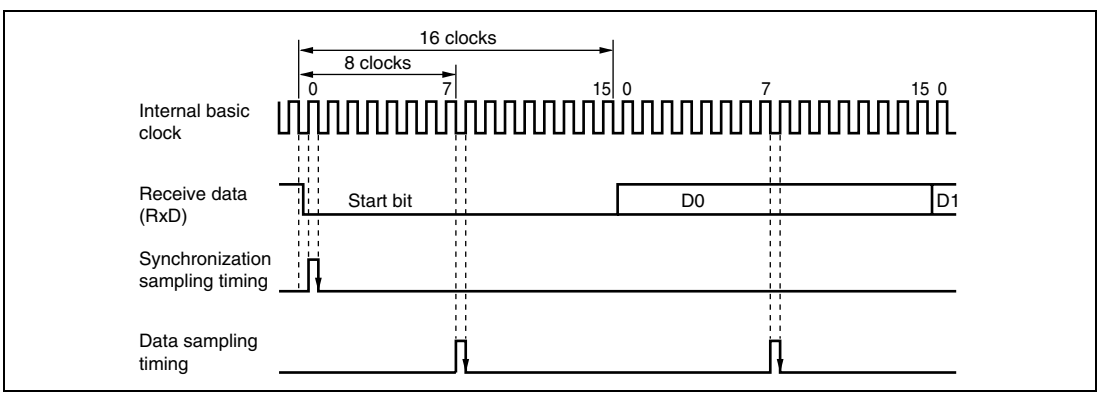


Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode

13.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE0 and CKE1 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin when setting CKE1 = 0 and CKE0 = 1. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 13.4.

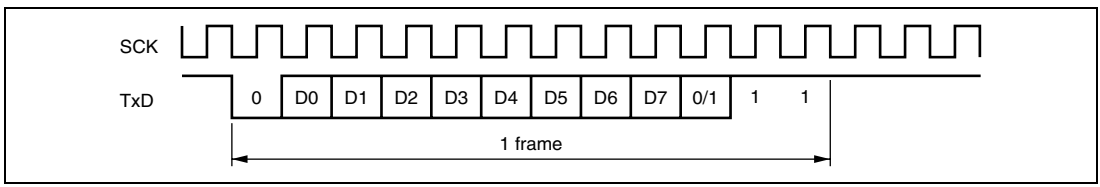


Figure 13.4 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)

13.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in figure 13.5. When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.

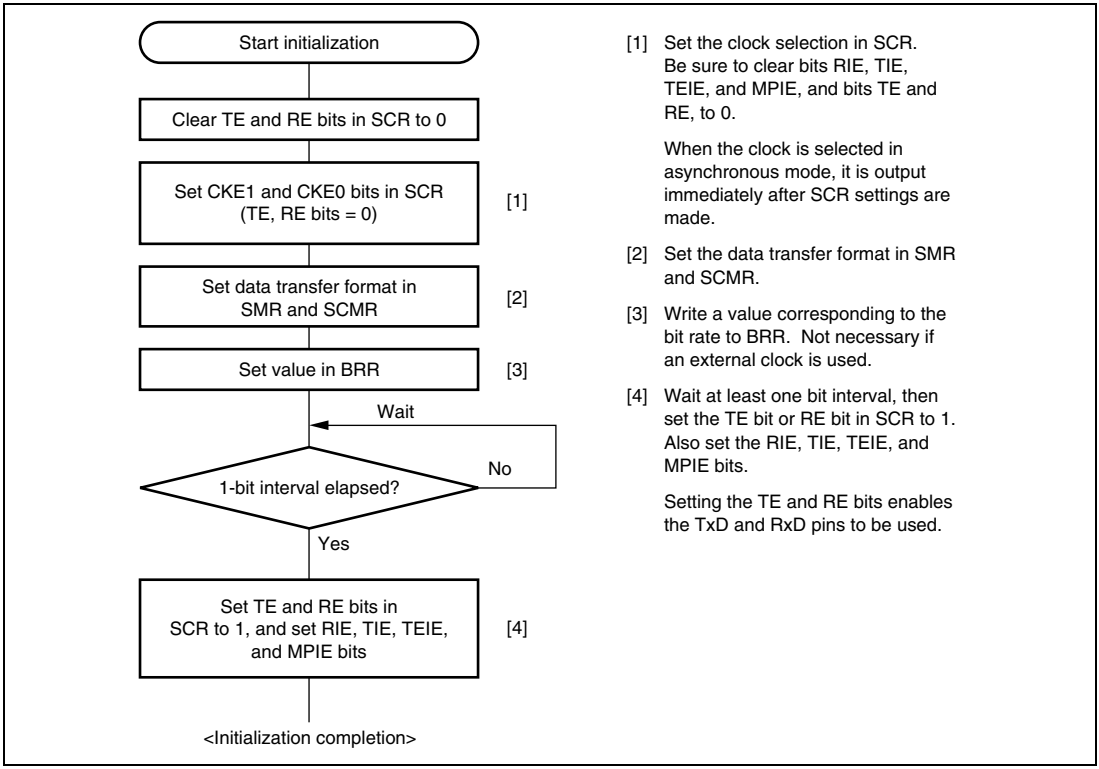


Figure 13.5 Sample SCI Initialization Flowchart

13.4.5 Serial Data Transmission (Asynchronous Mode)

Figure 13.6 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Continuous transmission is possible because the TXI interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

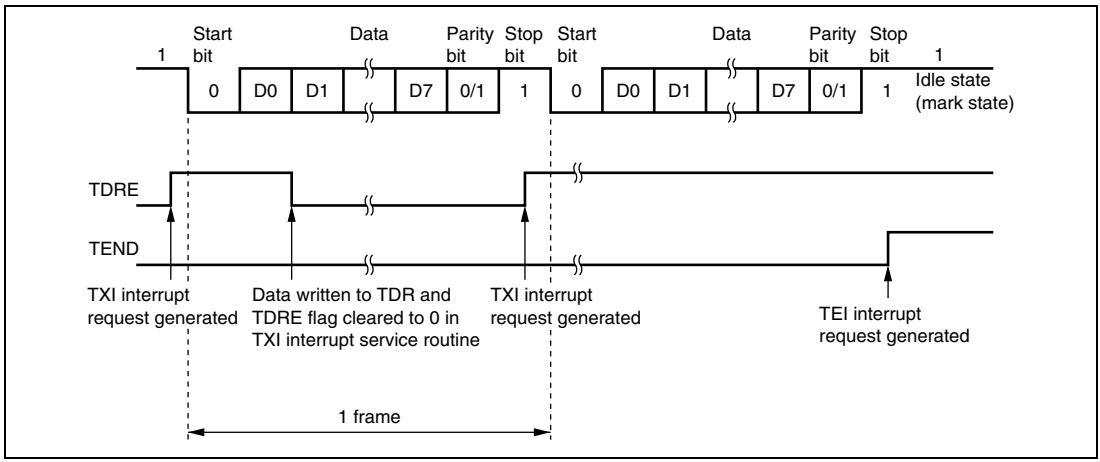


Figure 13.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Figure 13.7 shows a sample flowchart for data transmission.

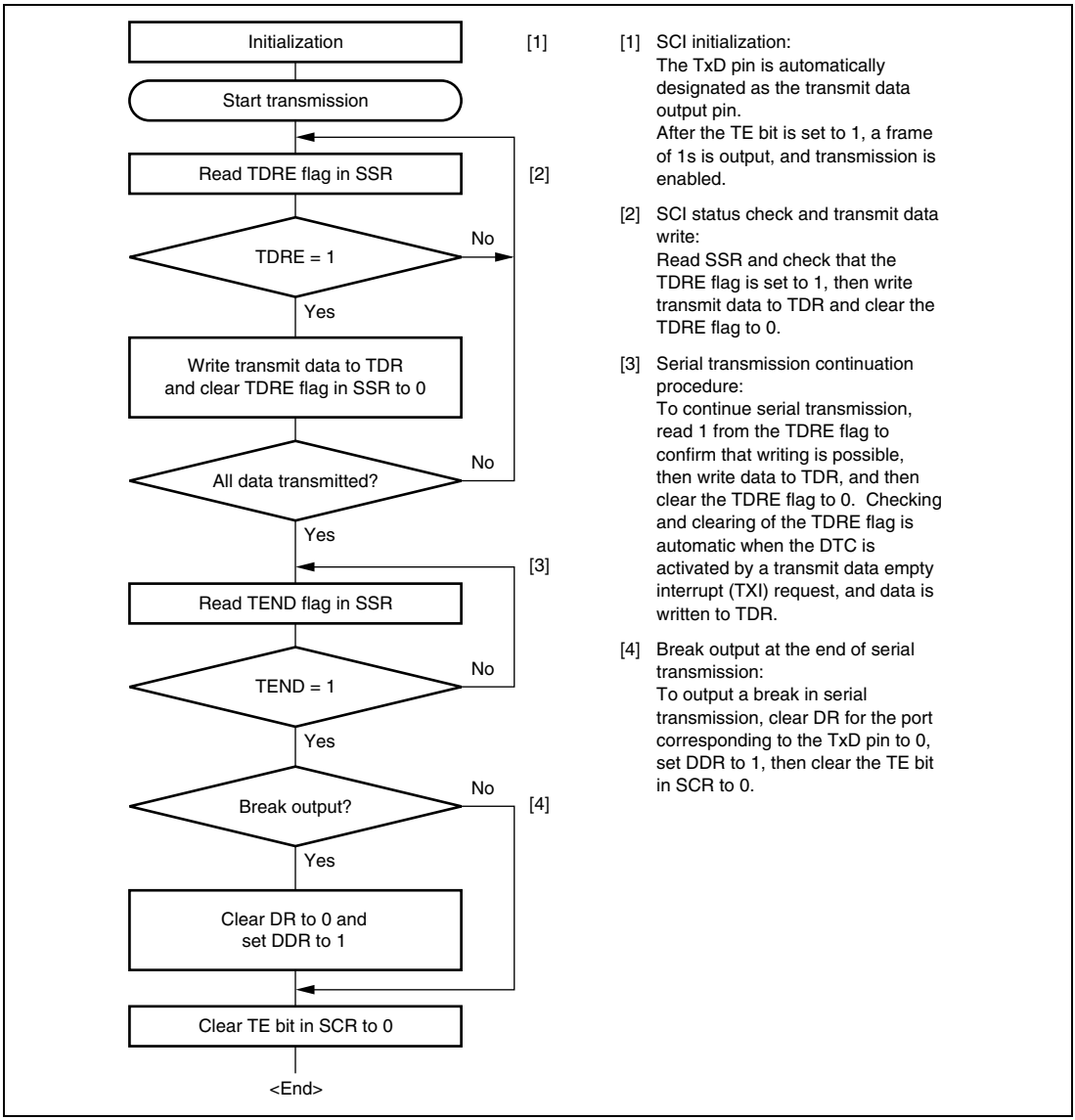


Figure 13.7 Sample Serial Transmission Flowchart

13.4.6 Serial Data Reception (Asynchronous Mode)

Figure 13.8 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line. If a start bit is detected, the SCI performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

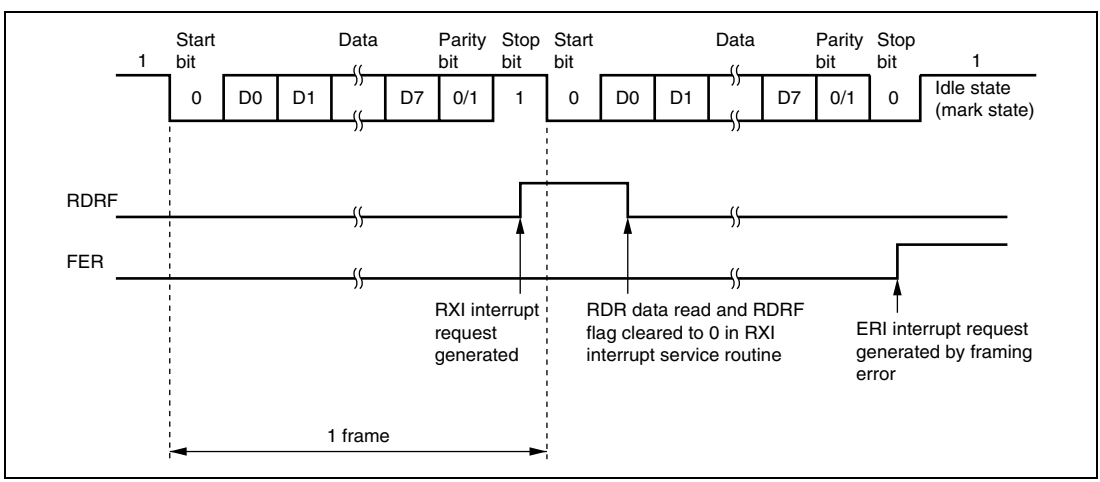


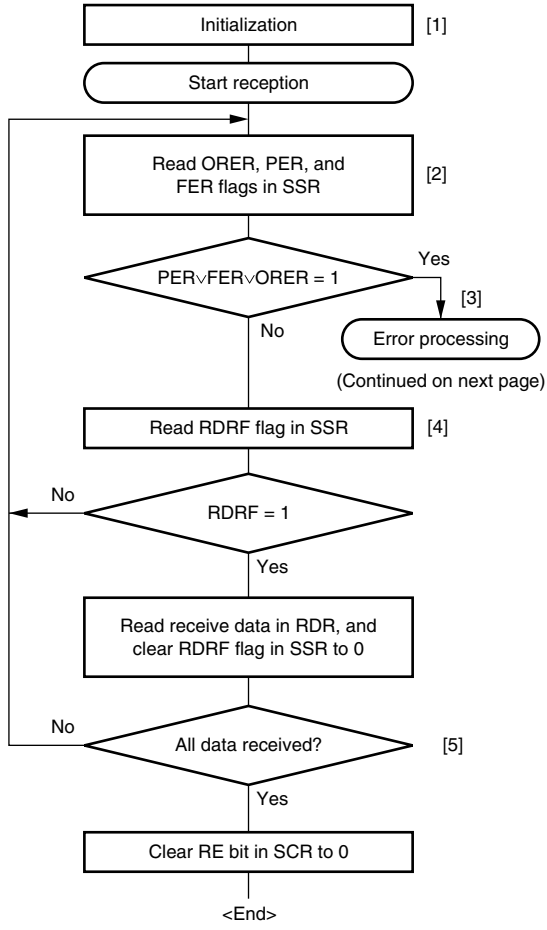
Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 13.11 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.9 shows a sample flowchart for serial data reception.

Table 13.11 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**ORER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains the state it had before data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read:
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when DTC is activated by an RXI interrupt and the RDR value is read.

Figure 13.9 Sample Serial Reception Data Flowchart (1)

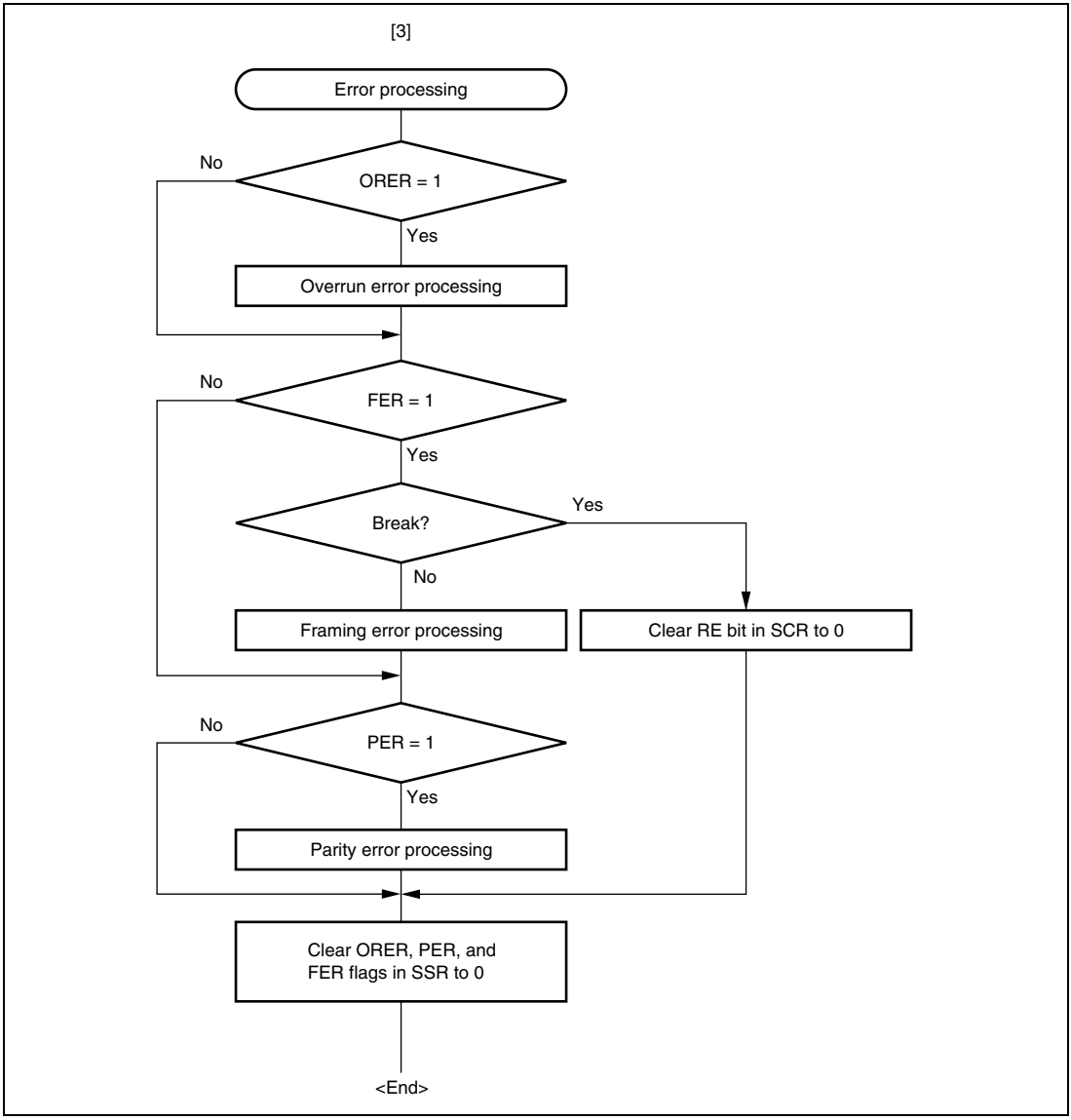


Figure 13.9 Sample Serial Reception Data Flowchart (2)

13.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 13.13 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and ORER to 1, are inhibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPB bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is rendered invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

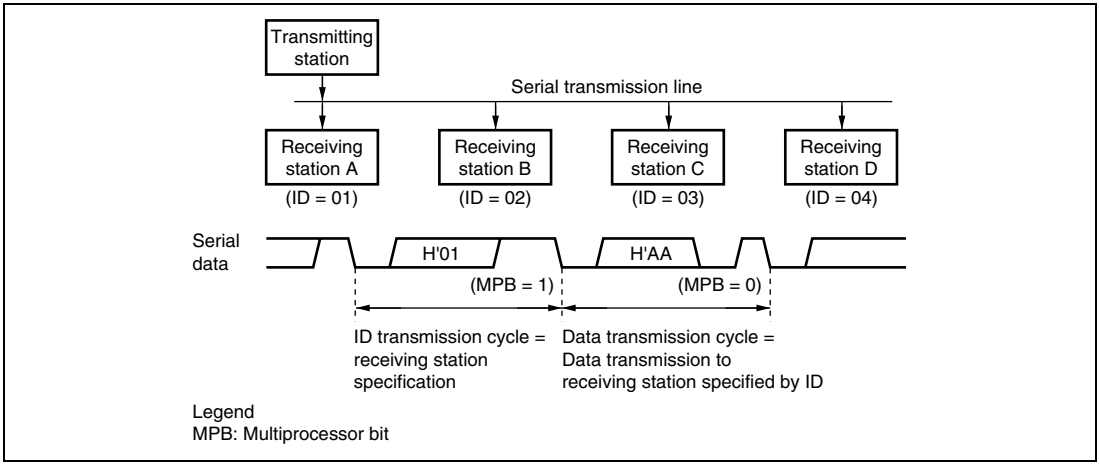
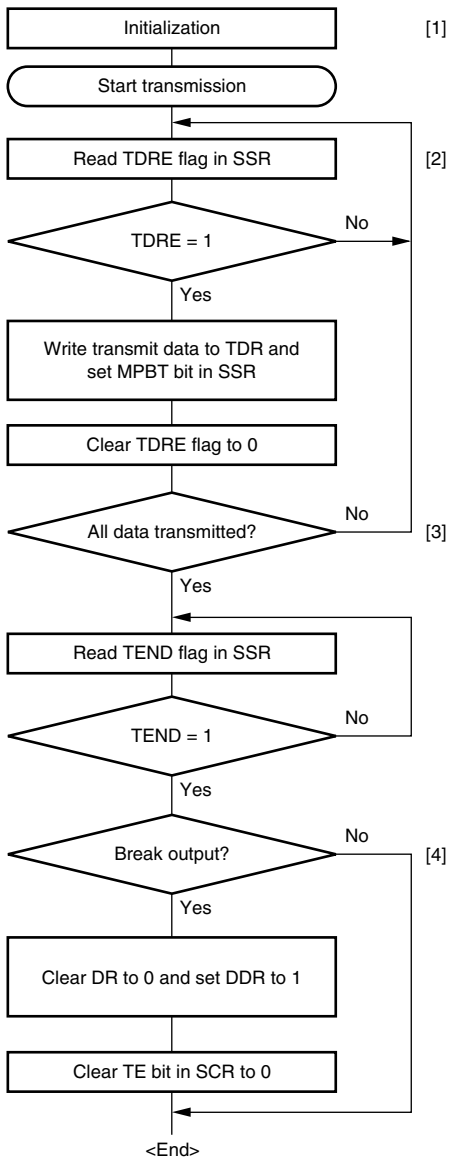


Figure 13.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

13.5.1 Multiprocessor Serial Data Transmission

Figure 13.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1.
Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DR to 0, set DDR to 1, then clear the TE bit in SCR to 0.

Figure 13.11 Sample Multiprocessor Serial Transmission Flowchart

13.5.2 Multiprocessor Serial Data Reception

Figure 13.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 13.12 shows an example of SCI operation for multiprocessor format reception.

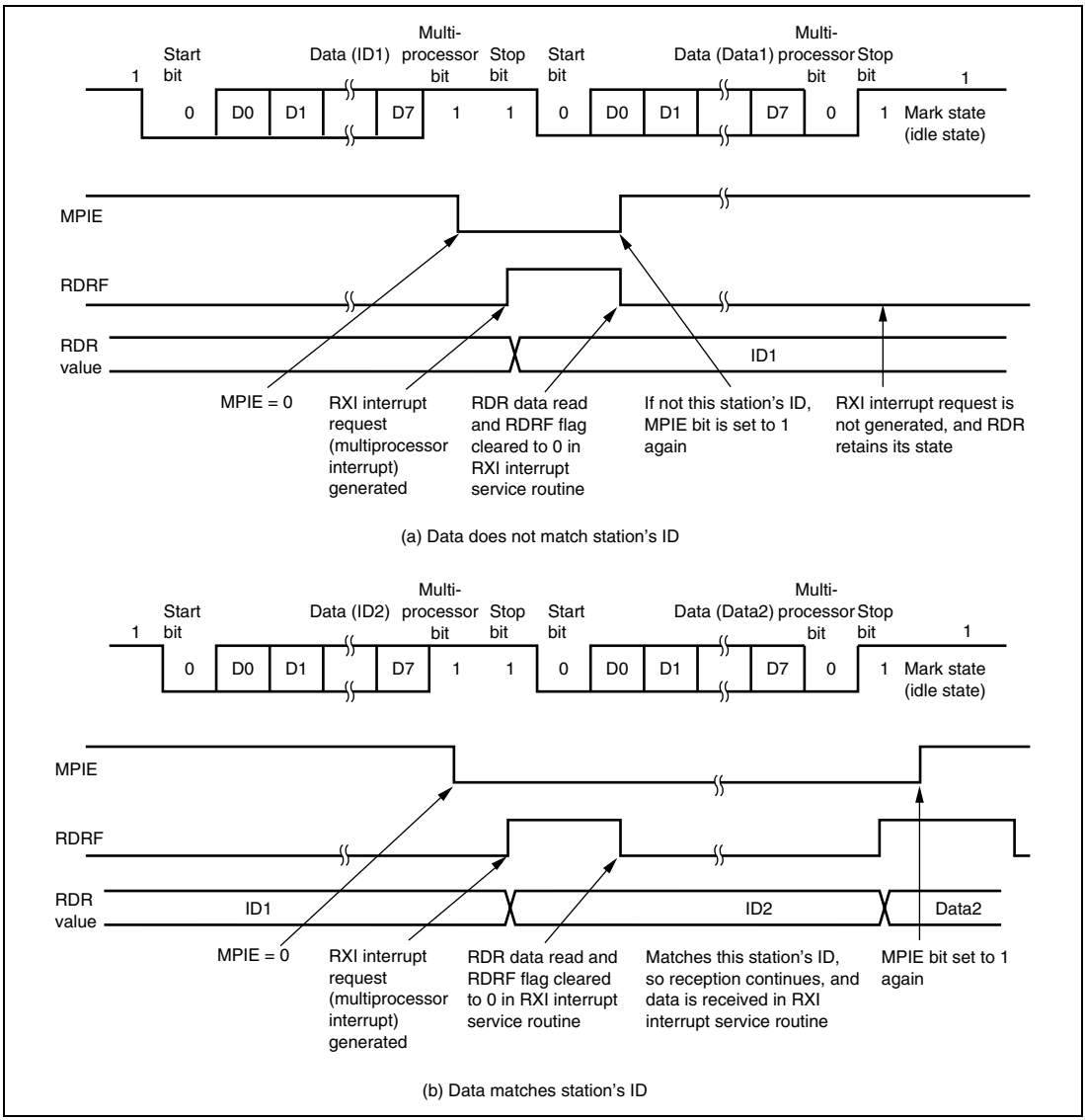


Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

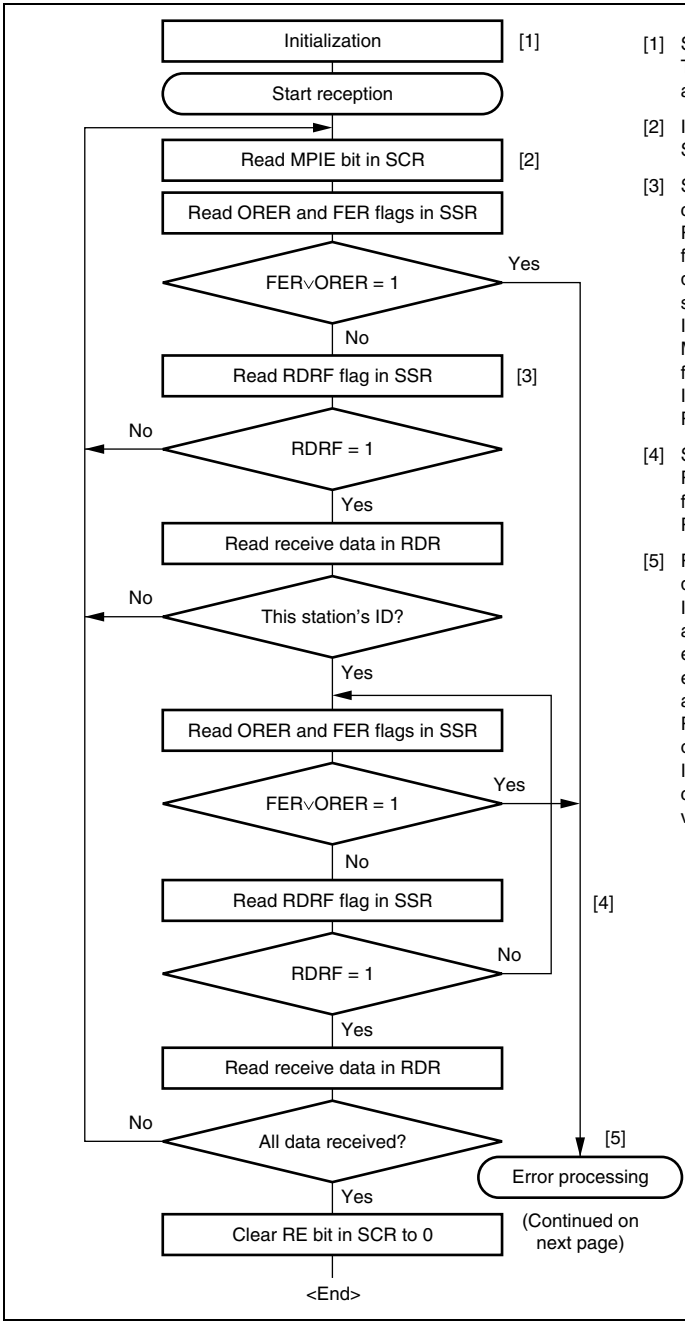


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

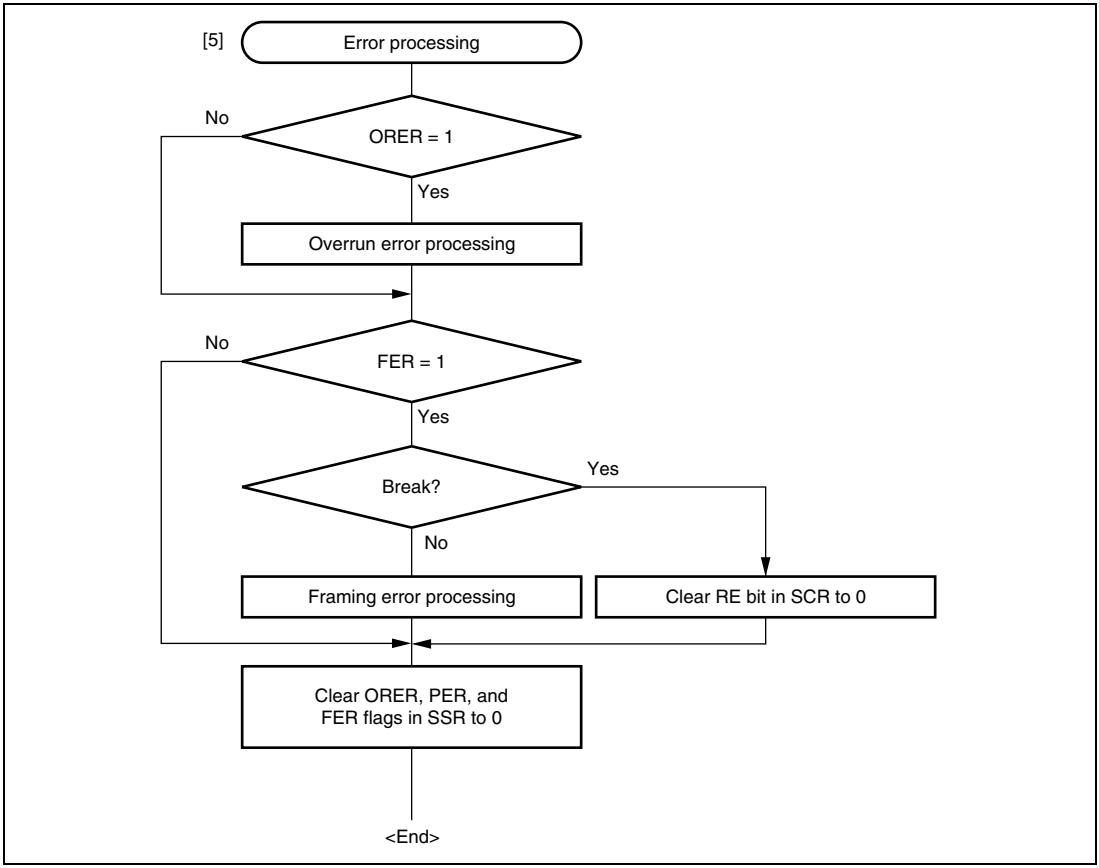


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)

13.6 Operation in Clocked Synchronous Mode

Figure 13.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received synchronous with clock pulses. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronous with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

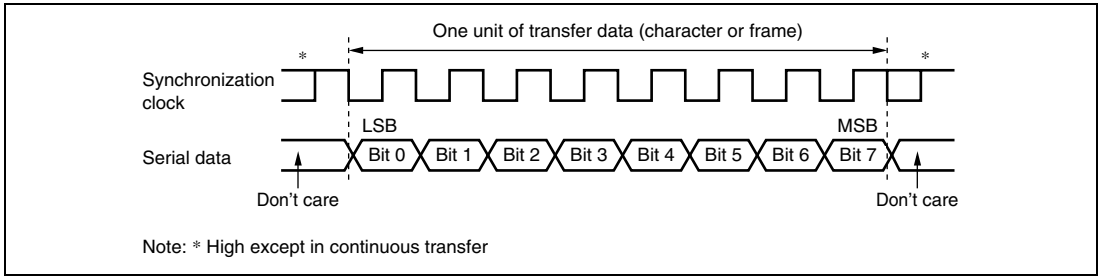


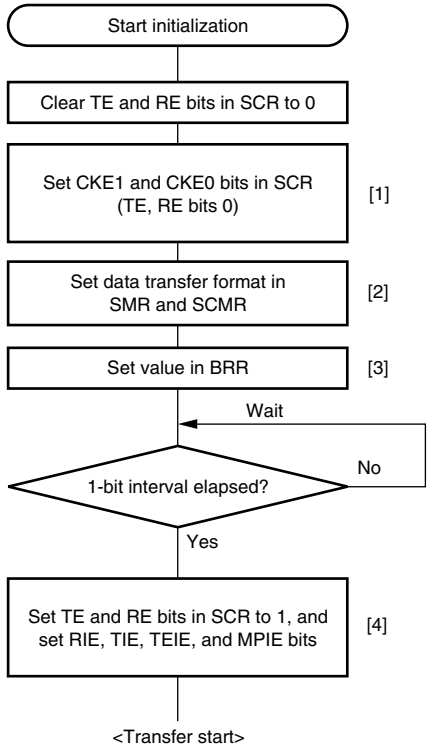
Figure 13.14 Data Format in Clocked Synchronous Communication (For LSB-First)

13.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CKE0 and CKE1 bits in SCR. When the SCI is operated on an internal clock, the serial clock is output from the SCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

13.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, the TE and RE bits in SCR should be cleared to 0, then the SCI should be initialized as described in a sample flowchart in figure 13.15. When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 13.15 Sample SCI Initialization Flowchart

13.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 13.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if the flag is 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated. Continuous transmission is possible because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has been completed.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified, and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 13.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

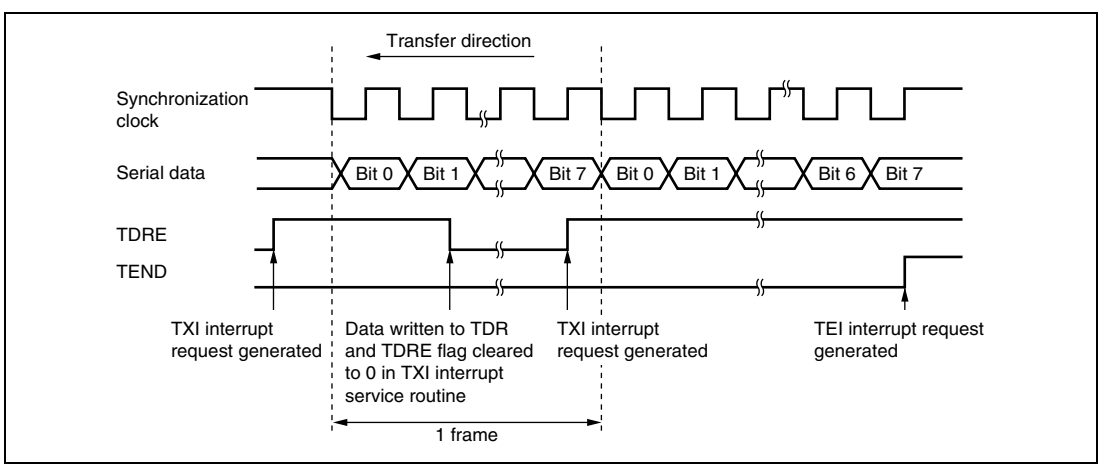
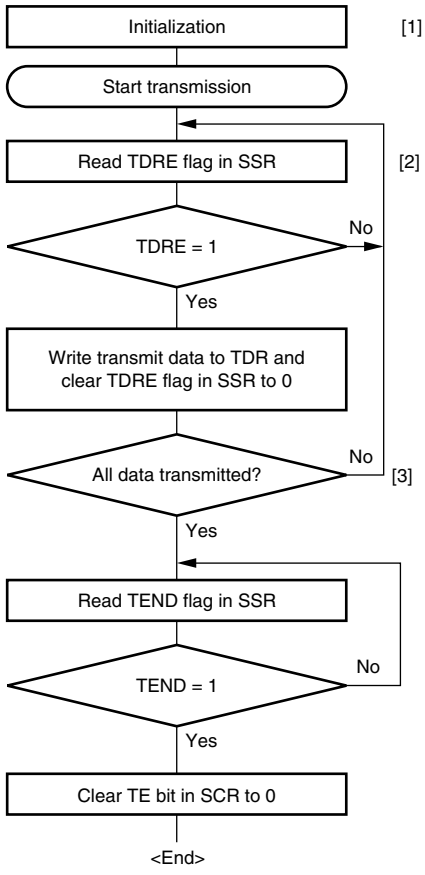


Figure 13.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Figure 13.17 Sample Serial Transmission Flowchart

13.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 13.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization synchronous with a synchronous clock input or output, starts receiving data, and stores the received data in RSR.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated, receive data is not transferred to RDR, and the RDRF flag remains to be set to 1.
3. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished.

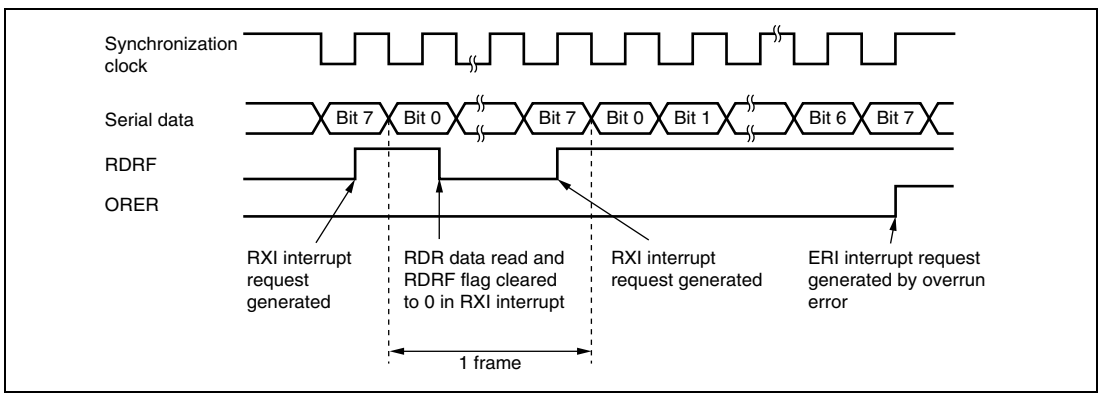
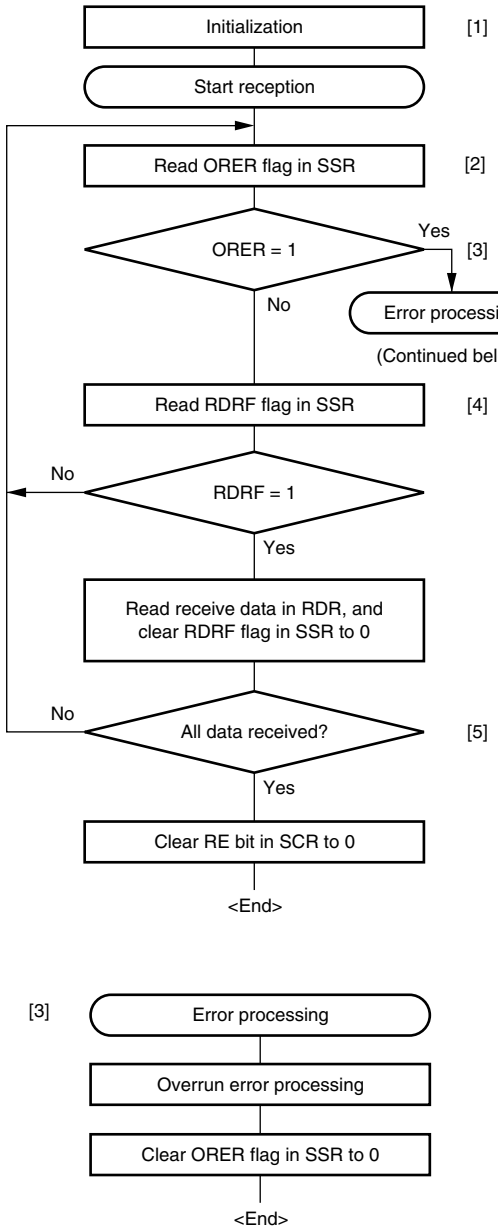


Figure 13.18 Example of SCI Operation in Reception

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample flow chart for serial data reception.

An overrun error occurs or synchronous clocks are output until the RE bit is cleared to 0 when an internal clock is selected and only receive operation is possible. When a transmission and reception will be carried out in a unit of one frame, be sure to carry out a dummy transmission with only one frame by the simultaneous transmit and receive operations at the same time.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

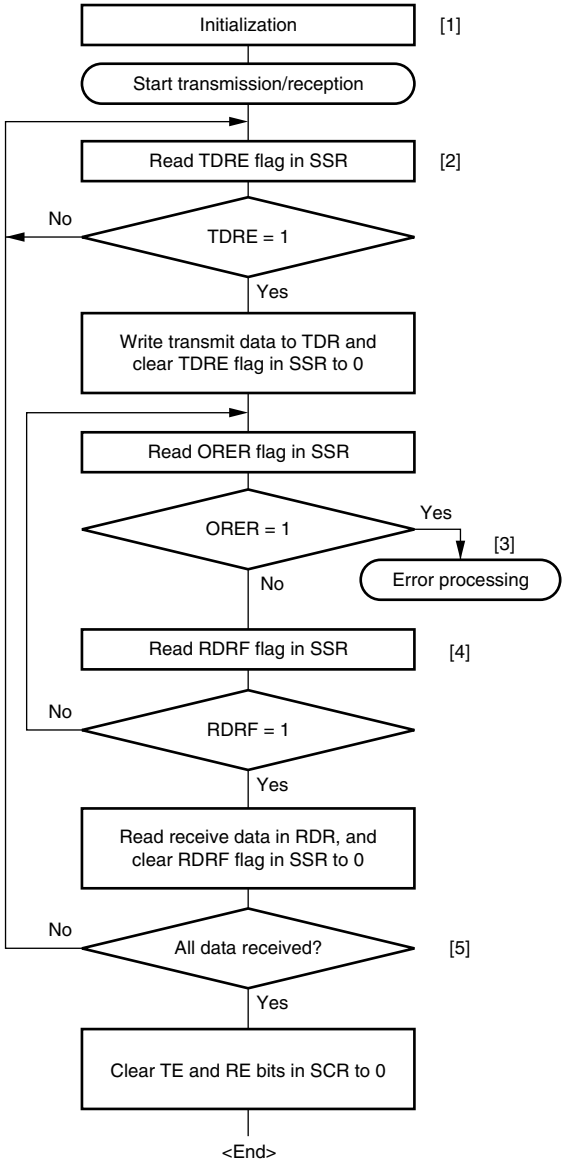
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0.
Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the final bit of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished. The RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Figure 13.19 Sample Serial Reception Flowchart

13.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 13.20 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the final bit of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0.
Also, before the final bit of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 by one instruction simultaneously.

Figure 13.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

13.7 Operation in Smart Card Interface

The SCI supports an IC card (Smart Card) interface that conforms to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function. Switching between the normal serial communication interface and the Smart Card interface mode is carried out by means of a register setting.

13.7.1 Pin Connection Example

Figure 13.21 shows an example of connection with the Smart Card. In communication with an IC card, as both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected to the LSI pin. The data transmission line should be pulled up to the power supply (other than channel 2: P2Vcc, channel 2: P1Vcc) with a resistor. If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out. When the clock generated on the SCI is supplied to an IC card, the SCK pin output is input to the CLK pin of the IC card. When an internal clock is used in an IC card, this connection is not necessary. This LSI port output is used as the reset signal. Adding to these connections, connection of pins with power supply and ground is necessary.

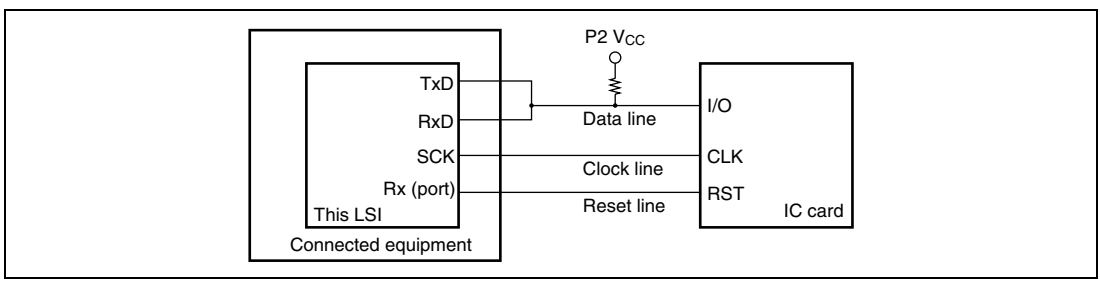


Figure 13.21 (1) Schematic Diagram of Smart Card Interface Pin Connections (Channels 0, 1, 3 and 4)

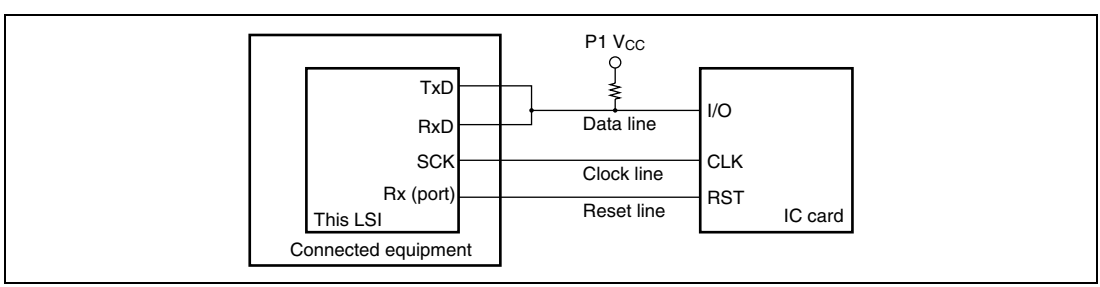


Figure 13.21 (2) Schematic Diagram of Smart Card Interface Pin Connections (Channel 2)

13.7.2 Data Format (Except for Block Transfer Mode)

Figure 13.22 shows the transfer data format in Smart Card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one bit) is left 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 one etu period, 10.5 etu after the start bit.
- If an error signal is sampled during transmission, the same data is retransmitted automatically after a delay of 2 etu or longer.

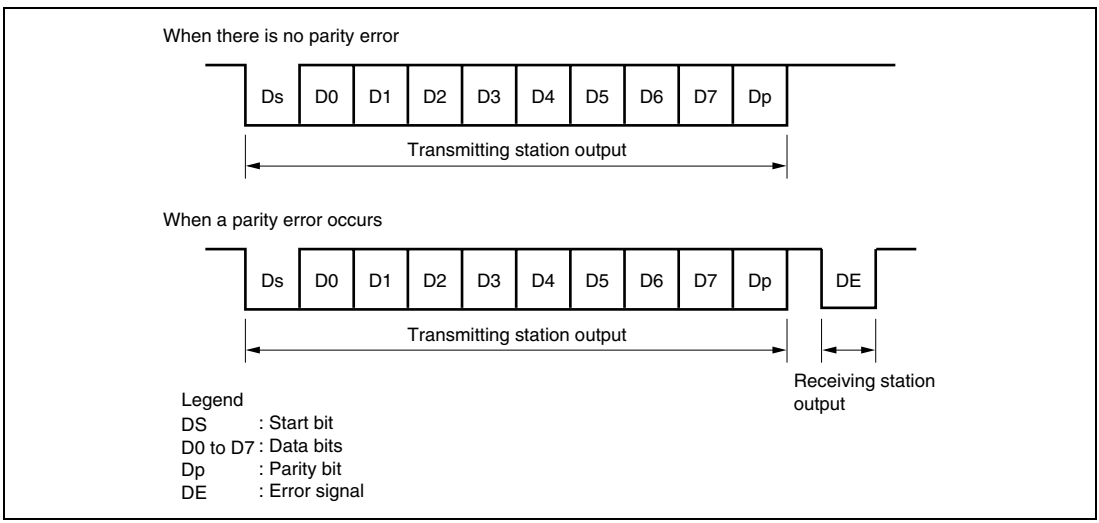


Figure 13.22 Normal Smart Card Interface Data Format

Data transfer with other types of IC cards (direct convention and inverse convention) are performed as described in the following.

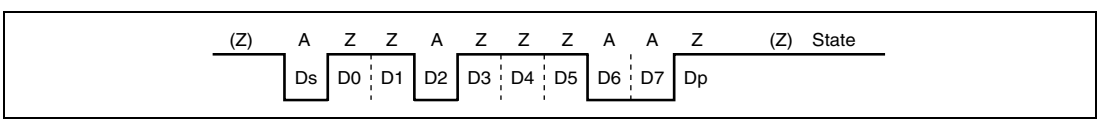


Figure 13.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

With the direction convention type IC and the above sample start character, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B. For the direct convention type, clear the SDIR and SINV bits in SCMR to 0. According to Smart Card regulations, clear the O/$\overline{E}$ bit in SMR to 0 to select even parity mode.

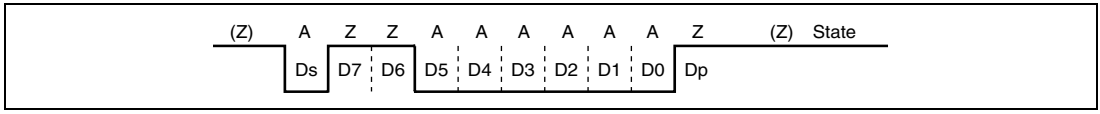


Figure 13.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data for the above is H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. According to Smart Card regulations, even parity mode is the logic 0 level of the parity bit, and corresponds to state Z. In this LSI, the SINV bit inverts only data bits D0 to D7. Therefore, set the O/$\overline{E}$ bit in SMR to 1 to invert the parity bit for both transmission and reception.

13.7.3 Block Transfer Mode

Operation in block transfer mode is the same as that in the normal Smart Card interface mode, except for the following points.

- In reception, though the parity check is performed, no error signal is output even if an error is detected. However, the PER bit in SSR is set to 1 and must be cleared before receiving the parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit and the start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, 11.5 etu after transmission start.
- As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transfer is not performed, this flag is always cleared to 0.

13.7.4 Receive Data Sampling Timing and Reception Margin

In Smart Card interface mode an internal clock generated by the on-chip baud rate generator can only be used as a transmission/reception clock. In this mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the transfer rate (fixed to 16 times in normal asynchronous mode) as determined by bits BCP1 and BCP0. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. As shown in figure 13.25, by sampling receive data at the rising-edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock, data can be latched at the middle of the bit. The reception margin is given by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|\,D - 0.5\,|}{N}(1 + F) \right| \times 100\%$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)

D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

$M = (0.5 - 1/2 \times 372) \times 100\%$
$= 49.866\%$

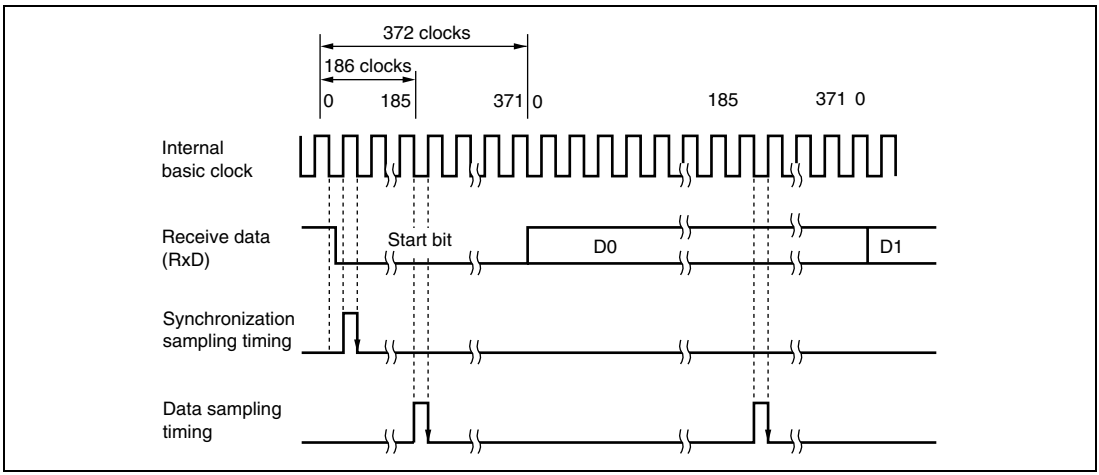


Figure 13.25 Receive Data Sampling Timing in Smart Card Interface Mode (Using Clock of 372 Times the Transfer Rate)

13.7.5 Initialization

Before transmitting and receiving data, initialize the SCI 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 in SCR to 0.
2. Clear the error flags ERS, PER, and ORER in SSR to 0.
3. Set the GM, BLK, O/$\overline{E}$, BCP0, BCP1, CKS0, CKS1 bits in SMR. Set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR.
   When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE 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.

To switch from receive mode to transmit mode, after checking that the SCI has finished reception, initialize the SCI, and set RE to 0 and TE to 1. Whether SCI has finished reception or not can be checked with the RDRF, PER, or ORER flags. To switch from transmit mode to receive mode, after checking that the SCI has finished transmission, initialize the SCI, and set TE to 0 and RE to 1. Whether SCI has finished transmission or not can be checked with the TEND flag.

13.7.6 Serial Data Transmission (Except for Block Transfer Mode)

As data transmission in Smart Card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 13.26 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sent back from the receiving end after transmission of one frame is complete, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be cleared to 0 by the time the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame in which an error signal indicating an abnormality is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set. Transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated. Writing transmit data to TDR transfers the next transmit data.

Figure 13.28 shows a flowchart for transmission. A sequence of transmit operations can be performed automatically by specifying the DTC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is set to 1 at the same time as the TEND flag in SSR is set, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC activation source, the DTC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data is transferred by the DTC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC is not activated. Therefore, the SCI and DTC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC before carrying out SCI setting. For details on the DTC setting procedures, refer to section 8, Data Transfer Controller (DTC).

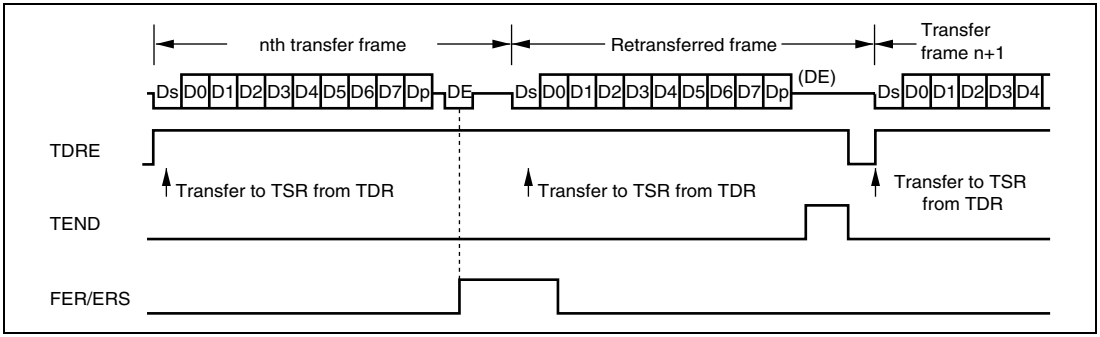


Figure 13.26 Retransfer Operation in SCI Transmit Mode

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag set timing is shown in figure 13.27.

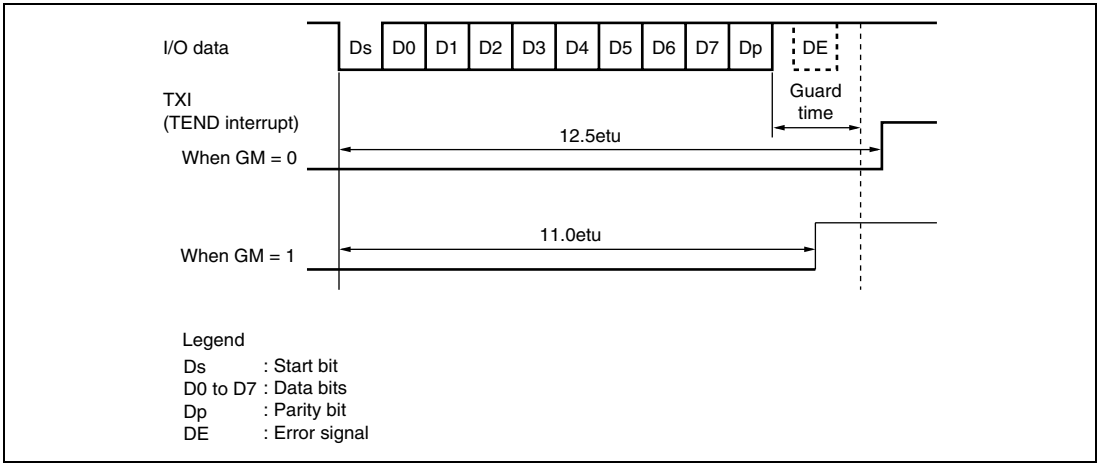


Figure 13.27 TEND Flag Generation Timing in Transmission Operation

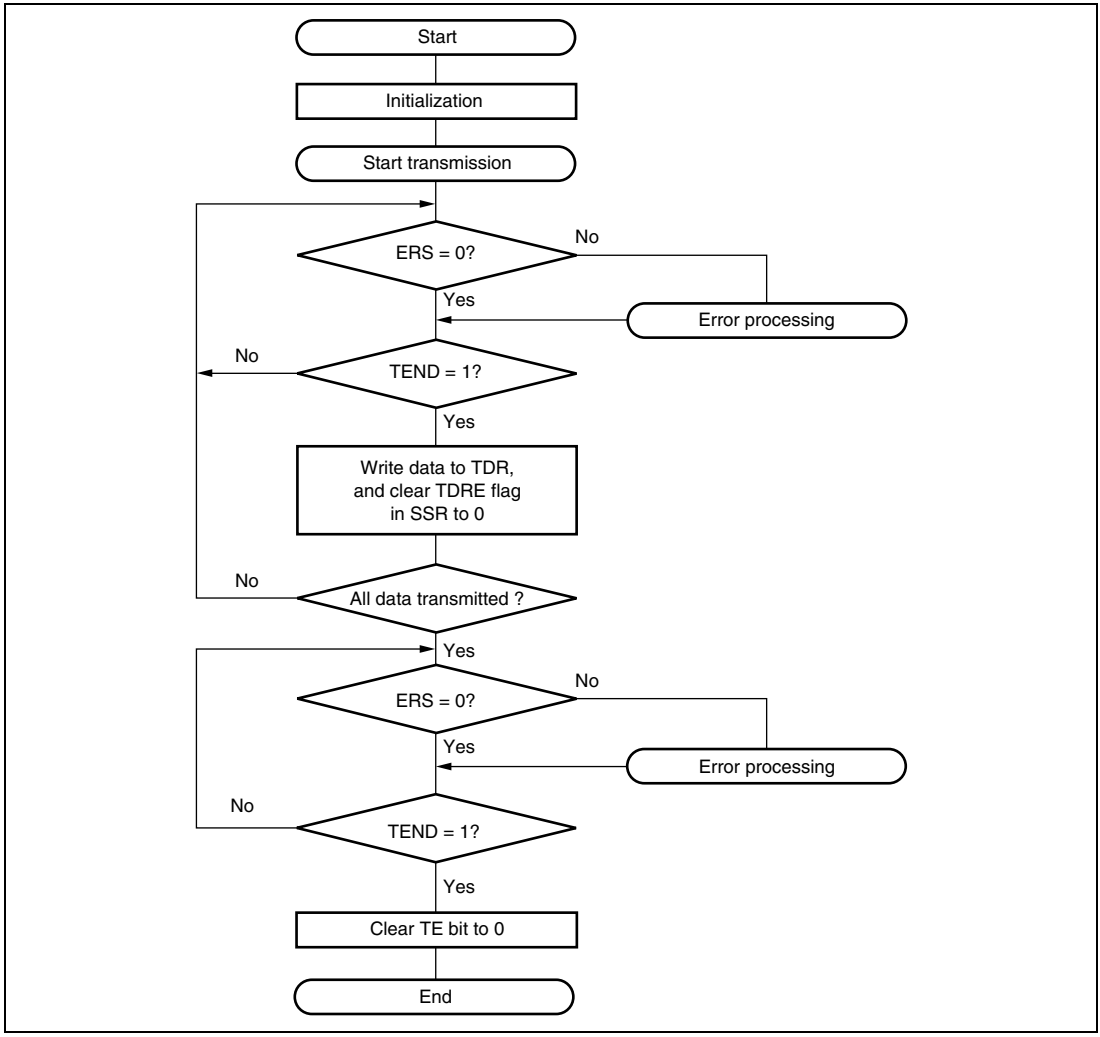


Figure 13.28 Example of Transmission Processing Flow

13.7.7 Serial Data Reception (Except for Block Transfer Mode)

Data reception in Smart Card interface mode uses the same operation procedure as for normal serial communication interface mode. Figure 13.29 illustrates the retransfer operation when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.
2. The RDRF bit in SSR is not set for a frame in which an error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.

Figure 13.30 shows a flowchart for reception. A sequence of receive operations can be performed automatically by specifying the DTC to be activated with an RXI interrupt source. In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC activation source, the DTC will be activated by the RXI request, and the receive data will be transferred. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC. If an error occurs in receive mode and the ORER or PER flag is set to 1, a transfer error interrupt (ERI) request will be generated. Hence, so the error flag must be cleared to 0. In the event of an error, the DTC is not activated and receive data is skipped. Therefore, receive data is transferred for only the specified number of bytes in the event of an error. Even when a parity error occurs in receive mode and the PER flag is set to 1, the data that has been received is transferred to RDR and can be read from there.

Note: For details on receive operations in block transfer mode, refer to section 13.4, Operation in Asynchronous Mode.

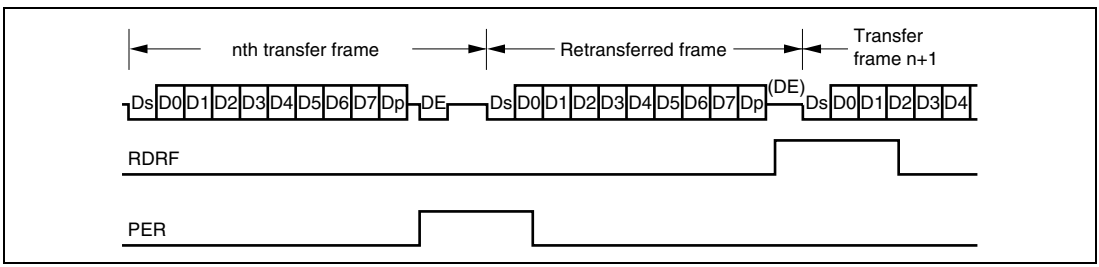


Figure 13.29 Retransfer Operation in SCI Receive Mode

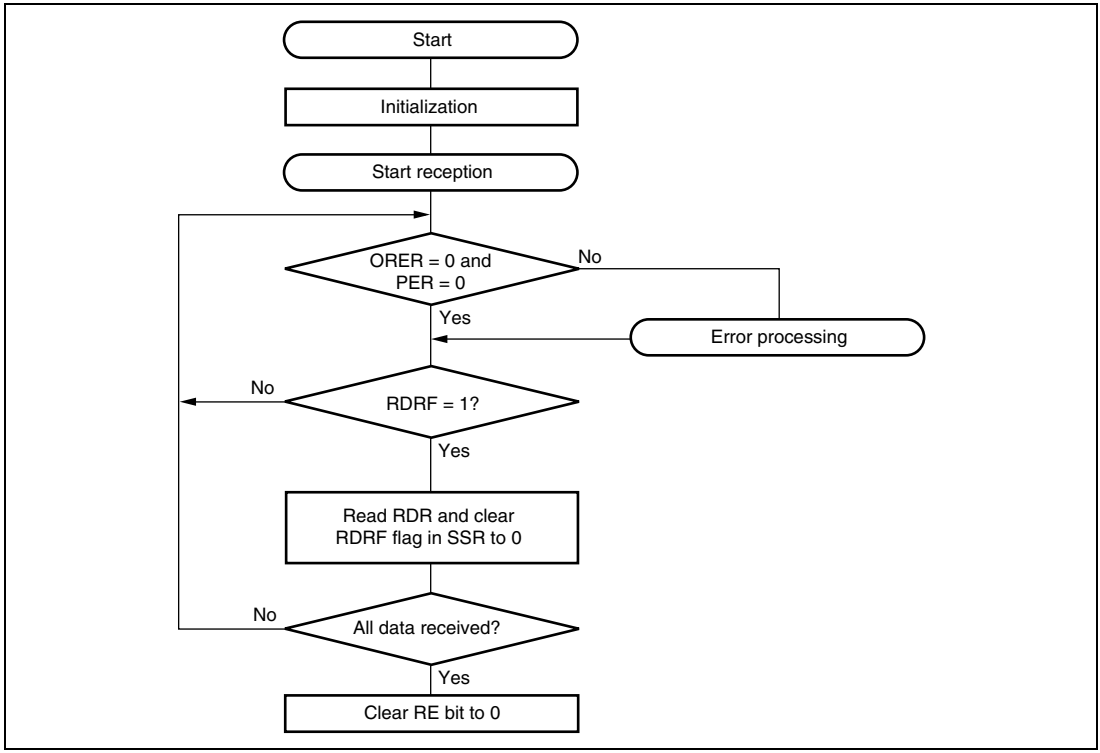


Figure 13.30 Example of Reception Processing Flow

13.7.8 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE0 and CKE1 in SCR. At this time, the minimum clock pulse width can be made the specified width. Figure 13.31 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

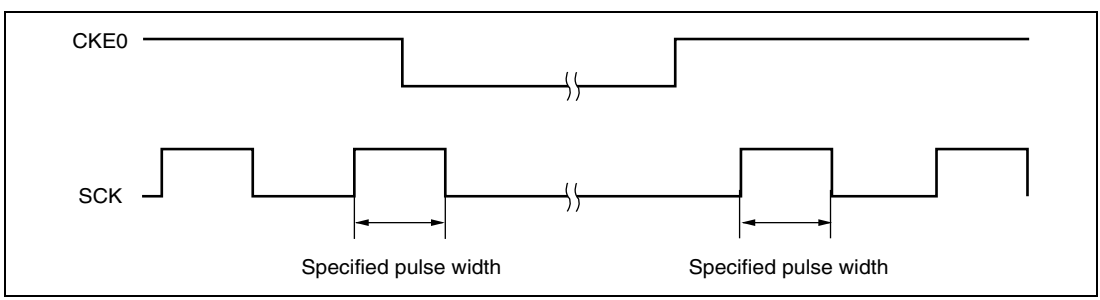


Figure 13.31 Timing for Fixing Clock Output Level

When turning on the power or switching between Smart Card interface mode and software standby mode, the following procedures should be followed in order to maintain the clock duty.

Powering on: To secure clock duty from power-on, the following switching procedure should be followed.

1. The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.
2. Fix the SCK pin to the specified output level with the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card mode operation.
4. Set the CKE0 bit in SCR to 1 to start clock output.

When changing from Smart Card interface mode to software standby mode:

1. Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.
2. Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to halt the clock.
4. Wait for one serial clock period.
   During this interval, clock output is fixed at the specified level, with the duty preserved.
5. Make the transition to the software standby state.

When returning to Smart Card interface mode from software standby mode:

1. Exit the software standby state.
2. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty.

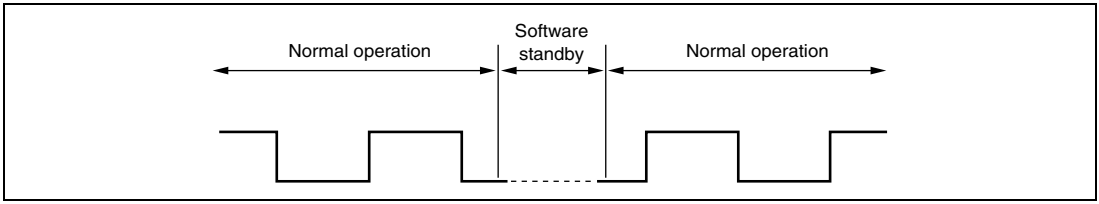


Figure 13.32 Clock Halt and Restart Procedure

13.8 Interrupt Sources

13.8.1 Interrupts in Normal Serial Communication Interface Mode

Table 13.12 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data is transferred by the DTC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DTC to transfer data. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 and the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority for acceptance. However, if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Table 13.12 Interrupt Sources of Serial Communication Interface Mode

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority*
0	ERI0	Receive Error	ORER, FER, PER	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	
	TXI0	Transmit Data Empty	TDRE	Possible	
	TEI0	Transmission End	TEND	Not possible	
1	ERI1	Receive Error	ORER, FER, PER	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	
	TXI1	Transmit Data Empty	TDRE	Possible	
	TEI1	Transmission End	TEND	Not possible	
2	ERI2	Receive Error	ORER, FER, PER	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	
	TXI2	Transmit Data Empty	TDRE	Possible	
	TEI2	Transmission End	TEND	Not possible	
3	ERI3	Receive Error	ORER, FER, PER	Not possible	
	RXI3	Receive Data Full	RDRF	Possible	
	TXI3	Transmit Data Empty	TDRE	Possible	
	TEI3	Transmission End	TEND	Not possible	
4	ERI4	Receive Error	ORER, FER, PER	Not possible	
	RXI4	Receive Data Full	RDRF	Possible	
	TXI4	Transmit Data Empty	TDRE	Possible	
	TEI4	Transmission End	TEND	Not possible	Low

Note: * Indicates the initial state immediately after a reset. Priorities in channels can be changed by the interrupt controller.

13.8.2 Interrupts in Smart Card Interface Mode

Table 13.13 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

Note: In case of block transfer mode, see section 13.8.1, Interrupts in Nomal Serial Communication Interface Mode.

Table 13.13 Interrupt Sources in Smart Card Interface Mode

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority*
0	ERI0	Receive Error, detection	ORER, PER, ERS	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	
	TXI0	Transmit Data Empty	TEND	Possible	
1	ERI1	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	
	TXI1	Transmit Data Empty	TEND	Possible	
2	ERI2	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	
	TXI2	Transmit Data Empty	TEND	Possible	
3	ERI3	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI3	Receive Data Full	RDRF	Possible	
	TXI3	Transmit Data Empty	TEND	Possible	
4	ERI4	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI4	Receive Data Full	RDRF	Possible	
	TXI4	Transmit Data Empty	TEND	Possible	Low

Note: * Indicates the initial state immediately after a reset. Priorities in channels can be changed by the interrupt controller.

13.9 Usage Notes

13.9.1 Module Stop Mode Setting

SCI operation can be disabled or enabled using the module stop control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

13.9.2 Break Detection and Processing (Asynchronous Mode Only)

When framing error (FER) detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, setting the FER flag, and possibly the PER flag. Note that as the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

13.9.3 Mark State and Break Detection (Asynchronous Mode Only)

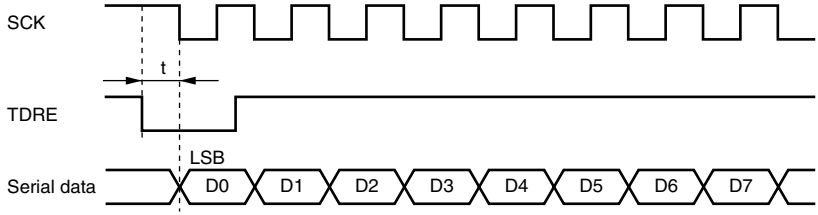
When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by DDR. This can be used to set the TxD pin to mark state (high level) or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both DDR and DR to 1. As TE is cleared to 0 at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial transmission, first set PDR to 1 and DR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

13.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

13.9.5 Restrictions on Use of DTC

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 ϕ clock cycles after TDR is updated by the DTC. Misoperation may occur if the transmit clock is input within 4 ϕ clocks after TDR is updated. (Figure 13.33)
- When RDR is read by the DTC, be sure to set the activation source to the relevant SCI reception data full interrupt (RXI).



Note: When operating on an external clock, set t >4 clocks.

Figure 13.33 Example of Clocked Synchronous Transmission by DTC

13.9.6 Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE bits to 0) before making a module stop mode, software standby mode, or watch mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode, software standby mode, or watch mode depend on the port settings, and become high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined. When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence: SSR read -> TDR write -> TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization. Figure 13.34 shows a sample flowchart for mode transition during transmission. Port pin states are shown in figures 13.35 and 13.36.

  Operation should also be stopped (by clearing TE, TIE, and TEIE bits to 0) before making a transition from transmission by DTC transfer to module stop mode, software standby mode, or watch mode. To perform transmission with the DTC after the relevant mode is cleared, setting the TE and TIE bits to 1 will set the TXI flag and start DTC transmission.

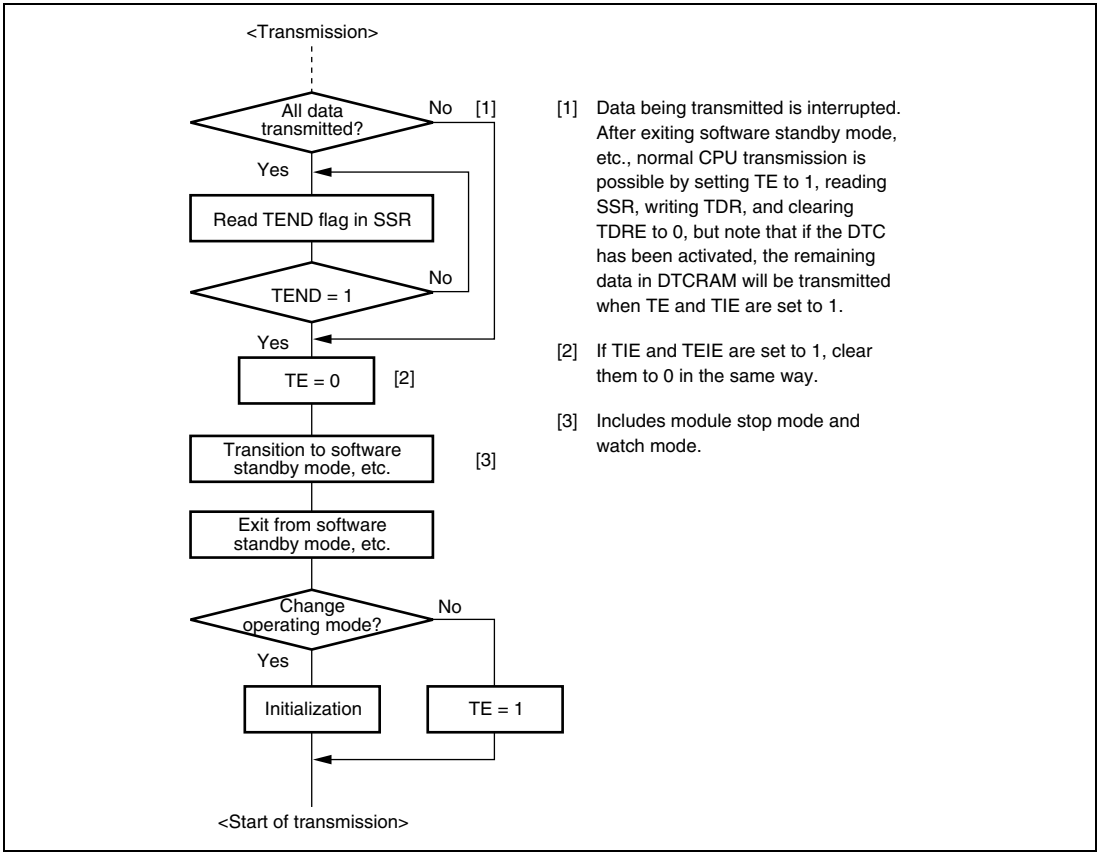


Figure 13.34 Sample Flowchart for Mode Transition during Transmission

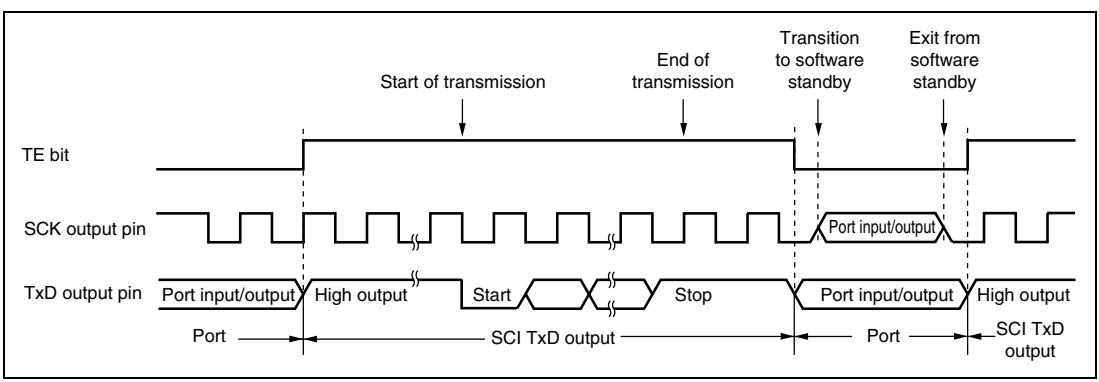


Figure 13.35 Asynchronous Transmission Using Internal Clock

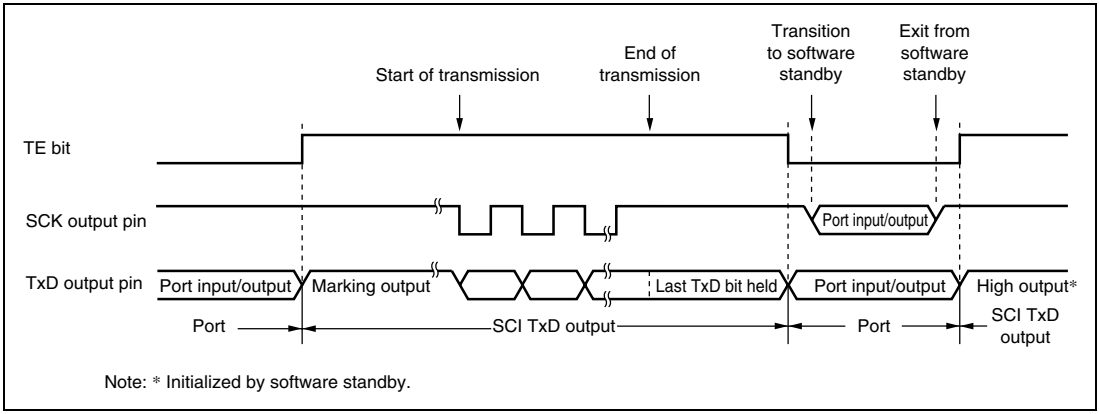


Figure 13.36 Clocked Synchronous Transmission Using Internal Clock

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode, software standby mode, or watch mode transition. RSR, RDR, and SSR are reset. If a transition is made without stopping operation, the data being received will be invalid.

  To continue receiving without changing the reception mode after the relevant mode is cleared, set RE to 1 before starting reception. To receive with a different receive mode, the procedure must be started again from initialization.

  Figure 13.37 shows a sample flowchart for mode transition during reception.

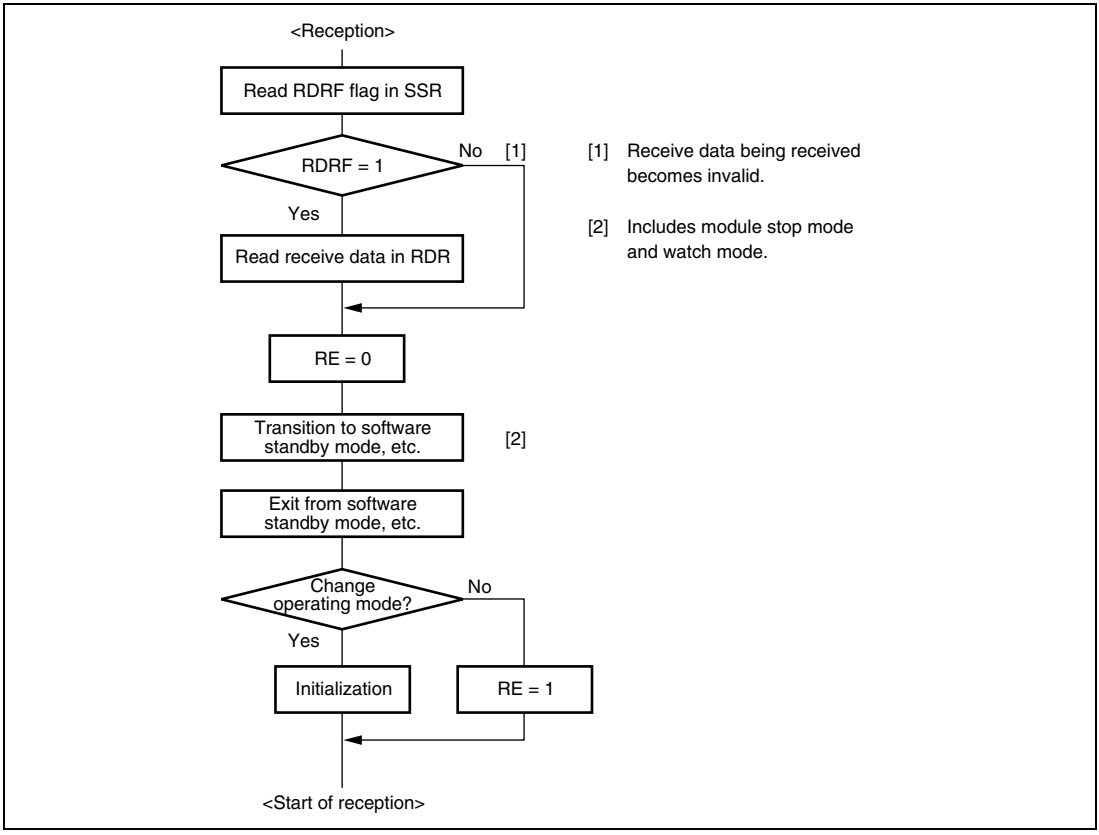


Figure 13.37 Sample Flowchart for Mode Transition during Reception

13.9.7 Notes when Switching from SCK Pin to Port Pin

- Problem in Operation: When DDR and DR are set to 1, SCI clock output is used in clocked synchronous mode, and the SCK pin is changed to the port pin while transmission is ended, port output is enabled after low-level output occurs for one half-cycle.

  When switching the SCK pin to the port pin by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, 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 13.38)

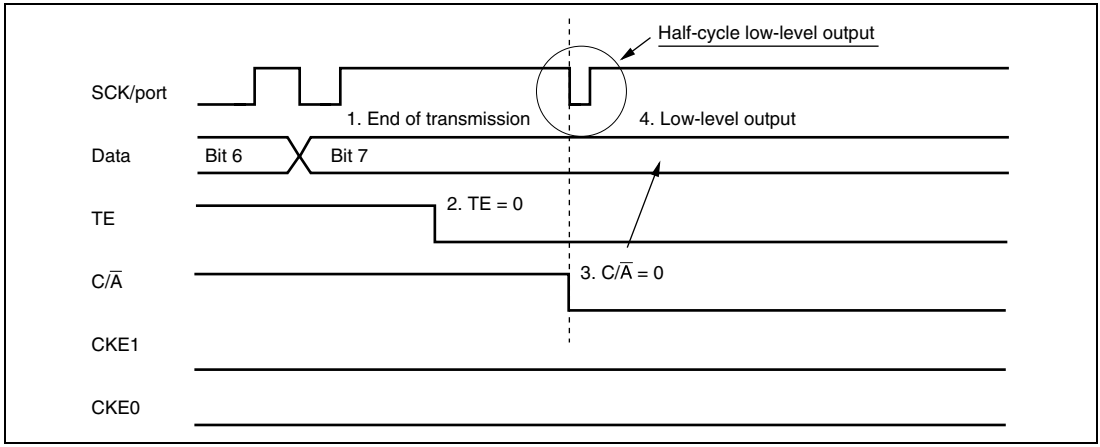


Figure 13.38 Operation when Switching from SCK Pin to Port Pin

- Usage Note: To prevent low-level output occurred when switching the SCK pin to port pin, follow the procedure described below.

  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

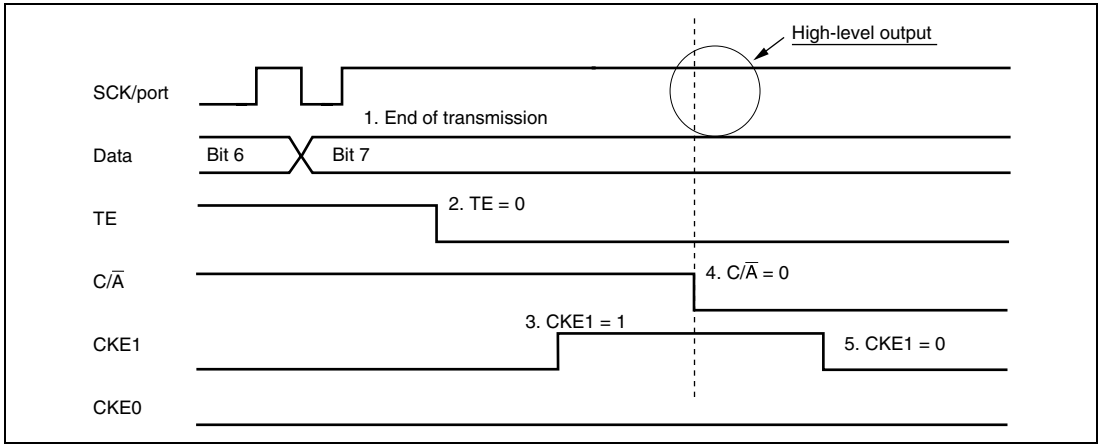


Figure 13.39 Operation when Switching from SCK Pin to Port Pin (Example of Preventing Low-Level Output)

Section 14 I^2C Bus Interface 2 (IIC2)

This LSI includes 2-channel I^2C bus interface. When using this function, please contact your Hitachi sales agency.

The I^2C bus interface 2 conforms to and provides a subset of the Philips I^2C bus (inter-IC bus) interface functions. The register configuration that controls the I^2C bus differs partly from the Philips configuration, however.

Figure 14.1 shows a block diagram of the I^2C bus interface 2.

Figure 14.2 shows an example of I/O pin connections to external circuits.

14.1 Features

- Selection of I^2C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.

I^2C bus format

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically.

  If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive

  Two pins, pins P35/SCL0 and P34/SDA0, function as NMOS open-drain outputs when the bus drive function is selected.

  Two pins, pins P33/SCL1 and P32/SDA1, are driven only by NMOS transistors when the bus drive function is selected.

Clocked synchronous format

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error

The I/O pins for channel 0 function as NMOS open-drain outputs, and it is possible to apply voltages in excess of the power supply (P2Vcc) voltage for this LSI. The maximum voltage must not exceed 0.3 V + this LSI's power supply voltage (P2Vcc). Since the I/O pins for channel 1 are driven only by NMOS transistors, so in terms of appearance they carry out the same operations as an NMOS open drain. However, the voltage which can be applied to the I/O pins depends on the voltage of the power supply (P2Vcc) of this LSI.

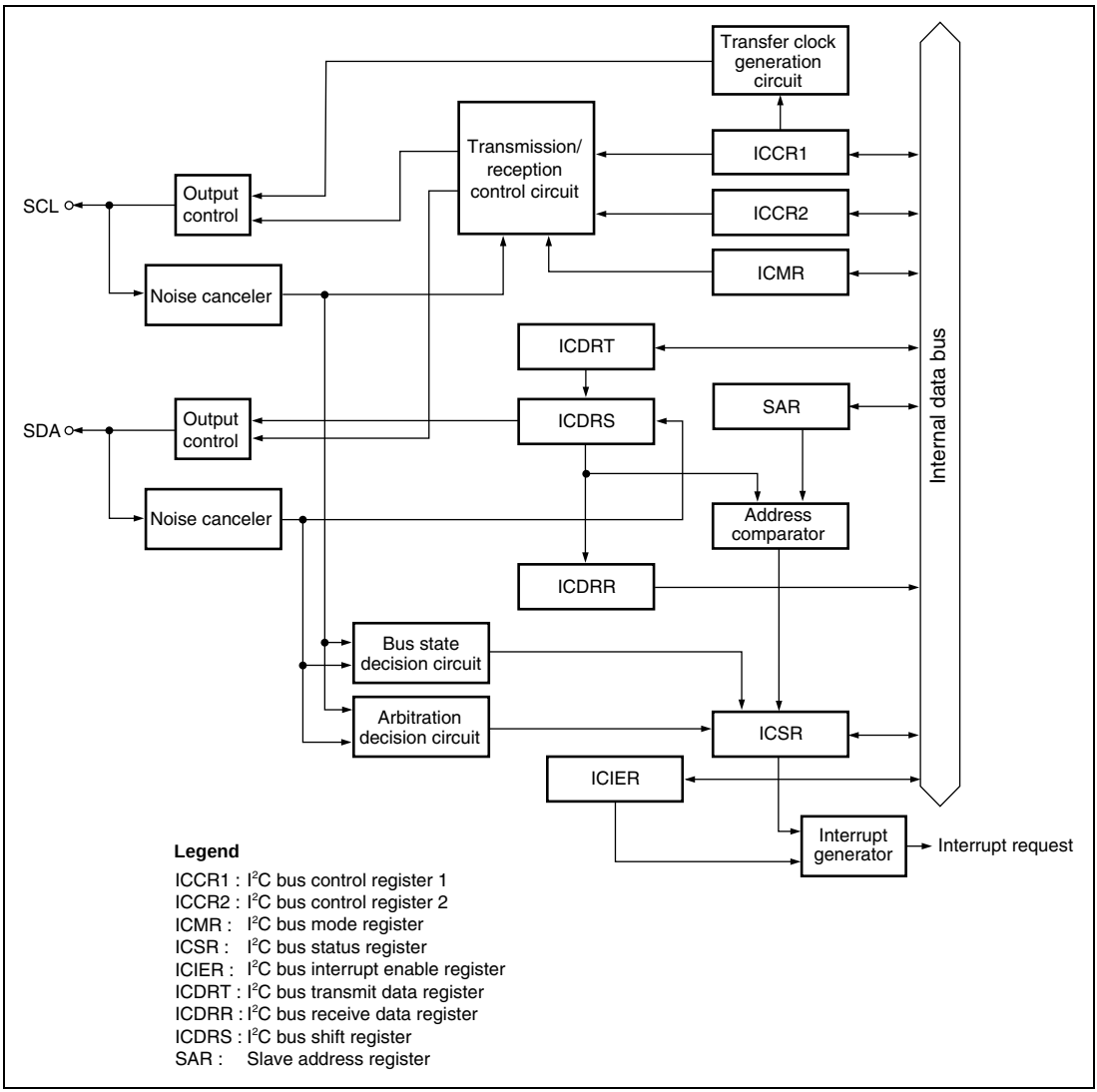


Figure 14.1 Block Diagram of I²C Bus Interface 2

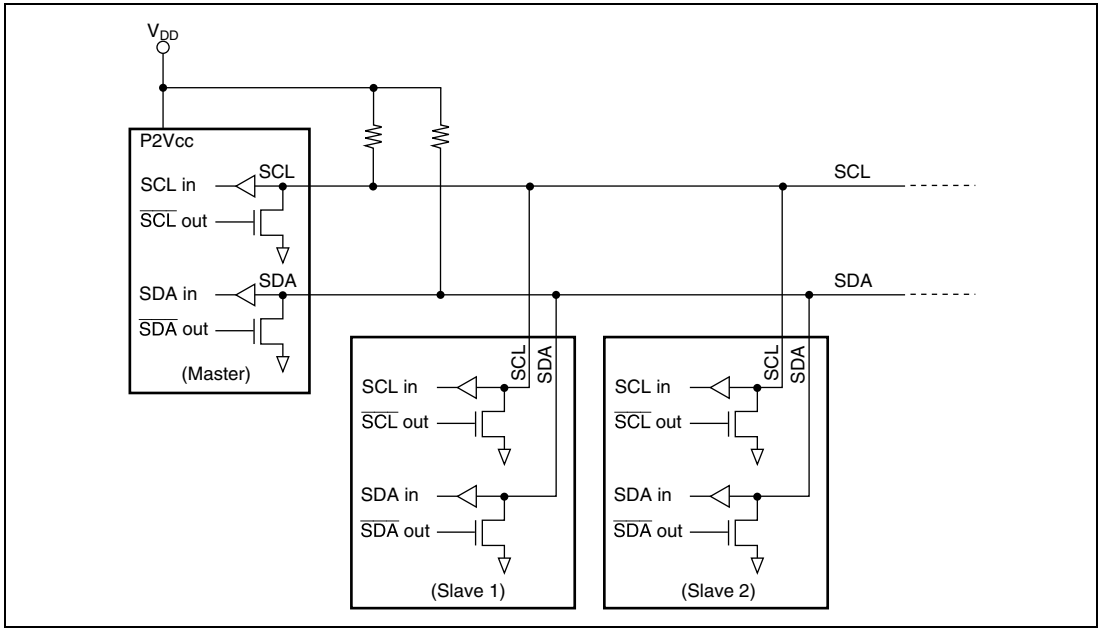


Figure 14.2 External Circuit Connections of I/O Pins

14.2 Input/Output Pins

Table 14.1 summarizes the input/output pins used by the I^2C bus interface 2.

Table 14.1 I^2C Bus Interface Pins

Channel	Abbreviation	I/O	Description
0	SCL0	I/O	Serial clock input/output for channel 0
	SDA0	I/O	Serial data input/output for channel 0
1	SCL1	I/O	Serial clock input/output for channel 1
	SDA1	I/O	Serial data input/output for channel 1

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used.

14.3 Register Descriptions

The I²C bus interface 2 has the following registers.

- Channel 0
  I²C bus control register 1_0 (ICCR1_0)
  I²C bus control register 2_0 (ICCR2_0)
  I²C bus mode register_0 (ICMR_0)
  I²C bus interrupt enable register_0 (ICIER_0)
  I²C bus status register_0 (ICSR_0)
  I²C bus slave address register_0 (SAR_0)
  I²C bus transmit data register_0 (ICDRT_0)
  I²C bus receive data register_0 (ICDRR_0)
  I²C bus shift register_0 (ICDRS_0)

- Channel 1
  I²C bus control register 1_1 (ICCR1_1)
  I²C bus control register 2_1 (ICCR2_1)
  I²C bus mode register_1 (ICMR_1)
  I²C bus interrupt enable register_1 (ICIER_1)
  I²C bus status register_1 (ICSR_1)
  I²C bus slave address register_1 (SAR_1)
  I²C bus transmit data register_1 (ICDRT_1)
  I²C bus receive data register_1 (ICDRR_1)
  I²C bus shift register_1 (ICDRS_1)

14.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 enables or disables the I²C bus interface 2, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I^2C Bus Interface Enable 0: This module is halted. (SCL and SDA pins are set to port function.) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I^2C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. After data receive has been started in slave receive mode, when the first seven bits of the receive data agree with the slave address that is set to SAR and the eighth bit is 1, TRS is automatically set to 1. If an overrun error occurs in master mode with the clock synchronous serial format, MST is cleared to 0 and slave receive mode is entered. Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST is 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer Clock Select 3 to 0 These bits are valid only in master mode and should be set according to the necessary transfer rate. For details on transfer rate, see table 14.2, Transfer Rate.
2	CKS2	0	R/W	
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Table 14.2 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transφer Rate				
CKS3	CKS2	CKS1	CKS0	Clock	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz	φ = 25 MHz
0	0	0	0	φ/28	286 kHz	357 kHz	571 kHz	714 kHz	893 kHz
			1	φ/40	200 kHz	250 kHz	400 kHz	500 kHz	625 kHz
		1	0	φ/48	167 kHz	208 kHz	333 kHz	417 kHz	521 kHz
			1	φ/64	125 kHz	156 kHz	250 kHz	313 kHz	391 kHz
	1	0	0	φ/168	47.6 kHz	59.5 kHz	95.2 kHz	119 kHz	149 kHz
			1	φ/100	80.0 kHz	100 kHz	160 kHz	200 kHz	250 kHz
		1	0	φ/112	71.4 kHz	89.3 kHz	143 kHz	179 kHz	223 kHz
			1	φ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz	195 kHz
1	0	0	0	φ/56	143 kHz	179 kHz	286 kHz	357 kHz	446 kHz
			1	φ/80	100 kHz	125 kHz	200 kHz	250 kHz	313 kHz
		1	0	φ/96	83.3 kHz	104 kHz	167 kHz	208 kHz	260 kHz
			1	φ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz	195 kHz
	1	0	0	φ/336	23.8 kHz	29.8 kHz	47.6 kHz	59.5 kHz	74.4 kHz
			1	φ/200	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz	125 kHz
		1	0	φ/224	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz	112 kHz
			1	φ/256	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz	97.7 kHz

14.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR1 issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus interface 2.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy This bit enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clocked synchronous serial format, this bit has no meaning. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition. To issue start/stop conditions, use the MOV instruction.
6	SCP	1	W	Start/Stop Issue Condition Disable The SCP bit controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).
4	SDAOP	1	R/W	SDAO Write Protect This bit controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0 by the MOV instruction. This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
3	SCLO	1	R	This bit monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	—	Reserved This bit is always read as 1, and cannot be modified.
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I^2C2 registers. If this bit is set to 1 when hang-up occurs because of communication failure during I^2C2 operation, I^2C2 control part can be reset without setting ports and initializing registers.
0	—	1	—	Reserved This bit is always read as 1, and cannot be modified.

14.3.3 I^2C Bus Mode Register (ICMR)

ICMR selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I^2C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit In master mode with the I^2C bus format, this bit selects whether to insert a wait after data transfer except the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The setting of this bit is invalid in slave mode with the I^2C bus format or with the clocked synchronous serial format.
5, 4	—	All 1	—	Reserved These bits are always read as 1, and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. In clock synchronous serial mode, BC should not be modified. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are invalid.
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I^2C bus format, the data is transferred with one addition acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL pin is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. With the clock synchronous serial format, these bits should not be modified. I^2C Bus Format / Clock Synchronous Serial Format 000: 9 bits / 000: 8 bits 001: 2 bits / 001: 1 bits 010: 3 bits / 010: 2 bits 011: 4 bits / 011: 3 bits 100: 5 bits / 100: 4 bits 101: 6 bits / 101: 5 bits 110: 7 bits / 110: 6 bits 111: 8 bits / 111: 7 bits
0	BC0	0	R/W	

14.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits to be received.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data full interrupt request (RXI) and the overrun error interrupt request (ERI) with the clocked synchronous format, when a receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clocked synchronous format are disabled. 1: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clocked synchronous format are enabled.
4	NAKIE	0	R/W	NACK Receive Interrupt Enable This bit enables or disables the NACK receive interrupt request (NAKI) and the overrun error (setting of the OVE bit in ICSR) interrupt request (ERI) with the clocked synchronous format, when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.

Bit	Bit Name	Initial Value	R/W	Description
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgement Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

14.3.5 I²C Bus Status Register (ICSR)

ICSR performs confirmation of interrupt request flags and status.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When a start condition (including re-transfer) has been issued • When transmit mode is entered from receive mode in slave mode [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT with an instruction
6	TEND	0	R/W	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • When the final bit of transmit frame is sent with the clock synchronous serial format [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT with an instruction
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When a receive data is transferred from ICDRS to ICDRR [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read with an instruction

Bit	Bit Name	Initial Value	R/W	Description
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1 [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1
3	STOP	0	R/W	Stop Condition Detection Flag [Setting condition] • When a stop condition is detected after frame transfer [Clearing condition] • When 0 is written in STOP after reading STOP = 1
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag This flag indicates that arbitration was lost in master mode with the I^2C bus format and that the final bit has been received while RDRF = 1 with the clocked synchronous format. When two or more master devices attempt to seize the bus at nearly the same time, if the I^2C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master. [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • When the SDA pin outputs high in master mode while a start condition is detected • When the final bit is received with the clocked synchronous format while RDRF = 1 [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE=1

Bit	Bit Name	Initial Value	R/W	Description
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode. [Clearing condition] • When 0 is written in AAS after reading AAS=1
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in I^2C bus format slave receive mode. [Setting condition] • When the general call address is detected in slave receive mode [Clearing conditions] • When 0 is written in ADZ after reading ADZ=1

14.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in slave mode with the I^2C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	0	R/W	Slave Address 6 to 0
6	SVA5	0	R/W	These bits set a unique address in bits SVA6 to SVA0, differing form the addresses of other slave devices connected to the I^2C bus.
5	SVA4	0	R/W	
4	SVA3	0	R/W	
3	SVA2	0	R/W	
2	SVA1	0	R/W	
1	SVA0	0	R/W	
0	FS	0	R/W	Format Select 0: I^2C bus format is selected. 1: Clocked synchronous serial format is selected.

14.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. If the MLS bit of ICMR is set to 1 and when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial value of ICDRT is H′FF.

14.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register. The initial value of ICDRR is H′FF.

14.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

14.4 Operation

The I²C bus interface can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

14.4.1 I²C Bus Format

Figure 14.3 shows the I²C bus formats. Figure 14.4 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

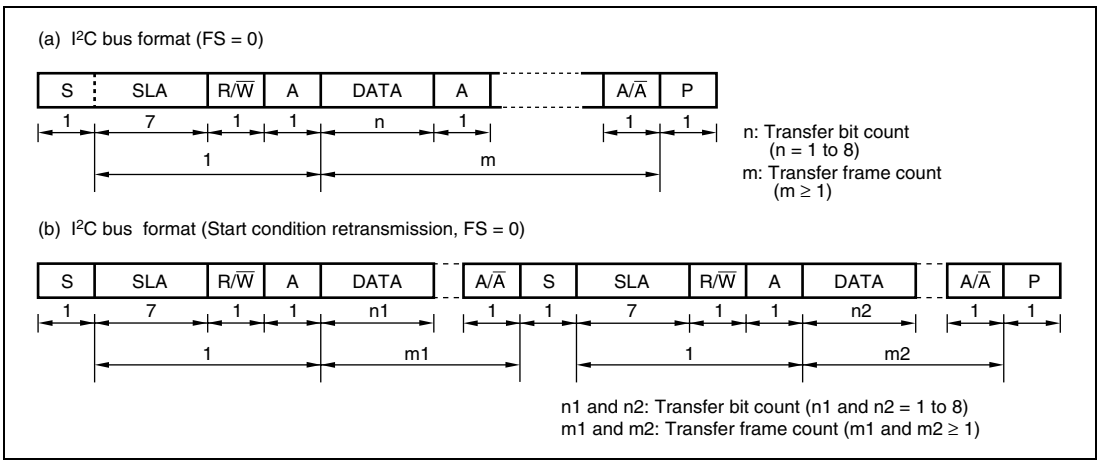


Figure 14.3 I²C Bus Formats

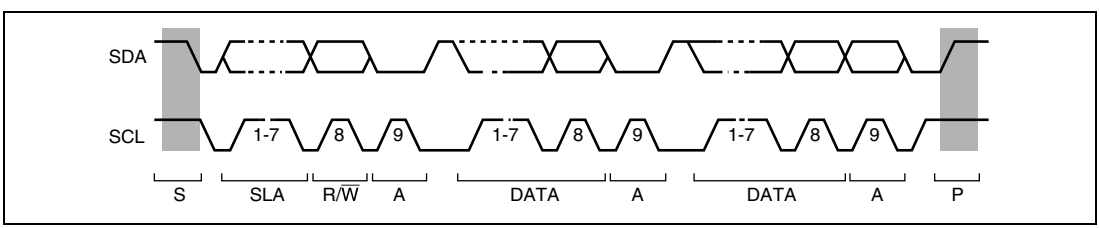


Figure 14.4 I²C Bus Timing

Legend

S: Start condition. The master device drives SDA from high to low while SCL is high.
SLA: Slave address
R/W: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.
A: Acknowledge. The receive device drives SDA to low.
DATA: Transfer data
P: Stop condition. The master device drives SDA from low to high while SCL is high.

14.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 14.5 and 14.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

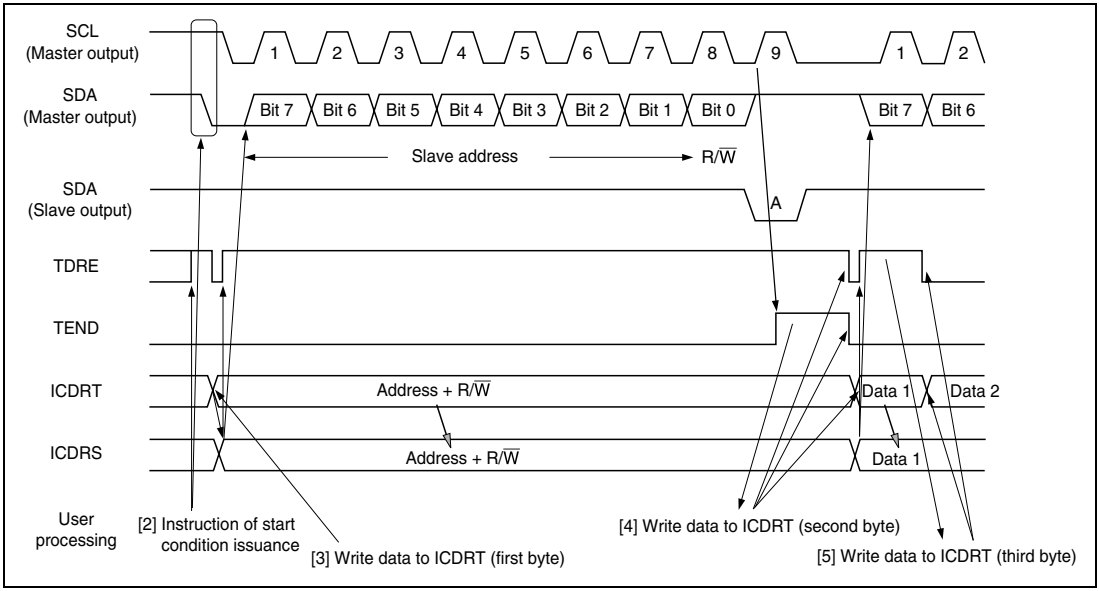


Figure 14.5 Master Transmit Mode Operation Timing (1)

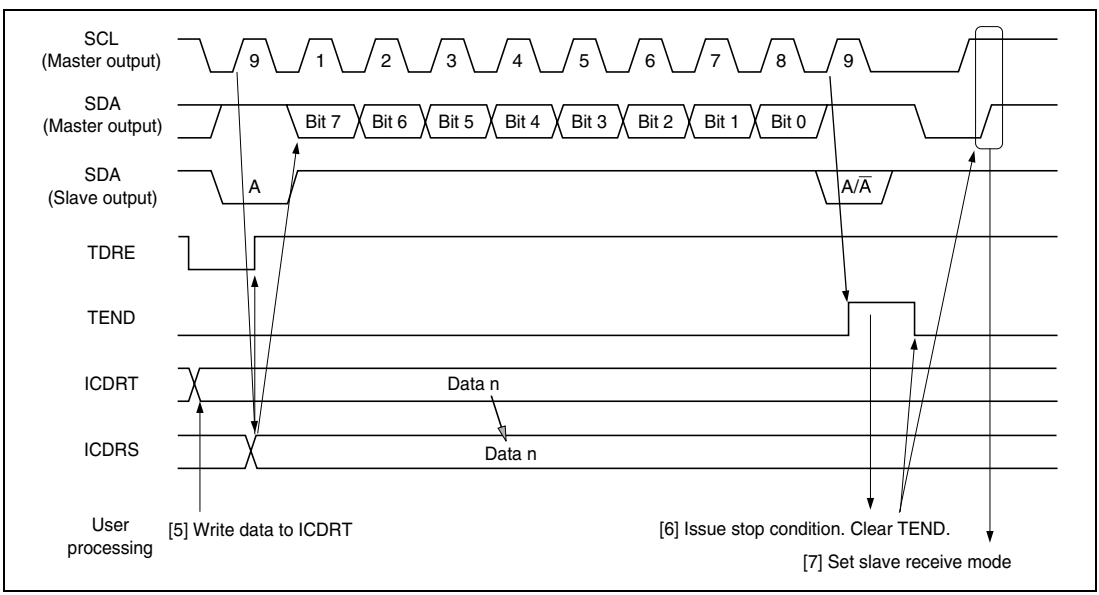


Figure 14.6 Master Transmit Mode Operation Timing (2)

14.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 14.7 and 14.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

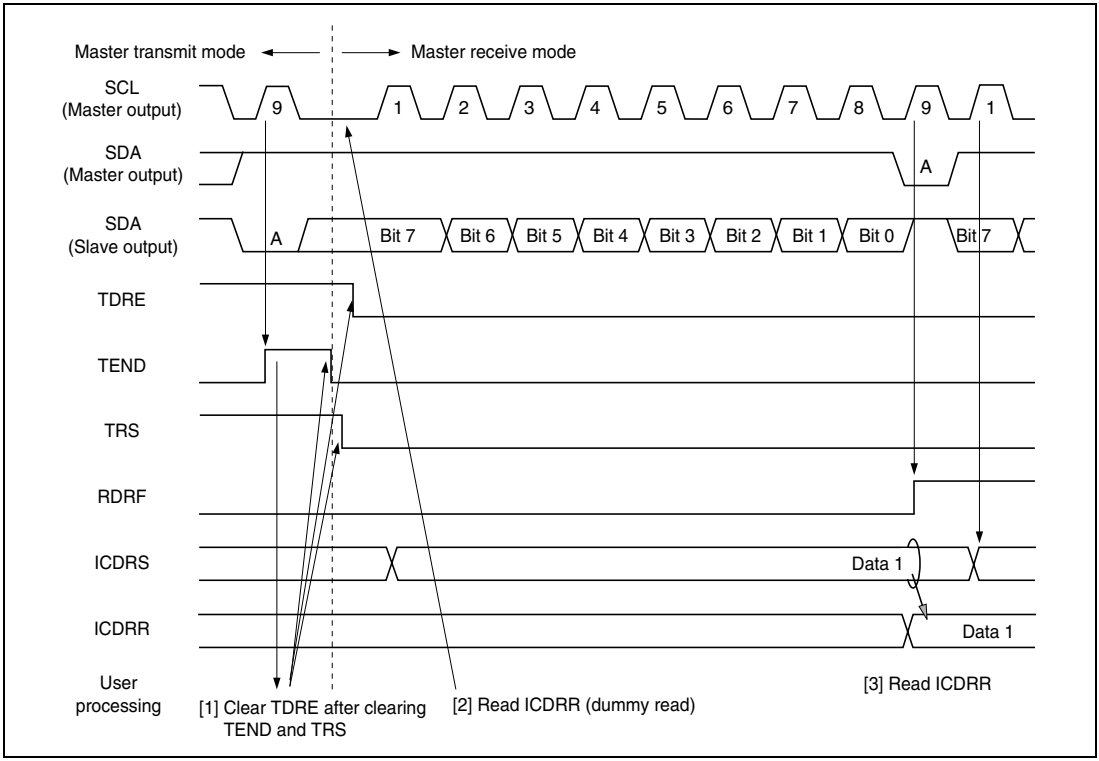


Figure 14.7 Master Receive Mode Operation Timing (1)

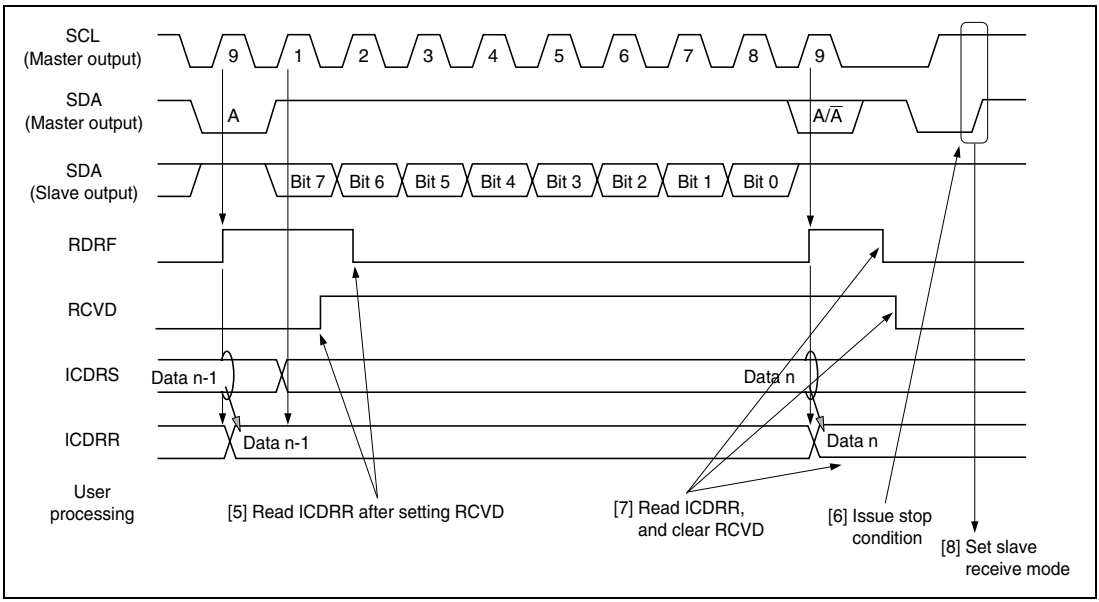


Figure 14.8 Master Receive Mode Operation Timing (2)

14.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 14.9 and 14.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in ICCR1 are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear TDRE.

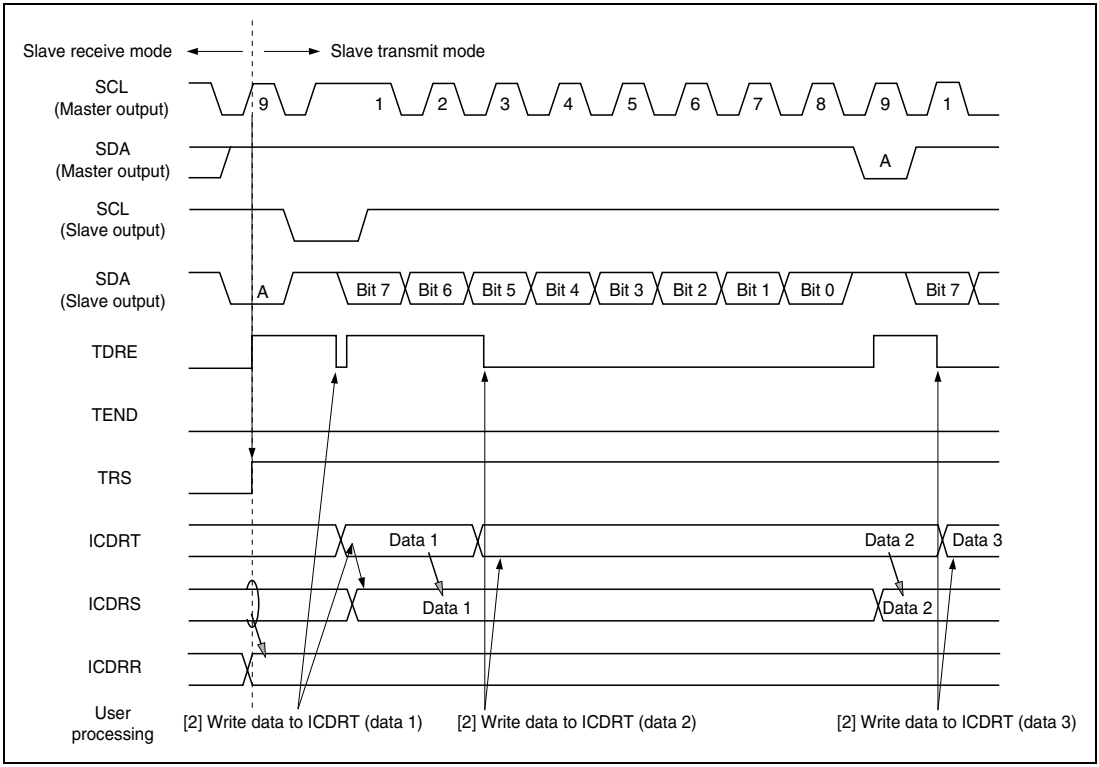


Figure 14.9 Slave Transmit Mode Operation Timing (1)

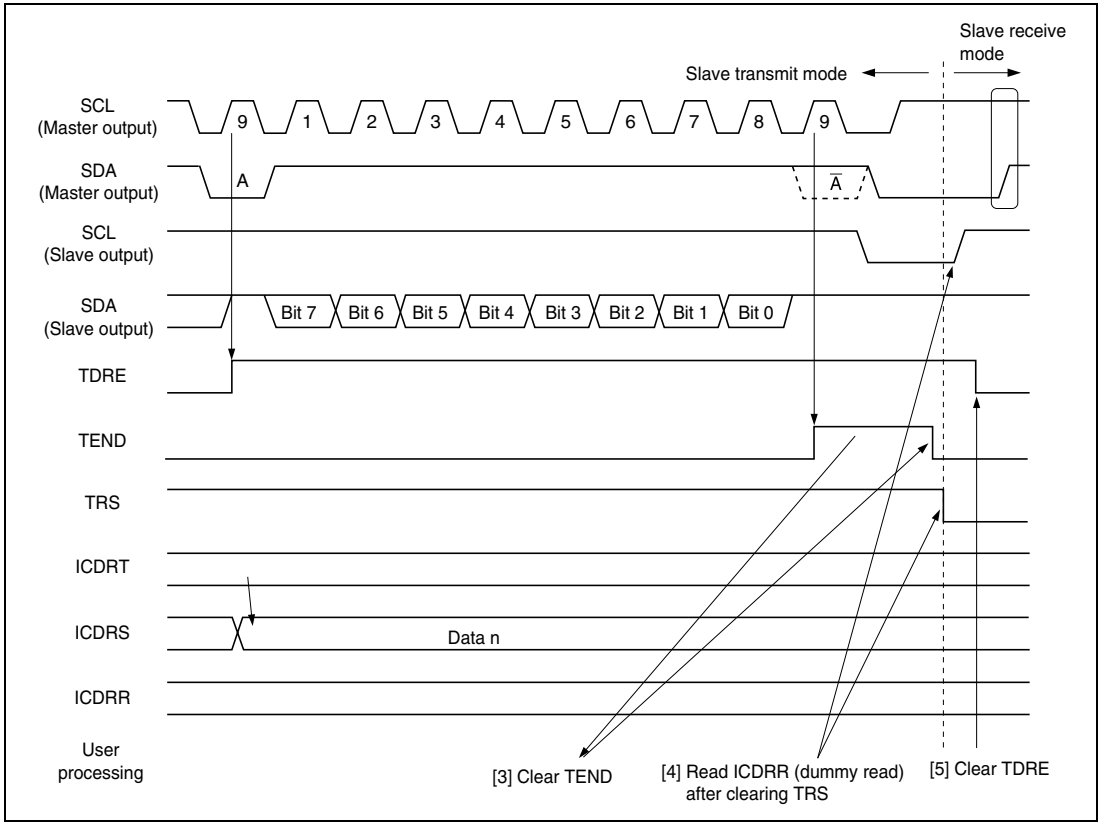


Figure 14.10 Slave Transmit Mode Operation Timing (2)

14.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 14.11 and 14.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.

4. The last byte data is read by reading ICDRR.

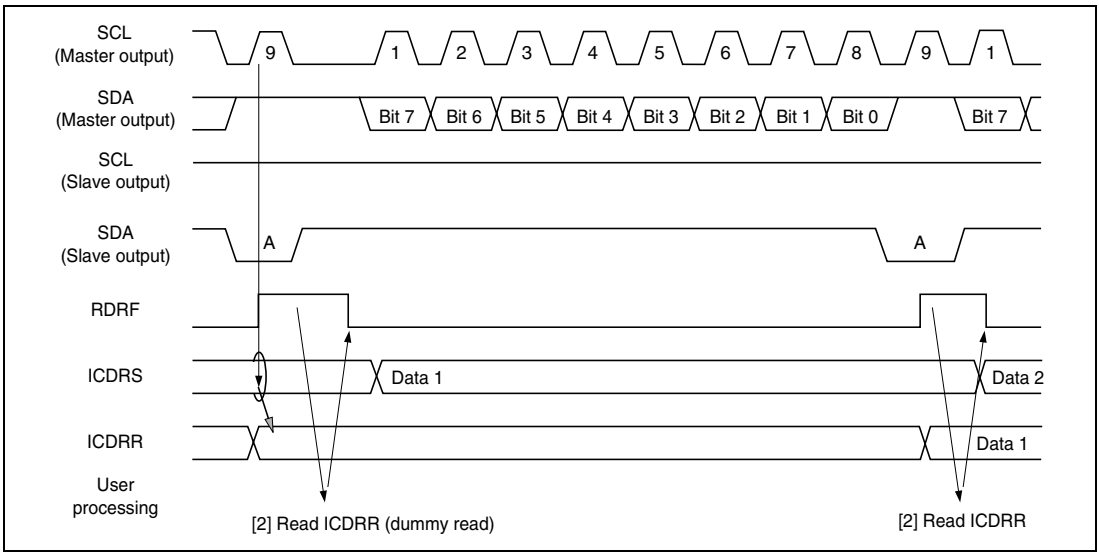


Figure 14.11 Slave Receive Mode Operation Timing (1)

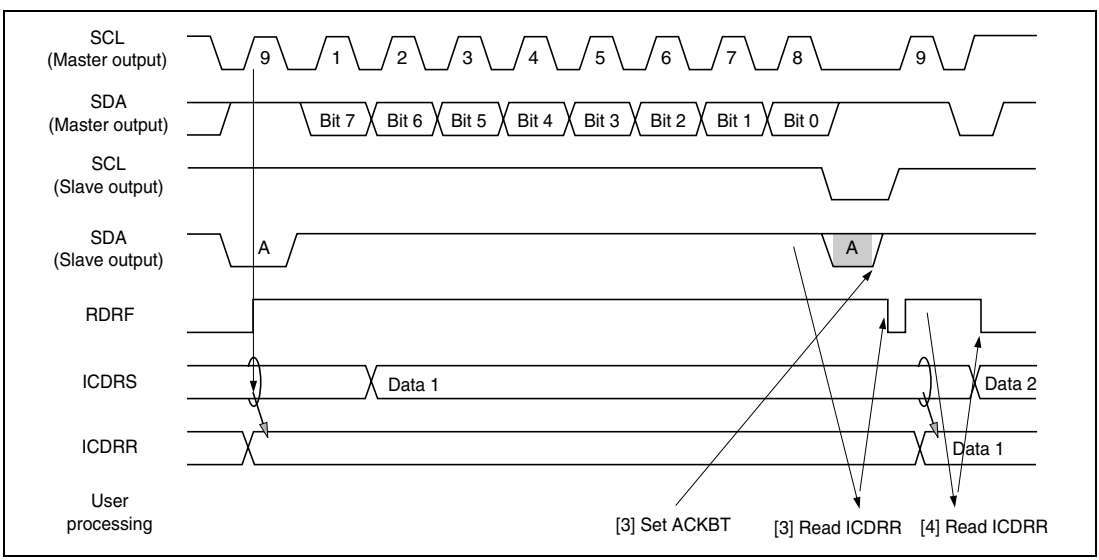


Figure 14.12 Slave Receive Mode Operation Timing (2)

14.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

Data Transfer Format: Figure 14.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the rise to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

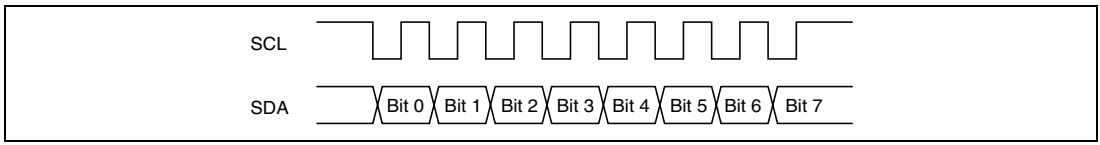


Figure 14.13 Clocked Synchronous Serial Transfer Format

Transmit Operation: In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 14.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Set the TRS bit in ICCR1 to select the transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

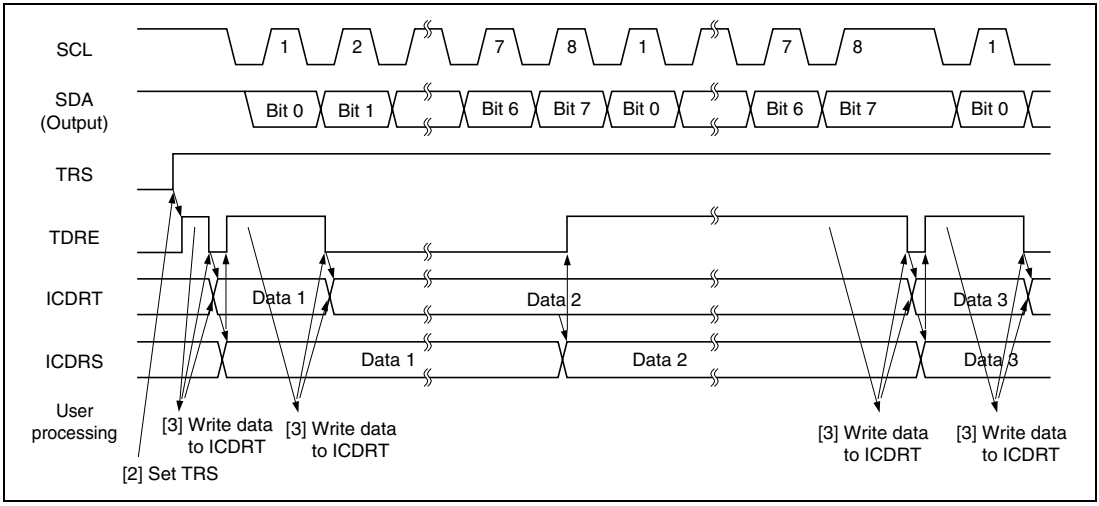


Figure 14.14 Transmit Mode Operation Timing

Receive Operation: In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 14.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

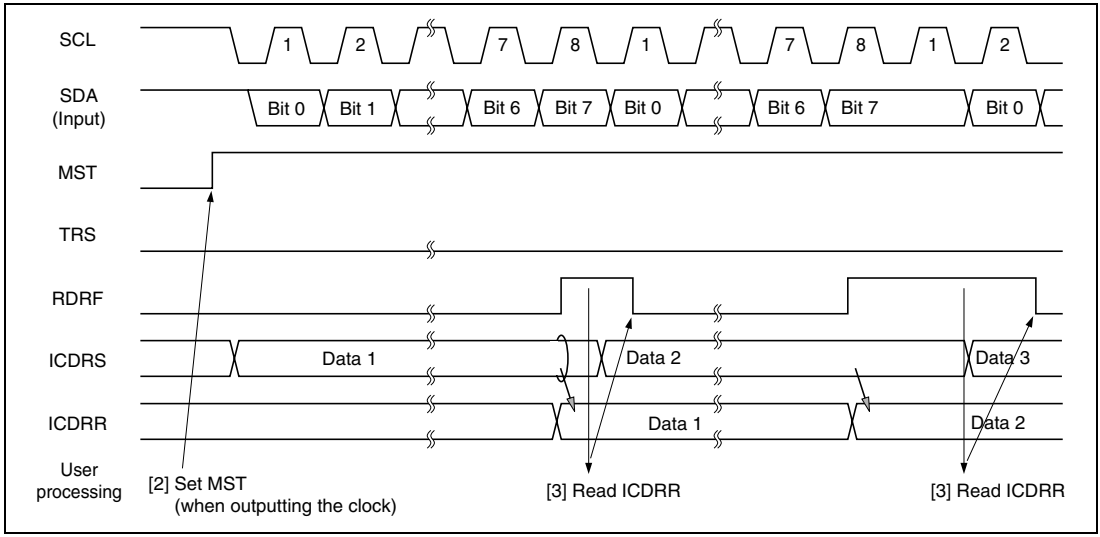


Figure 14.15 Receive Mode Operation Timing

14.4.7 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 14.16 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

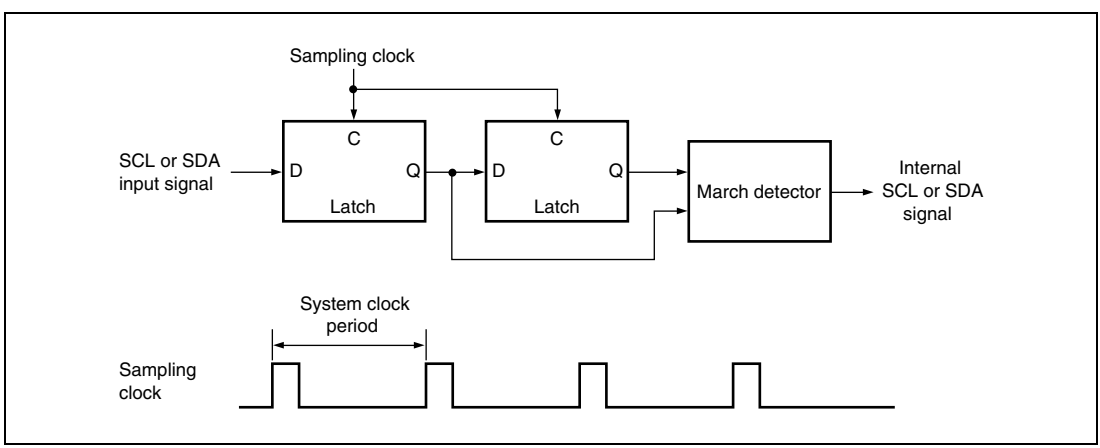


Figure 14.16 Block Diagram of Noise Canceler

14.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface are shown in figures 14.17 to 14.20.

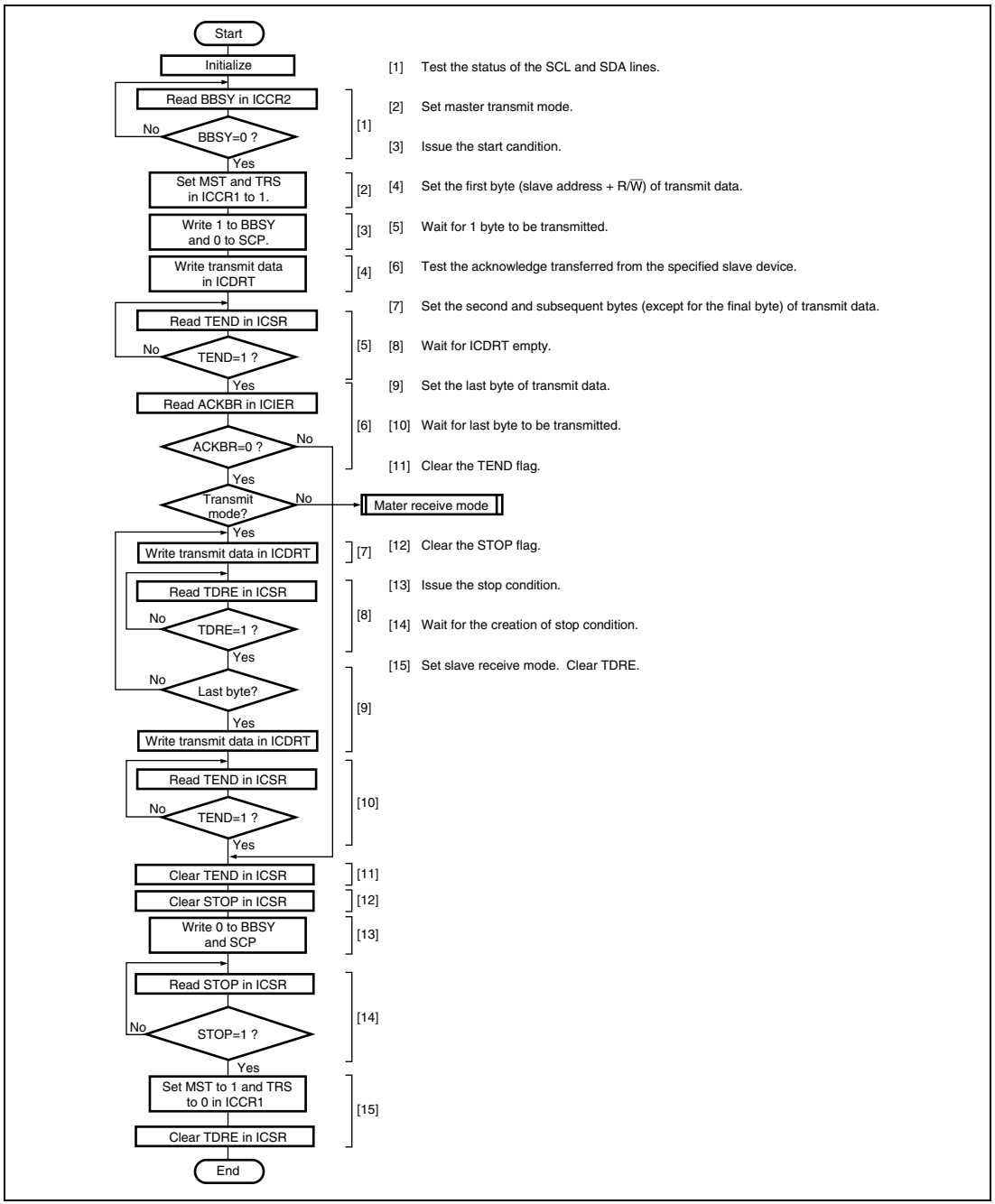


Figure 14.17 Sample Flowchart for Master Transmit Mode

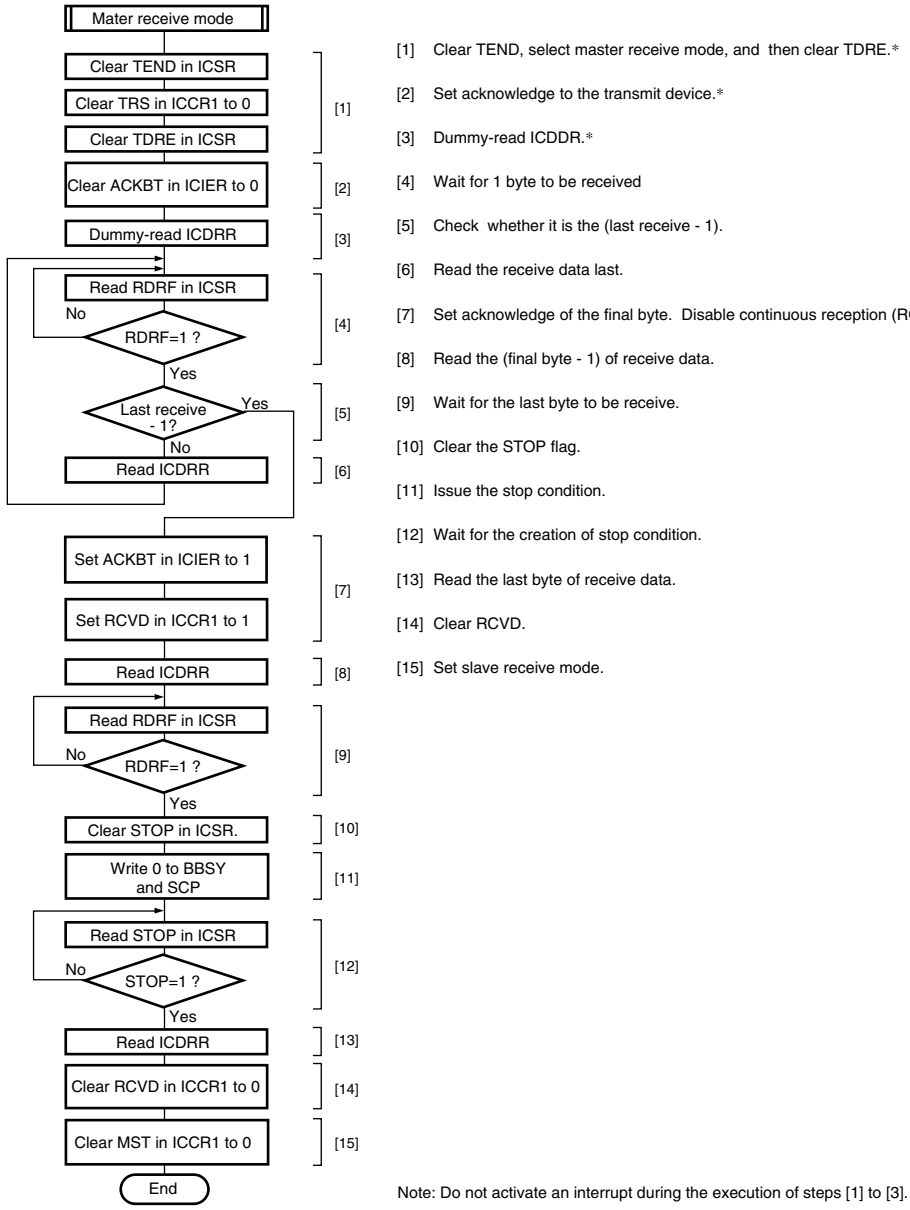


Figure 14.18 Sample Flowchart for Master Receive Mode

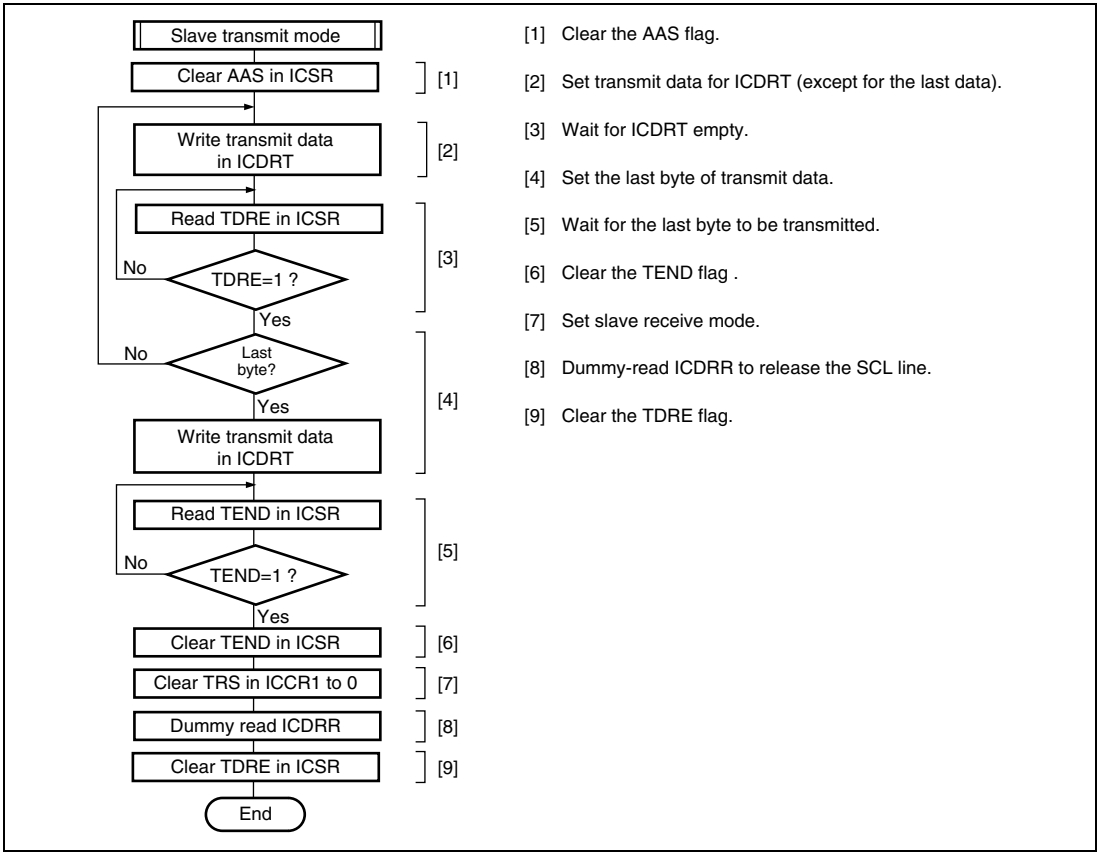


Figure 14.19 Sample Flowchart for Slave Transmit Mode

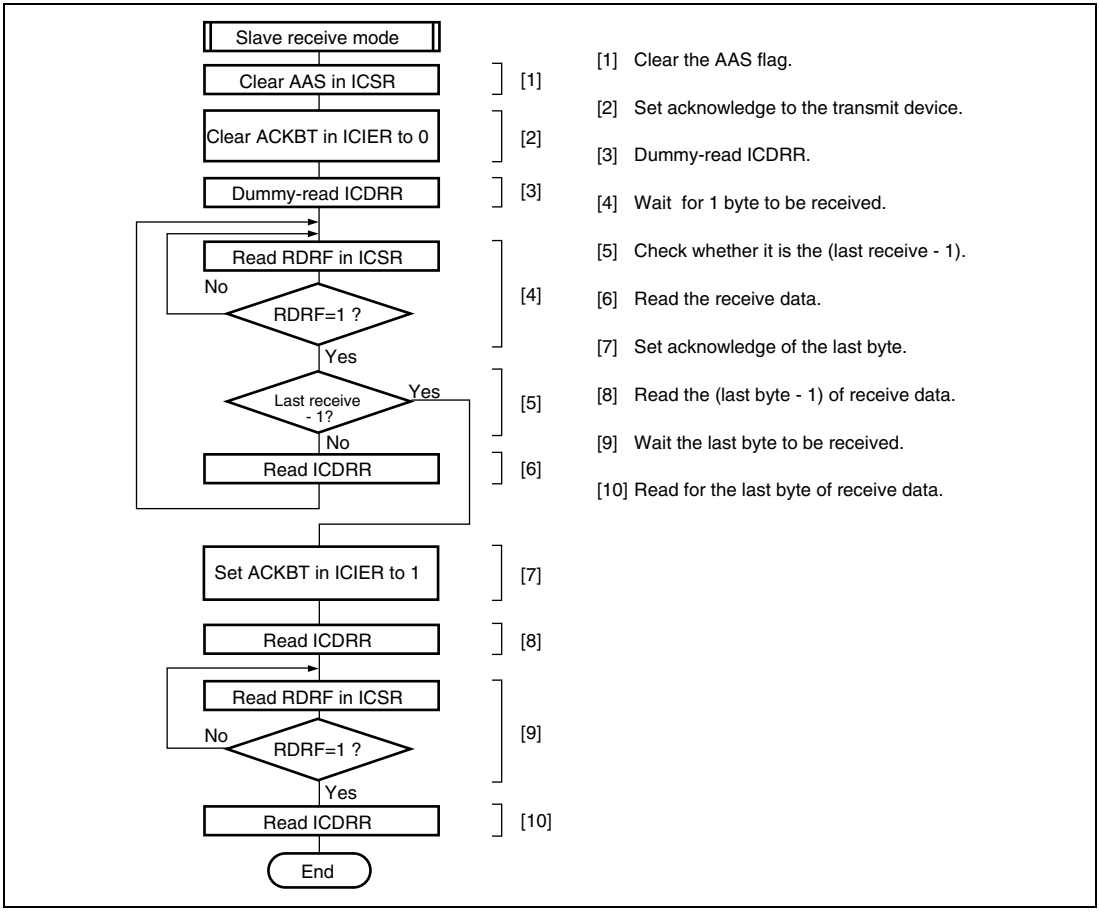


Figure 14.20 Sample Flowchart for Slave Receive Mode

14.5 Interrupt Request

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun error. Table 14.3 shows the contents of each interrupt request.

Table 14.3 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I^2C Mode	Clocked Synchronous Mode
Transmit data empty	TXI	(TDRE=1) • (TIE=1)	O	O
Transmit end	TEI	(TEND=1) • (TEIE=1)	O	O
Receive data full	RXI	(RDRF=1) • (RIE=1)	O	O
STOP recognition	STPI	(STOP=1) • (STIE=1)	O	X
NACK receive	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)	O	X
Arbitration lost/overrun Error			O	O

When interrupt conditions described in table 14.3 are 1 and the I bit in CCR is 0, the CPU executes an interrupt exception processing. Interrupt sources should be cleared in the exception processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data to ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 again at the same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an excessive data of one byte may be transmitted.

14.6 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 14.21 shows the timing of the bit synchronous circuit and table 14.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

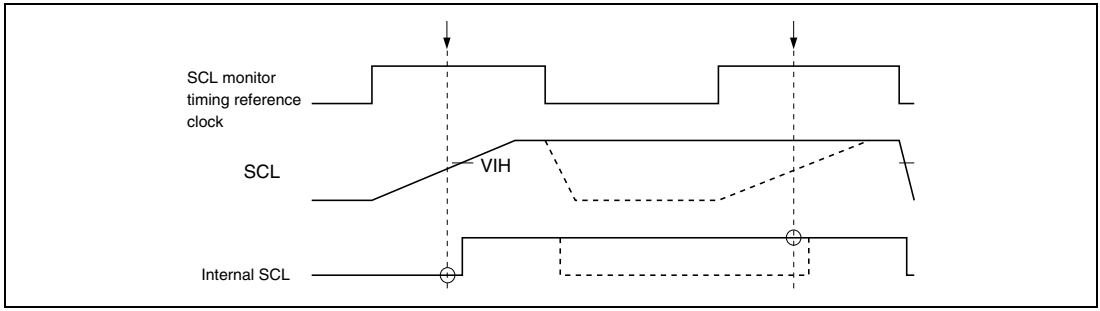


Figure 14.21 The Timing of the Bit Synchronous Circuit

Table 14.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	14.5 tcyc
	1	41.5 tcyc

Section 15 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to 16 analog input channels to be selected. A block diagram of the A/D converter is shown in figure 15.1.

15.1 Features

- 10-bit resolution
- 16 input channels
- Conversion time: 13.0 µs per channel (at 20-MHz operation), 10.0 µs per channel (at 26-MHz operation)
- Two operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three methods conversion start
  Software
  16-bit timer pulse unit (TPU or TMR) conversion start trigger
  External trigger signal
- Interrupt request
  An A/D conversion end interrupt request (ADI) can be generated
- Module stop mode can be set

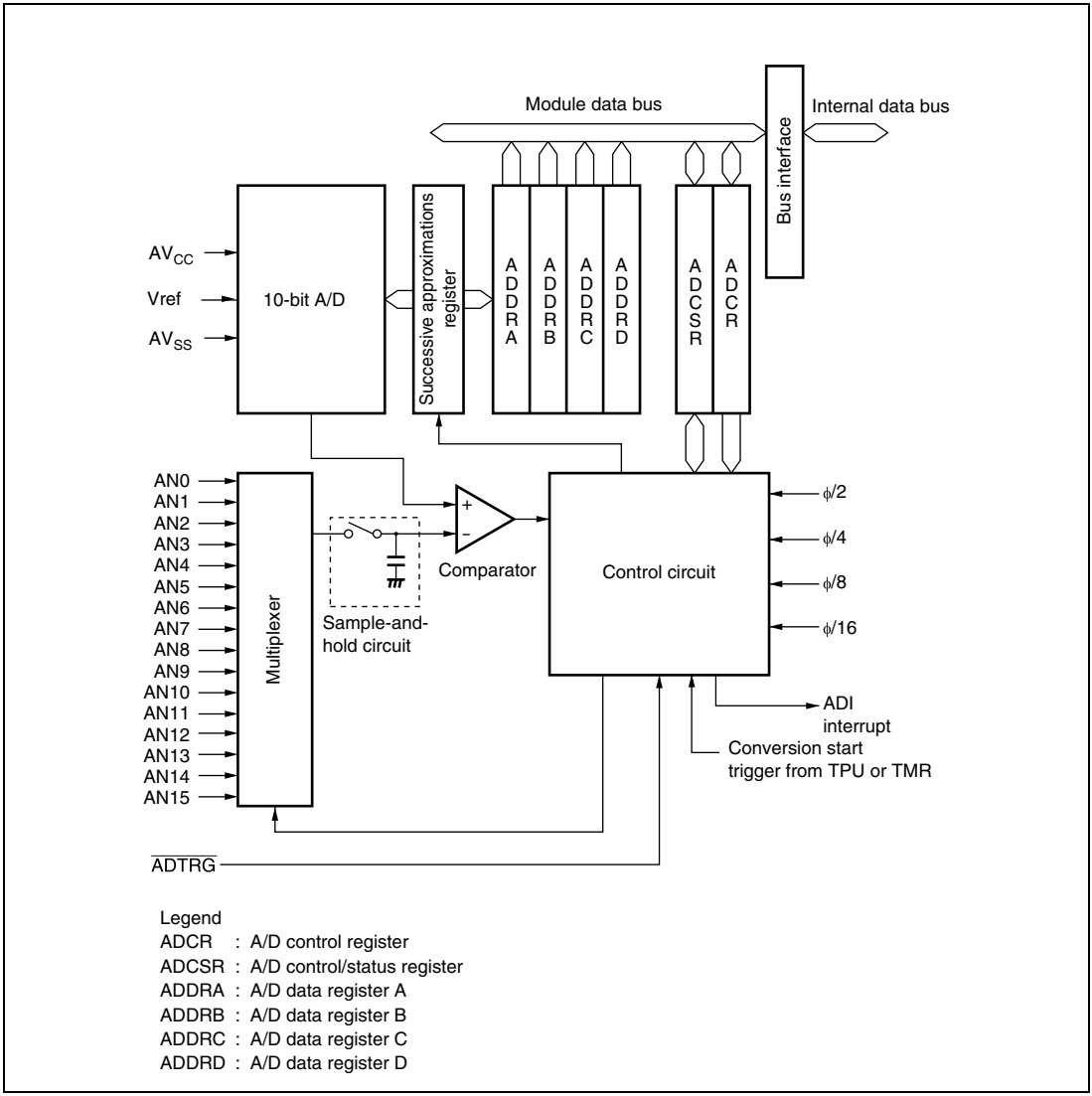


Figure 15.1 Block Diagram of A/D Converter

15.2 Input/Output Pins

Table 15.1 summarizes the input pins used by the A/D converter. The 16 analog input pins (AN0 to AN15) are divided into four groups each of which consists of two channels; analog input pins 0 to 7 (AN0 to AN7) comprising channel set 0, analog input pins 8 to 15 (AN8 to AN15) comprising channel set 1, analog input pins 0 to 3, 8 to 11 (AN0 to AN3, AN8 to AN11) comprising group 0, and analog input pins 4 to 7, 12 to 15 (AN4 to AN7, AN12 to AN15) comprising group 1. The AVcc and AVss pins are the power supply pins for the analog block in the A/D converter. The Vref pin is the A/D conversion reference voltage pin.

Table 15.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply pin
Analog ground pin	AV_{SS}	Input	Analog block ground and reference voltage
Reference voltage pin	Vref	Input	Reference voltage for A/D conversion
Analog input pin 0	AN0	Input	Channel set 0 (CH3 = 0), group 0 analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Channel set 0 (CH3 = 0), group 1 analog input pins
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Channel set 1 (CH3 = 1), group 0 analog input pins
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	Channel set 1 (CH3 = 1), group 1 analog input pins
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input	$\overline{\text{ADTRG}}$	Input	External trigger input pin for starting A/D conversion

15.3 Register Descriptions

The A/D converter has the following registers. For details on the module stop control register, refer to section 22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

15.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each analog input channel, are shown in table 15.2.

The converted 10-bit data is stored in bits 6 to 15. The lower 6 bits are always read as 0.

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. Therefore, when reading ADDR, read only the upper byte, or read in word unit.

Table 15.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
Channel Set 0 (CH3 = 0)		Channel Set 1 (CH3 = 1)		A/D Data Register to be Stored Results of A/D Conversion
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	
AN0	AN4	AN8	AN12	ADDRA
AN1	AN5	AN9	AN13	ADDRB
AN2	AN6	AN10	AN14	ADDRC
AN3	AN7	AN11	AN15	ADDRD

15.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified channels in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DTC is activated by an ADI interrupt and ADDR is read
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enabled when 1 is set
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and the A/D converter enters the wait state. Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when conversion on the specified channel is complete. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or a transition to software standby mode, hardware standby mode, or module stop mode. The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D external trigger input pin ($\overline{\text{ADTRG}}$).
4	SCAN	0	R/W	Scan Mode Selects single mode or scan mode as the A/D conversion operating mode. Only set the SCAN bit while conversion is stopped (ADST = 0). 0: Single mode 1: Scan mode

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel Select 3 Switches the analog channel allocated to group 0 and group 1. Group 0 / Group 1 0: Channel set 0 — AN0 to AN3 / AN4 to AN7 1: Channel set 1 — AN 8 to AN11 / AN12 to AN15
2	CH2	0	R/W	Channel Select 0 to 2
1	CH1	0	R/W	Select analog input channels.
0	CH0	0	R/W	When SCAN = 0 / When SCAN = 1 Channel set 0 (CH3 = 0) 000: AN0 / 000: AN0 001: AN1 / 001: AN0, AN1 010: AN2 / 010: AN0 to AN2 011: AN3 / 011: AN0 to AN3 100: AN4 / 100: AN4 101: AN5 / 101: AN4, AN5 110: AN6 / 110: AN4 to AN6 111: AN7 / 111: AN4 to AN7 Channel set 1 (CH3 = 1) 000: AN8 / 000: AN8 001: AN9 / 001: AN8, AN9 010: AN10 / 010: AN8 to AN10 011: AN11 / 011: AN8 to AN11 100: AN12 / 100: AN12 101: AN13 / 101: AN12, AN13 110: AN14 / 110: AN12 to AN14 111: AN15 / 111: AN12 to AN15

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

15.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 0 and 1
6	TRGS0	0	R/W	Enables the start of A/D conversion by a trigger signal. Only set bits TRGS0 and TRGS1 while conversion is stopped (ADST = 0). 00: A/D conversion start by software is enabled 01: A/D conversion start by TPU conversion start trigger is enabled 10: A/D conversion start by 8-bit timer conversion start trigger is enabled 11: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled
5	—	1	—	Reserved
4	—	1	—	These bits are always read as 1 and cannot be modified.
3	CKS1	0	R/W	Clock Select 0 and 1
2	CKS0	0	R/W	These bits specify the A/D conversion time. The conversion time should be changed only when the A/D conversion stops (ADST = 0). 00: Conversion time = 530 states (max.) 01: Conversion time = 266 states (max.) 10: Conversion time = 134 states (max.) 11: Conversion time = 68 states (max.)
1	—	1	—	Reserved
0	—	1	—	These bits are always read as 1 and cannot be modified.

15.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes; single mode and scan mode. When changing the operating mode or analog input channel, in order to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

15.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. The operations are as follows.

1. A/D conversion is started when the ADST bit is set to 1, according to software, timer conversion start trigger, or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

15.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels (four channels maximum). The operations are as follows.

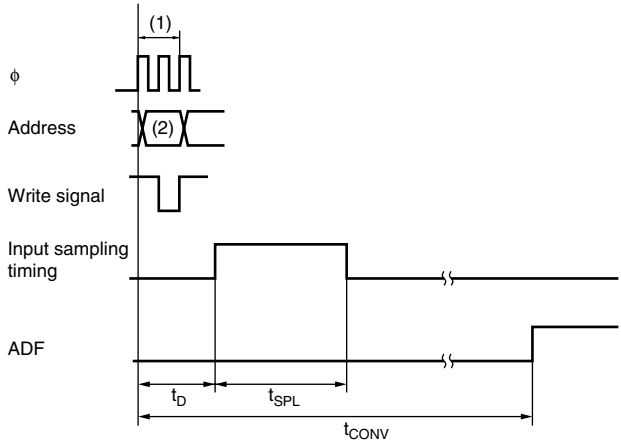
1. When the ADST bit is set to 1 by software, timer conversion start trigger, or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH3 = 0 and CH2 = 0, AN4 when CH3 = 0 and CH2 = 1, AN8 when CH3 = 1 and CH2 = 0, or AN12 when CH3 = 1 and CH2 = 1).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends. Conversion of the first channel in the group starts again.
4. Steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters the wait state. After that, when the ADST bit is set to 1, conversion of the first channel in the group starts again.

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 when the A/D conversion start delay time (t_D) has passed after the ADST bit is set to 1, then starts conversion. Figure 15.2 shows the A/D conversion timing. Table 15.3 shows the A/D conversion time.

As indicated in figure 15.2, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time (t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 15.3.

In scan mode, the values given in table 15.3 apply to the first conversion time. The values given in table 15.4 apply to the second and subsequent conversions.



Legend
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 15.2 A/D Conversion Timing

Table 15.3 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	18	—	33	10	—	17	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67	—	68

Note: All values represent the number of states.

Table 15.4 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

15.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS0 and TRGS1 bits are set to 1 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ 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 when the bit ADST has been set to 1 by software. Figure 15.3 shows the timing.

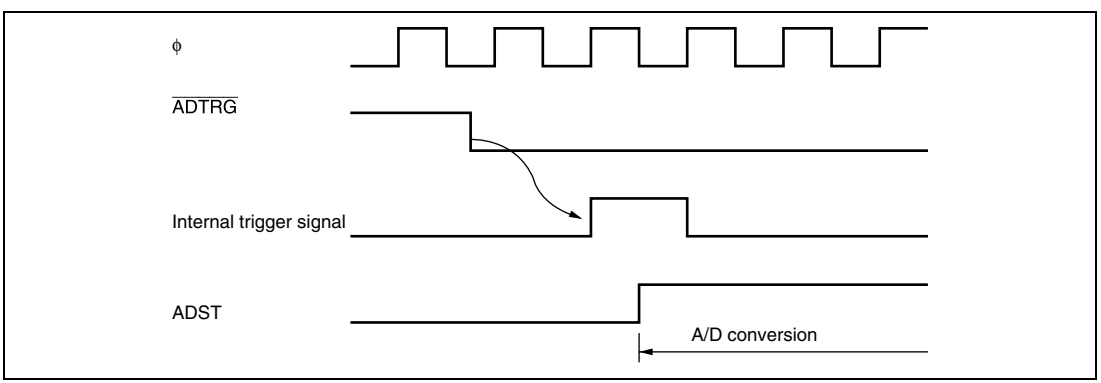


Figure 15.3 External Trigger Input Timing

15.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. Setting the ADIE bit to 1 enables ADI interrupt requests while the ADF bit in ADCSR is set to 1 after A/D conversion is completed. The data transfer controller (DTC) can be activated by an ADI interrupt. Having the converted data read by the DTC in response to an ADI interrupt enables continuous conversion without imposing a load on software.

Table 15.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
ADI	A/D conversion end	ADF	Possible

15.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 15.4).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 15.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 15.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 15.5).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.

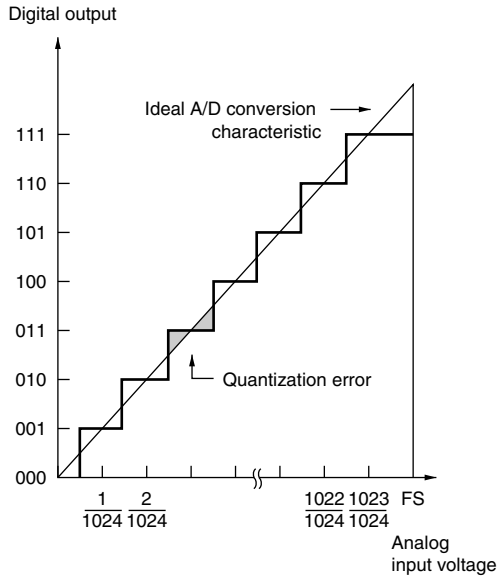


Figure 15.4 A/D Conversion Accuracy Definitions

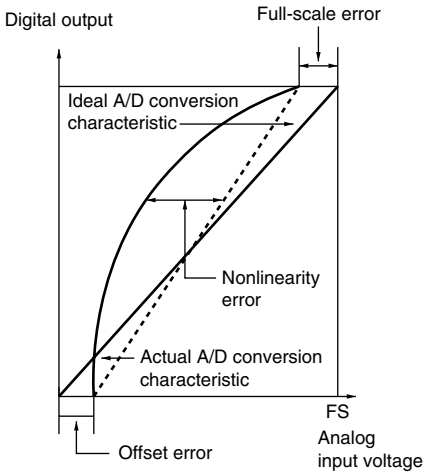


Figure 15.5 A/D Conversion Accuracy Definitions

15.7 Usage Notes

15.7.1 Module Stop Mode Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

15.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion accuracy. However, for A/D conversion in single mode with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater) (see figure 15.6). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

15.7.3 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute accuracy. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

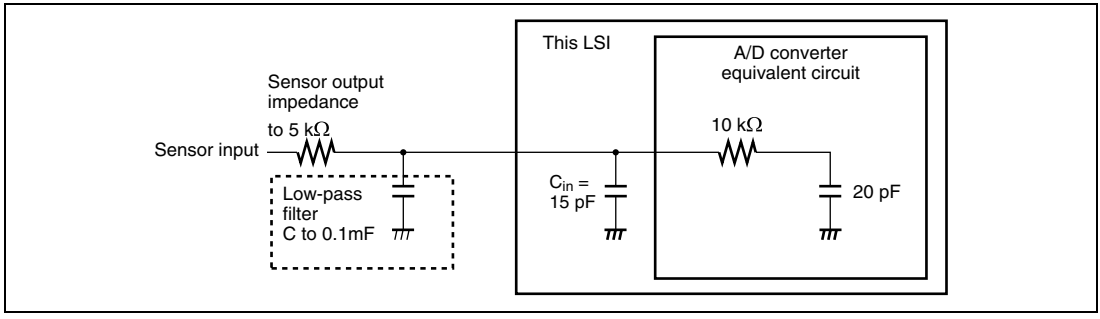


Figure 15.6 Example of Analog Input Circuit

15.7.4 Range of Analog Power Supply and Other Pin Settings

If the conditions below are not met, the reliability of the device may be adversely affected.

- Analog input voltage range

  The voltage applied to analog input pin ANn during A/D conversion should be in the range AVss ≤ ANn ≤ AVcc.
- Relationship between AVcc, AVss and Vcc, Vss

  Set AVss = Vss as the relationship between AVcc, AVss and Vcc, Vss. If the A/D converter is not used, the AVcc and AVss pins must not be left open.
- Vref range

  The reference voltage input from the Vref pin should be set to AVcc or less.

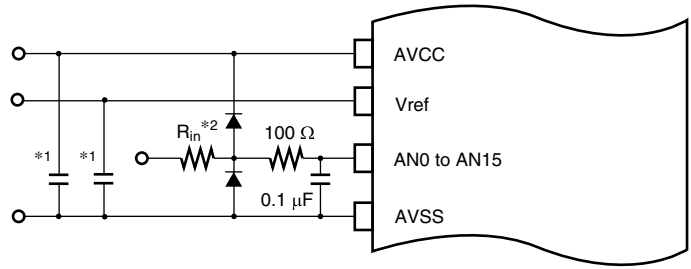
15.7.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values. Also, digital circuitry must be isolated from the analog input signals (AN0 to AN15), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (Vss) on the board.

15.7.6 Notes on Noise Countermeasures

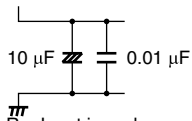
A protection circuit should be connected in order to prevent damage due to abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN15), between AVcc and AVss, as shown in figure 15.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding circuit constants.



Notes: Values are reference values.

1.

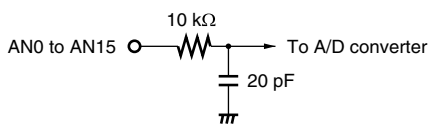


2. R_{in}: Input impedance

Figure 15.7 Example of Analog Input Protection Circuit

Table 15.6 Analog Pin Specifications

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ



Note: Values are reference values.

Figure 15.8 Analog Input Pin Equivalent Circuit

16.1 Features

- 8-bit resolution
- Two output channels
- Conversion time: 10 µs, maximum (when load capacitance is 20 pF)
- Output voltage: 0 V to Vref
- D/A output retaining function in software standby mode
- Module stop mode can be set

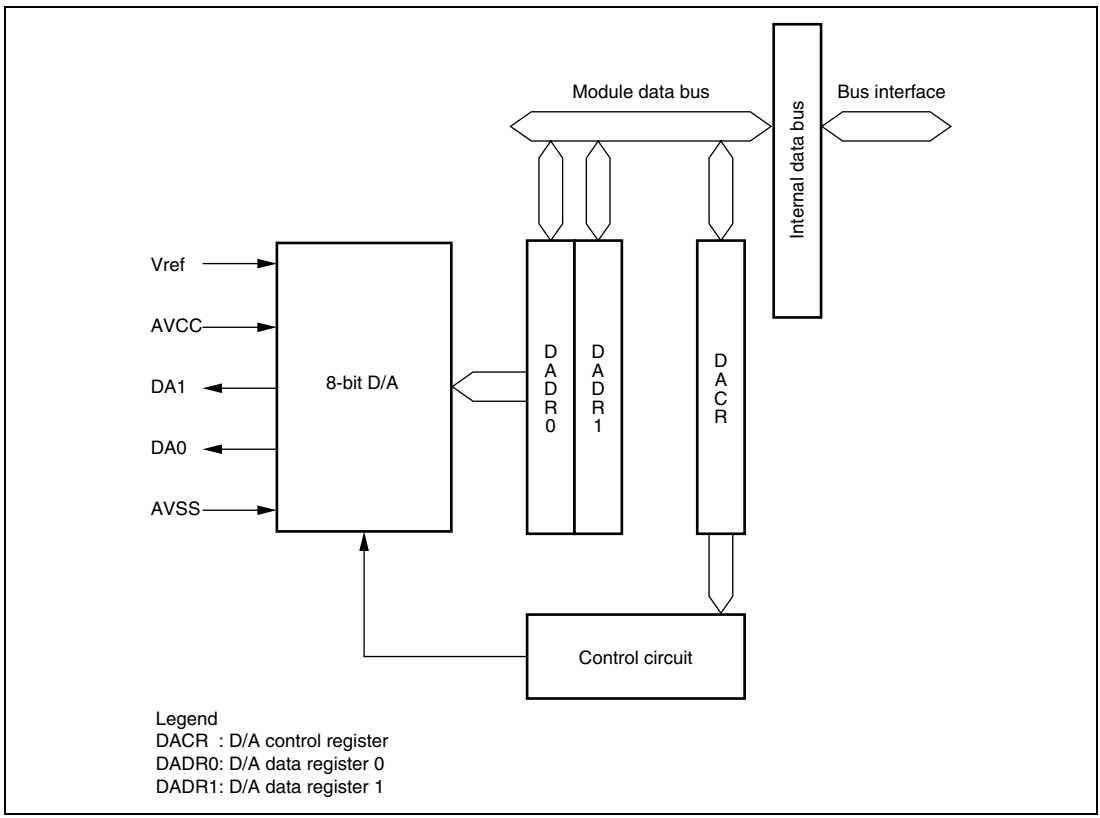


Figure 16.1 Block Diagram of D/A Converter

16.2 Input/Output Pins

Table 16.1 shows the pin configuration for the D/A converter.

Table 16.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground and reference voltage
Analog output pin 0	DA0	Output	Channel 0 analog output pin
Analog output pin 1	DA1	Output	Channel 1 analog output pin
Reference voltage pin	Vref	Input	Analog block reference voltage

16.3 Register Descriptions

The D/A converter has the following registers. For details on the module stop control register, refer to section 22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register (DACR)

16.3.1 D/A Data Registers 0, 1 (DADR0, DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data for D/A conversion. When analog output is permitted, D/A data register contents are converted and output to analog output pins.

16.3.2 D/A Control Register (DACR)

DACR controls D/A converter operation.

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output 0: Analog output DA1 is disabled 1: D/A conversion for channel 1 and analog output DA1 are enabled
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output 0: Analog output DA0 is disabled 1: D/A conversion for channel 0 and analog output DA0 are enabled
5	DAE	0	R/W	D/A Enable Controls D/A conversion in conjunction with the DAOE0 and DAOE1 bits. When the DAE bit is cleared to 0, D/A conversion for channels 0 and 1 are controlled individually. When DAE is set to 1, D/A conversion for channels 0 and 1 are controlled as one. Conversion result output is controlled by the DAOE0 and DAOE1 bits. For details, see table 16.2.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 16.2 D/A Conversion Control

Bit 5	Bit 7	Bit 6	
DAE	**DAOE1**	**DAOE0**	**Description**
0	0	0	Disables D/A Conversion
		1	Enables D/A Conversion for channel 0
	1	0	Enables D/A Conversion for channel 1
		1	Enables D/A Conversion for channels 0 and 1
1	0	0	Disables D/A Conversion
		1	Enables D/A Conversion for channels 0 and 1
	1	0	
		1	

16.4 Operation

Two channels of the D/A converter can perform conversion individually.

When the DAOE bit in DACR is set to 1, D/A conversion is enabled and the conversion results are output.

An example of D/A conversion of channel 0 is shown below. The operation timing is shown in figure 16.2.

1. Write conversion data to DADR0.
2. When the DAOE0 bit in DACR is set to 1, D/A conversion starts. After the interval of t_{DCONV}, the conversion results are output from the analog output pin DA0. The conversion results are output continuously until DADR0 is modified or DAOE0 bit is cleared to 0. The output value is calculated by the following formula:

   (DADR contents)/256 x Vref
3. Conversion starts immediately after DADR0 is modified. After the interval of t_{DCONV}, conversion results are output.
4. When the DAOE bit is cleared to 0, analog output is disabled.

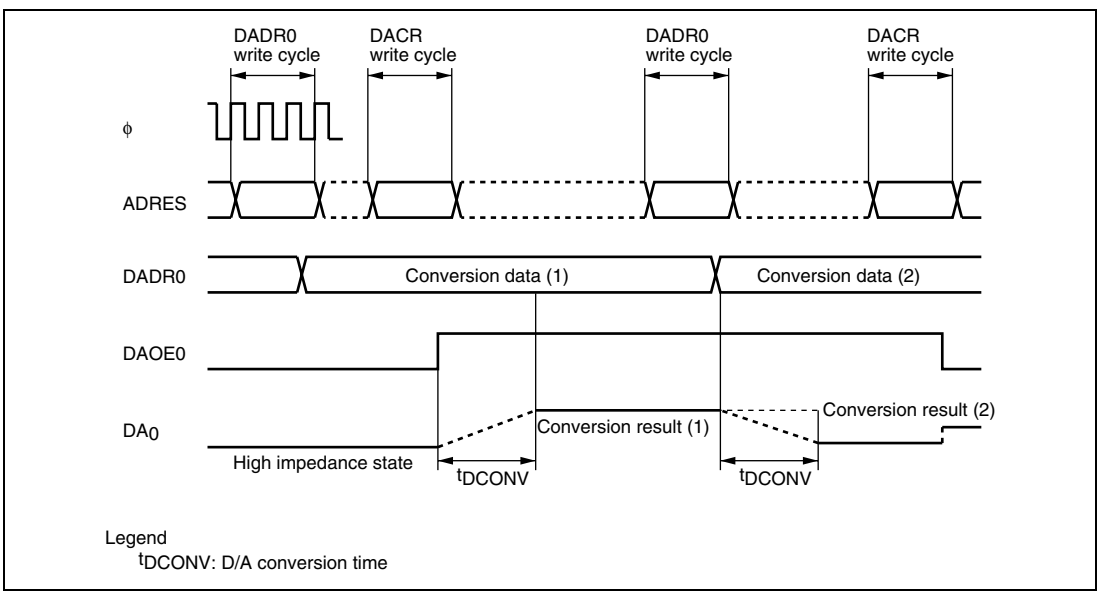


Figure 16.2 D/A Converter Operation Example

16.5 Usage Notes

16.5.1 Analog Power Supply Current in Software Standby Mode

If this LSI enters software standby mode while D/A conversion is enabled, D/A output is retained and the analog power supply current is equivalent to that during D/A conversion. To reduce analog power supply current in software standby mode, clear the DAOE0, DAOE1 and DAE bits to 0 to disable D/A output.

16.5.2 Setting for Module Stop Mode

It is possible to enable/disable the D/A converter operation using the module stop control register, the D/A converter does not operate by the initial value of the register. The register can be accessed by releasing the module stop mode. For more details, see section 22, Power-Down Modes.

Section 17 IEBus™ Controller (IEB) [H8S/2552 Series]

This LSI has an on-chip one-channel IEBus controller (IEB). The Inter Equipment Bus™ (IEBus™)*[1] is a small-scaled digital data transfer system for inter equipment data transfer.

This LSI does not have an on-chip IEBus driver/receiver, so it is necessary to mount a dedicated driver/receiver*[2] externally.

Notes: 1. IEBus is a trademark of NEC Electronics Corporation.
2. Bus interface driver/receiver IC: HA12187FP is recommended.

17.1 Features

- IEBus protocol control (layer 2) supported
  - Half duplex asynchronous communications
  - Multi-master system
  - Broadcast communications function
  - Selectable mode (three types) with different transfer speeds
- Data transfer by the data transfer controller (DTC)
  - Transfer buffer: 1 byte
  - Reception buffer: 1 byte
  - Up to 128 bytes of consecutive transfer/reception (maximum number of transfer bytes in mode 2)
- Operating frequency
  - 12 MHz, 12.58 MHz (IEB uses 1/2 divided external clock.)
  - 18 MHz, 18.87 MHz (IEB uses 1/3 divided external clock.)
  - 24 MHz, 25.16 MHz (IEB uses 1/4 divided external clock.)
- Noise resistance is improved by mounting the IEBus driver/receiver (layer 1) externally.

Figure 17.1 shows an IEB block diagram.

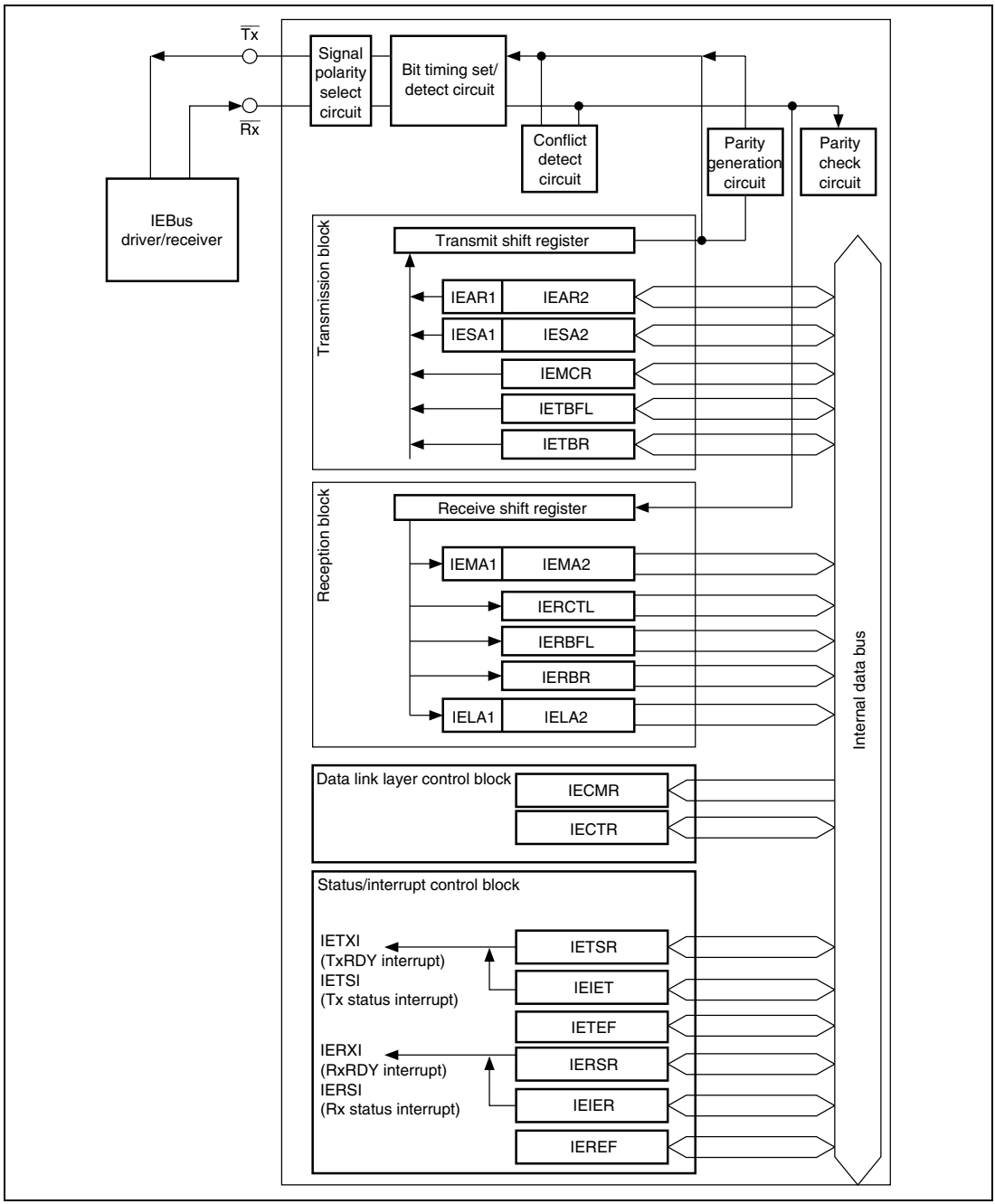


Figure 17.1 Block Diagram of IEB

17.1.1 IEBus Communications Protocol

The overview of the IEBus is described below.

- Communications method: Half duplex asynchronous communications
- Multi-master system
  All units connected to the IEBus can transfer data to other units.
- Broadcast communications function (one-to-many communications)
  - — Group broadcast communications: Broadcast communications to group unit
  - — General broadcast communications: Broadcast communications to all units
- Mode is selectable (three modes with different transfer speeds).

Table 17.1 Mode Types

Mode	ϕ = 12, 18, 24 MHz	ϕ = 12.58, 18.87, 25.16 MHz	Maximum Number Of Transfer Bytes (byte/frame)
0	About 3.9 kbps	About 4.1 kbps	16
1	About 17 kbps	About 18 kbps	32
2	About 26 kbps	About 27 kbps	128

- Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
  Priority of bus mastership is as follows.
  - — Broadcast communications (one-to-many communications) have priority rather than normal communications (one-to-one communications).
  - — Smaller master address has priority.
- Communications scale
  - — Number of units: Up to 50
  - — Cable length: Up to 150 m (when using a twisted pair cable)

Note: The communications scale of the actual system depends on the externally mounted IEBus driver/receiver characteristics and the characteristics of the cable to be used.

(1) Determination of Bus Mastership (Arbitration)

A unit connected to the IEBus performs an operation for getting the bus to control other units. This operation is called arbitration. In arbitration, when the multiple units start transfer simultaneously, the bus mastership is given to one unit among them.

Only one unit can get bus mastership through arbitration, so the following priority for bus mastership is defined.

(a) Priority according to communications type

Broadcast communications (one-to-many communications) has priority over normal communications (one-to-one communications).

(b) Priority according to master address

A unit with the smallest master address has priority among units with the same communications type.

Example: The master address is configured with 12 bits. A unit with H′0000 has the highest priority, and a unit with H′FFFF has the lowest priority.

Note: When a unit loses arbitration, the unit can automatically enter retransfer mode (0 to 7 retransfer times can be selected by bits RN2 to RN0 in IEMCR).

(2) Communications Mode

The IEBus has three communications modes with different transfer speeds. Table 17.2 shows the transfer speed in each communications mode and the maximum number of transfer bytes in one communications frame.

Table 17.2 Transfer speed and Maximum Number of Transfer Bytes in Each Communications Mode

Communications Mode	Maximum Number of Transfer Bytes (byte/frame)	Effective Transfer Speed*[1] (kbps)	
		ϕ = 12, 18, 24 MHz*[2]	ϕ = 12.58, 18.87, 25.16 MHz*[2]
0	16	About 3.9	About 4.1
1	32	About 17	About 18
2	128	About 26	About 27

Notes: Each unit connected to the IEBus should select a communications mode prior to performing communications. Note that correct communications is not guaranteed if the master and slave units do not adopt the same communications mode.

In the case of communications between a unit with ϕ = 12 MHz and a unit with ϕ = 12.58 MHz, correct communications is not possible even if the same communications mode is adopted. This is similar to the case of communications between a unit with ϕ = 24 MHz and a unit with ϕ = 25.16 MHz, or between a unit with ϕ = 18 MHz and a unit with ϕ = 18.87 MHz. Communications must be performed at the same oscillation frequency.

1. An effective transfer speed when the maximum number of transfer bytes is transmitted.
2. Oscillation frequency when this LSI is used

(3) Communications Address

In the IEBus, a 12-bit specific communications addresses are allocated to individual units. A communications address is configured as follows.

- Upper four bits: group number (number identifying a group to which the unit belongs)
- Lower eight bits: unit number (number identifying individual units in a group)

(4) Broadcast Communications

In normal transfer, a single master unit communicates with a single slave unit. So, one-to-one transfer or reception is performed. In broadcast communications, a single master unit communicates with multiple slave units. Since there are multiple slave units, acknowledgement is not returned from the slave units during communications.

A broadcast bit decides whether broadcast or normal communications is performed. (For details of the broadcast bit, refer to section 17.1.2 (1) (b), Broadcast Bit.

There are two types of broadcast communications.

(a) Group broadcast communications

Broadcast communications is performed to units with the same group number, meaning that those units have the same upper four bits of the communications address.

(b) General broadcast communications

Broadcast communications is performed to all units regardless of the group number.

Group broadcast and general broadcast communications are identified by a slave address. (For details on the slave address, refer to section 17.1.2 (3), Slave Address Field.)

17.1.2 Communications Protocol

Figure 17.2 shows an IEBus transfer signal format.

Communications data is transferred as a series of signals referred to as a communications frame. The number of data which can be transmitted in a single communications frame and the transfer speed differ according to communications mode.

(When ϕ = 12, 18, or 24MHz)

Field name	Header		Master address field		Slave address field			Control field			Message length field			Data field								
Number of bits	1	1	12	1	12	1	1	4	1	1	8	1	1	8	1	1		8	1	1
	Start bit	Broad-cast bit	Master address	P	Slave address	P	A	Control bits	P	A	Message length bits	P	A	Data bits	P	A		Data bits	P	A
Transfer time																				
Mode 0	Approximately 7330 µs													Approximately 1590 × N µs						
Mode 1	Approximately 2090 µs													Approximately 410 × N µs						
Mode 2	Approximately 1590 µs													Approximately 300 × N µs						

P: Parity bit (1 bit)
A: Acknowledge bit (1 bit)
When A = 0: ACK
When A = 1: NAK
N: Number of bytes

Note: The value of acknowledge bit is ignored in broadcast communications.

Figure 17.2 Transfer Signal Format

(1) Header

Header is comprised of a start bit and a broadcast bit.

(a) Start Bit

The start bit is a signal for informing a start of data transfer to other units. A unit, which attempts to start data transfer, outputs a low-level signal (start bit) for a specified period and then outputs the broadcast bit.

If another unit is already outputting a start bit when a unit attempts to output a start bit, the unit waits for completion of output of the start bit from the other unit without outputting the start bit, and then outputs the broadcast bit synchronized with the completion timing.

Other units enter the receive state after detecting the start bit.

(b) Broadcast Bit

The broadcast bit is a bit to identify the type of communications: broadcast or normal.

When this bit is cleared to 0, it indicates the broadcast communications. When it is set to 1, it indicates the normal communications. Broadcast communications includes group broadcast and general broadcast, which are identified by a value of the slave address. (For details of the slave address, refer to section 17.1.2 (3), Slave Address Field.)

Since there are multiple slave units, which are communications destination units, in the case of broadcast communications, the acknowledge bit is not returned from each field described in (2) and below.

When more than one unit starts transfer of communications frame at the same timing, broadcast communications has priority over normal communications, and arbitration occurs.

(2) Master Address Field

The master address field is a field for transmitting the unit address (master address) to other units. The master address field is comprised of master address bits and a parity bit.

The master address has 12 bits and are output MSB first.

When more than one unit starts transfer of the broadcast bit having the same value at the same timing, arbitration is decided by the master address field.

In the master address field, self-output data and data on the bus are compared for every one-bit transfer. If the self-output master address and data on the bus are different, the unit that loses arbitration, stops transfer, and enters the receive state.

Since the IEBus is configured with wired AND, a unit having the smallest master address of the units in arbitration (arbitration master) wins in arbitration.

Finally, only a single unit remains in the transfer state as a master unit after outputting 12-bit master address.

Next, this master unit outputs a parity bit*, defines the master address to other units, and then enters the slave address field output state.

Note: Since even parity is used, when the number of one bits in the master address is odd, the parity bit is 1.

(3) Slave Address Field

The slave address field is a field to transmit an address (slave address) of a unit (slave unit) to which a master transmit data. The slave address field is comprised of slave address bits, a parity bit, and an acknowledge bit.

The slave address has 12 bits and is output MSB first. The parity bit is output after the 12-bit slave address is transmitted in order to avoid receiving the slave address accidentally. The master unit then detects the acknowledgement from the slave unit in order to confirm that the slave unit exists on the bus. When the acknowledgement is detected, the master unit enters the control field output state. However, the master unit enters the control field output state without detecting the acknowledgement in broadcast communications.

The slave unit returns the acknowledgement when the slave addresses match and the parities of the master and slave addresses are correct. When either of the parities of the master and slave addresses is wrong, the slave unit decides that the master or slave address is not correctly received and does not return the acknowledgement. In this case, the master unit enters the waiting (monitor) state, and communications end.

In the case of broadcast communications, the slave address is used to identify the type of broadcast communications (group or general) as follows:

- When the slave address is H′FFFF: General broadcast communications
- When the slave address is other than H′FFFF: Group broadcast communications

Note: The group number is the upper 4-bit value of the slave address in group broadcast communications.

(4) Control Field

The control field is a field for transmitting the type and direction of the following data field. The control field is comprised of control bits, a parity bit, and an acknowledge bit.

The control bits include four bits and are output MSB first.

The parity bit is output following the control bits. When the parity is correct, and the slave unit can implement the function required from the master unit, the slave unit returns the acknowledgement and enters the message length field output state. However, if the slave unit cannot implement the requirements from the master unit even though the parity is correct, or if the parity is not correct, the slave unit does not return the acknowledgement, and returns to the waiting (monitor) state.

The master unit enters the subsequent message length field output state after confirming the acknowledgement.

When the acknowledgement is not confirmed, the master unit enters the waiting (monitor) state, and communications end. However, in the case of broadcast communications, the master unit enters the following message length field output state without confirming the acknowledgement. For details of the contents of the control bit, refer to table 17.4.

(5) Message Length Field

The message length field is a field for specifying the number of transfer bytes. The message length field is comprised of message length bits, a parity bit, and an acknowledge bit.

The message length has eight bits and is output MSB first. Table 17.3 shows the number of transfer bytes.

Table 17.3 Contents of Message Length bits

Message Length bits (Hexadecimal)	Number of Transfer Bytes
H′01	1 byte
H′02	2 bytes
.	.
.	.
H′FF	255 bytes
H′00	256 bytes

Note: * If a number greater than the maximum number of transfer bytes in one frame is specified, communications are performed in multiple frames depending on the communications mode. In this case, the message length bits indicate the number of remaining communications data after the first transfer. In this LSI, after the first transfer, the message length bits must be specified to the number of remaining communications data by a program, since these bits are not automatically specified by the hardware.

This field operation differs depending on the value of bit 3 in the control field: master transmission (bit 3 in the control bits is 1) or master reception (bit 3 in the control bits is 0).

(a) Master Transmission

The master unit outputs the message length bits and parity bit. When the parity is correct, the slave unit returns the acknowledgement and enters the following data field. Note that the slave unit does not return the acknowledgement in broadcast communications.

In addition, when the parity is not correct, the slave unit decides that the message length field is not correctly received, does not return the acknowledgement, and returns to the waiting (monitor) state. In this case, the master unit also returns to the waiting state, and communications end.

(b) Master Reception

The slave unit outputs the message length bits and parity bit. When the parity is correct, the master unit returns the acknowledgement.

When the parity is not correct, the master unit decides that the message length bits are not correctly received, does not return the acknowledgement, and returns to the waiting state. In this case, the slave unit also returns to the waiting state, and communications end.

(6) Data Field

The data field is a field for data transmission/reception to the slave unit. The master unit transmits/receives data to/from the slave unit using the data field. The data field is comprised of data bits, a parity bit, and an acknowledge bit.

The data bits include eight bits and are output MSB first.

The parity bit and acknowledge bit following the data bits are output from the master unit and slave unit, respectively.

Broadcast communications are performed only for the transmission of the master unit. In this case, the acknowledge bit is ignored. Operations in master transmission and master reception are described below.

(a) Master Transmission

The master unit transmits the data bits and parity bit to the slave unit to write data from the master unit to the slave unit. The slave unit receives the data bits and parity bit, and returns the acknowledgement if the parity bit is correct and the receive buffer is empty. If the parity bit is not correct or the receive buffer is not empty, the slave unit rejects acceptance of corresponding data and does not return the acknowledgement.

When the slave unit does not return the acknowledgement, the master unit retransmits the same data. This operation is repeated until either the acknowledgement from the slave unit is detected or the maximum number of data transfer bytes is exceeded.

When the parity is correct and the acknowledgement is output from the slave unit, the master unit transmits the subsequent data if data remains and the maximum number of transfer bytes is not exceeded.

In the case of broadcast communications, the slave unit does not return the acknowledgement, and the master unit transfers data byte by byte.

(b) Master Reception

The master unit outputs synchronous signals corresponding to all data bits to be read from the slave unit.

The slave unit outputs the data bits and parity bit on the bus in accordance with the synchronous signals from the master unit.

The master unit reads the parity bit output from the slave unit, and checks the parity. If the parity is not correct, or the receive buffer is not empty, the master unit rejects acceptance of the data, and does not return the acknowledgement. The master unit reads the same data repeatedly if the number of data does not exceed the maximum number of transfer bytes in one frame. If the parity is correct and the receive buffer is empty, the master unit accepts data and returns the acknowledgement. The master unit reads in the subsequent data if the number of data does not exceed the maximum number of transfer bytes in one frame.

(7) Parity bit

The parity bit is used to confirm that transfer data has no error.

The parity bit is added to respective data of the master address, slave address, control, message length, and data bits.

The even parity is used. When the number of one bits in data is odd, the parity bit is 1. When the number of one bits in data is even, the parity bit is 0.

(8) Acknowledge bit

In normal communications (a single unit to a single unit communications), the acknowledge bit is added to the following position in order to confirm that data is correctly accepted.

- At the end of the slave address field
- At the end of the control field
- At the end of the message length field
- At the end of the data field

The acknowledge bit is defined below.

- 0: indicates that the transfer data is acknowledged. (ACK)
- 1: indicates that the transfer data is not acknowledged. (NAK)

Note that the acknowledge bit is ignored in the case of broadcast communications.

(a) Acknowledge bit at the End of the Slave Address Field

The acknowledge bit at the end of the slave address field becomes NAK in the following cases and transfer is stopped.

— When the parity of the master address or slave address bits is incorrect
— When a timing error (an error in bit format) occurs
— When there is no slave unit

(b) Acknowledge bit at the End of the Control Field

The acknowledge bit at the end of the control field becomes NAK in the following cases and transfer is stopped.

— When the parity of the control bits is incorrect
— When bit 3 in the control bits is 1 (data write) although the slave receive buffer* is not empty
— When the control bits are set to the data read (H′3, H′7) although the slave transmit buffer* is empty
— When another unit which locked the slave unit requests H′3, H′6, H′7, H′A, H′B, H′E, or H′F in the control bits although the slave unit has been locked
— When the control bits are the locked address read (H′4, H′5) although the unit is not locked
— When a timing error occurs
— When the control bits are undefined

Note: See section 17.1.3 (1), Slave Status Read (Control Bits: H′0, H′6).

(c) Acknowledge Bit at the End of the Message Length Field

The acknowledge bit at the end of the message length field becomes NAK in the following cases and transfer is stopped.

— When the parity of the message length bits is incorrect

— When a timing error occurs

(d) Acknowledge Bit at the End of the Data Field

The acknowledge bit at the end of the data field becomes NAK in the following cases and transfer is stopped.

— When the parity of the data bits is incorrect*

— When a timing error occurs after the previous transfer of the acknowledge bit

— When the receive buffer becomes full and cannot accept further data

Note: In this case, data field is transferred repeatedly until the number of data reaches the maximum number of transfer bytes if the number of data does not exceed the maximum number of transfer bytes in one frame.

17.1.3 Transfer Data (Data Field Contents)

The data filed contents are specified by the control bits.

Table 17.4 Control Bit Contents

Setting Value	Bit 3*[1]	Bit 2	Bit 1	Bit 0	Function*[2]
H′0	0	0	0	0	Reads slave status (SSR)
H′1	0	0	0	1	Undefined
H′2	0	0	1	0	Undefined
H′3	0	0	1	1	Reads data and locks
H′4	0	1	0	0	Reads locked address (lower 8 bits)
H′5	0	1	0	1	Reads locked address (upper 4 bits)
H′6	0	1	1	0	Reads slave status (SSR) and unlocks
H′7	0	1	1	1	Reads data
H′8	1	0	0	0	Undefined
H′9	1	0	0	1	Undefined
H′A	1	0	1	0	Writes command and locks
H′B	1	0	1	1	Writes data and locks
H′C	1	1	0	0	Undefined
H′D	1	1	0	1	Undefined
H′E	1	1	1	0	Writes command
H′F	1	1	1	1	Writes data

Notes: 1. According to the value of bit 3 (MSB), the transfer directions of the message length bits in the following message length field and data in the data field vary.
When bit 3 is 1: Data is transferred from the master unit to the slave unit.
When bit 3 is 0: Data is transferred from the slave unit to the master unit.
2. H′3, H′6, H′A, and H′B are control bits to specify lock setting and cancellation.
When the undefined values of H′1, H′2, H′8, H′9, H′C, and H′D are transmitted, the acknowledge bit is not returned.

When the control bits received from another unit which locked are not included in table 17.5, the slave unit which has been locked by the master unit rejects acceptance of the control bits and does not return the acknowledge bit.

Table 17.5 Control Field for Locked Slave Unit

Setting Value	Bit 3	Bit 2	Bit 1	Bit 0	Function
H′0	0	0	0	0	Reads slave status
H′4	0	1	0	0	Reads locked address (upper 8 bits)
H′5	0	1	0	1	Reads locked address (lower 4 bits)

(1) Slave Status Read (Control Bits: H′0, H′6)

The master unit can decide the reason the slave unit does not return the acknowledgement (ACK) by reading the slave status (H′0, H′6). The slave status indicates the result of the last communications that the slave unit performs. All slave units can provide slave status information. Figure 17.3 shows bit configuration of the slave status.

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit	Value	Description
Bit 7, bit 6	00	Mode 0 (Indicates the highest mode supported by a unit. *1)
	01	Mode 1
	10	Mode 2
	11	For future use
Bit 5	0	Fixed 0
Bit 4*2	0	Slave transmission halted
	1	Slave transmission enabled
Bit 3	0	Fixed 0
Bit 2	0	Unit is unlocked
	1	Unit is locked
Bit 1*3	0	Slave receive buffer is empty
	1	Slave receive buffer is not empty
Bit 0*4	0	Slave transmit buffer is empty
	1	Slave transmit buffer is not empty

Notes: 1. Since this LSI can support up to mode 2, bits 6 and 7 are fixed to 10.
2. The value of bit 4 can be selected by the STE bit in the IEBus master unit address register 1 (IEAR1).
3. The slave receive buffer is a buffer which is accessed during data write (control bits: H'8, H'A, H'B, H'E, H'F).
In this LSI, the slave receive buffer corresponds to the IEBus receive buffer register (IERBR); and bit 2 is the value of the RxRDY flag in the IEBus receive status register (IERSR).
4. The slave transmit buffer is a buffer which is accessed during data read (control bits: H'3, H'7).
In this LSI, the slave transmit buffer corresponds to the IEBus transmit buffer register (IETBR) when SRQ = 1 in the IEBus general flag register (IEFLG); and bit 1 is a value which reverses the TxRDY flag in the IEBus transmit status register (IETSR).

Figure 17.3 Bit Configuration of Slave Status (SSR)

(2) Data Command Transfer (Control Bits: Read (H′3, H′7), Write (H′A, H′B, H′E, H′F))

In the case of data read (H′3, H′7), data in the data buffer of the slave unit is read in the master unit. In the case of data write (H′B or H′F) or command write (H′A or H′E), data received in the slave unit is processed in accordance with the operation specification of the slave unit.

Notes: 1. The user can select data and commands freely in accordance with the system.
2. H′3, H′A, or H′B may lock depending on the communications condition and status.

(3) Locked Address Read (Control Bits: H′4, H′5)

In the case of the locked address read (H′4 or H′5), the address (12 bits) of the master unit which issues lock instruction is configured in bytes shown in figure 17.4.

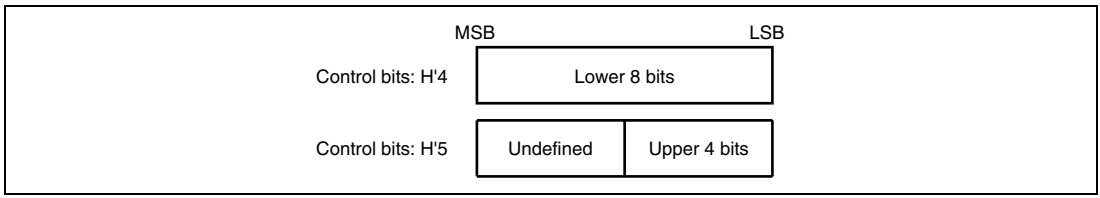


Figure 17.4 Locked Address Configuration

(4) Locking/Unlocking (Control Bits: Setting (H′3, H′A, HB), Cancellation: (H′6))

The lock function is used for message transfer over multiple communications frames. Locked unit receives data only from the unit which has locked.

Locking and unlocking are described below.

- Locking

  When the acknowledge bit of 0 in the message length field is transmitted/received with the control bits indicating the lock operation, and then the communications frame is completed before completion of data transmission/reception for the number of bytes specified by the message length bits, the slave unit is locked by the master unit. In this case, the bit (bit 2) relevant to lock in the byte data indicating the slave status is set to 1.

  Lock is set only when the number of data exceeds the maximum number of transfer bytes in one frame. Lock is not set by other error termination.
- Unlocking

  When the control bits indicate the lock (H′3, H′A, or H′B) or unlock (H′6) operation and the byte data for the number of bytes specified by the message length bits are transmitted/received in a single communications frame, the slave unit is unlocked by the master unit. In this case, a bit (bit 2) relevant to lock in the byte indicating the slave status is cleared to 0.

Note that locking and unlocking are not performed in broadcast communications.

Note: * There are three methods to unlock by a locked unit itself.
- Perform hardware reset
- Enter module stop mode
- Issue unlock command by the IEBus command register (IECMR)

Note that the LCK flag in IEFLG can be used to check whether the unit is locked/unlocked.

17.1.4 Bit Format

Figure 17.5 shows the bit format (conceptual diagram) configuring the IEBus communications frame.

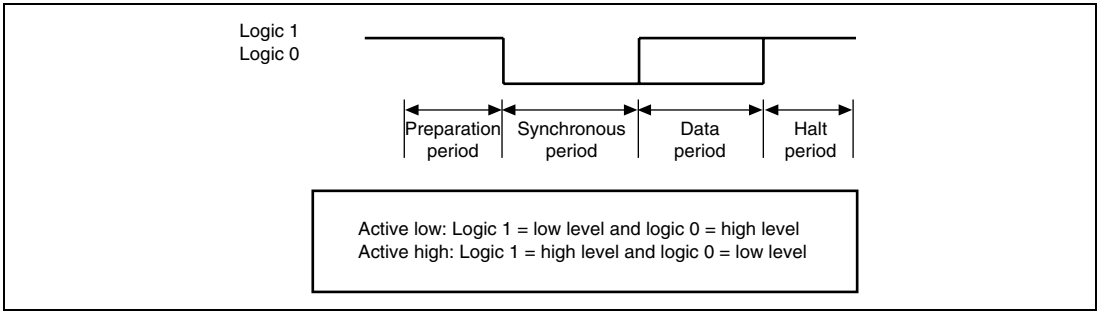


Figure 17.5 IEBus Bit Format (Conceptual Diagram)

Each period of bit format for use of active high signals is described below.

- Preparation period: first logic 1 period (high level)
- Synchronous period: subsequent logic 0 period (low level)
- Data period: period indicating bit value (logic 1: high level, logic 0: low level)
- Halt period: last logic 1 cycle (high level)

For use of active low signals, levels are reversed from the active high signals.

The synchronous and data periods have approximately the same length.

The IEBus is synchronized bit by bit. The specifications for the time of all bits and the periods allocated to the bits differ depending on the type of transfer bits and the unit (master or slave unit).

17.2 Input/Output Pins

Table 17.6 shows the IEB pin configuration.

Table 17.6 Pin Configuration

Name	Abbreviation	I/O	Function
IEBus transmit data pin	$\overline{Tx}$	O	Transmit data output pin
IEBus receive data pin	$\overline{Rx}$	I	Receive data input pin

17.3 Register Descriptions

The IEB has the following registers. For the module stop control register, see section 22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

- IEBus control register (IECTR)
- IEBUS command register (IECMR)
- IEBus master control register (IEMCR)
- IEBus master unit address register 1 (IEAR1)
- IEBus master unit address register 2 (IEAR2)
- IEBus slave address setting register 1 (IESA1)
- IEBus slave address setting register 2 (IESA2)
- IEBus transmit message length register (IETBFL)
- IEBus transmit buffer register (IETBR)
- IEBus reception master address register 1 (IEMA1)
- IEBus reception master address register 2 (IEMA2)
- IEBus receive control field register (IERCTL)
- IEBus receive message length register (IERBFL)
- IEBus receive buffer register (IERBR)
- IEBus lock address register 1 (IELA1)
- IEBus lock address register 2 (IELA2)
- IEBus general flag register (IEFLG)
- IEBus transmit status register (IETSR)
- IEBus transmit interrupt enable register (IEIET)
- IEBus transmit error flag register (IETEF)
- IEBus receive status register (IERSR)
- IEBus receive interrupt enable register (IEIER)
- IEBus receive error flag register (IEREF)

17.3.1 IEBus Control Register (IECTR)

IECTR controls IEB operation (switches IEBus pin/port functions, selects input/output level, and enables receive operation).

Bit	Bit Name	Initial Value	R/W	Description
7	IEE	0	R/W	IEB Pin Switch Switches IEB pin and port functions. 0: The PG3/$\overline{Rx}$/$\overline{CS1}$ and PG2/$\overline{Tx}$/$\overline{CS2}$ pins function as the PG3/$\overline{CS1}$ and PG2/$\overline{CS2}$ pins. 1: The PG3/$\overline{Rx}$/$\overline{CS1}$ and PG2/$\overline{Tx}$/$\overline{CS2}$ pins function as the $\overline{Tx}$ and $\overline{Rx}$ pins.
6	IOL	0	R/W	Input/Output Level Selects input/output pin level (polarity) for the $\overline{Rx}$ and $\overline{Tx}$ pins. 0: Pin input/output is set to active low. (Logic 1 is low level and logic 0 is high level.) 1: Pin input/output is set to active high. (Logic 1 is high level and logic 0 is low level.)
5	DEE	0	R/W	Broadcast Receive Error Interrupt Enable Since the acknowledgement is not returned between the master and slave units in broadcast reception, the master unit cannot decide whether the slave unit is in the receive enabled state. If this bit is set to 1, a reception error interrupt occurs (note that there is not the corresponding bit in the IEBus receive error flag register to this error) when the receive buffer is not in the receive enabled state during receiving the control field in broadcast reception (when the RE bit is not set to 1 or the RxRDY flag is set.). At this time, the master address is stored in IEMA1 and IEMA2. The receive data is not stored in the receive control field register. While this bit is 0, a reception error interrupt does not occur when the receive buffer is not in the receive enabled state, and the reception stops and enters the wait state. The master address is not saved. 0: A broadcast reception error is not generated up to the control field. 1: A broadcast receive error is generated up to the control field.

Bit	Bit Name	Initial Value	R/W	Description
4	CKS1	0	R/W	Input Clock Select Selects clock used by the IEB. See table 17.7.
3	RE	0	R/W	Receive Enable Enables/disables IEB reception. This bit must be set at the initial setting before frame reception. Changing this bit before receiving the control field is valid, however, changing this bit after receiving the control field is invalid and the value before the change is validated. 0: Reception is disabled. 1: Reception is enabled.
2	LUEE	0	R/W	Last Byte Underrun Enable Sets whether to generate an underrun error when the last data field byte is transferred in data transmission. If the IEB reads from IETBR when the TxRDY flag is set (the transmit buffer register (IETBR) is empty), an underrun error occurs. In transmission using the DTC, an underrun error occurs at the last byte transmission if the CPU did not clear the TxRDY flag, because the DTC does not clear the TxRDY flag. When the DTC is used, set this bit to 0 to mask an underrun error generated at the last byte transmission. When the DTC is not used, set this bit to 1 to generate an underrun error at the last byte transmission. 0: An underrun error does not occur at the last byte transmission (when using the DTC) 1: An underrun error does not occur at the last byte transmission (when not using the DTC)
1	CKS0	0	R/W	Input Clock Select Selects clock used by the IEB. See table 17.7.
0	—	0	R/W	Reserved This bit is always read as 0 and cannot be modified.

Table 17.7 List of System Clock Division Ratio

Bit 4	Bit 1	
CKS1	**CKS0**	**Function**
0	0	1/4 system clock is used (ϕ = 24MHz, 25.16 MHz)
0	1	1/3 system clock is used (ϕ = 18MHz, 18.87 MHz)
1	0	1/2 system clock is used (ϕ = 12MHz, 12.58 MHz)
1	1	Setting prohibited

17.3.2 IEBus Command Register (IECMR)

IECMR issues commands to control IEB communications.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	W	Reserved The read value is undefined. In order to avoid malfunction, do not use bit manipulation instructions. These bits cannot be modified.
2	CMD2	0	W	Command Bits These bits issue a command to control IEB communications. When the CMX flag in IEFLG is set after the command issuance, the command is indicated to be in execution. When the CMX flag becomes 0, the operation state is entered. These bits are read as 0. The read value is undefined. Do not use a bit manipulation instruction that causes malfunction. 000: No operation. Operation is not affected. 001: Unlock (required from other units)*1 010: Requires communications as the master 011: Stops master communications*2 100: Undefined bits. Operation is not affected by this command. 101: Requires data transfer from the slave. 110: Stops data transfer from the slave*3. 111: Undefined bits. Operation is not affected by this command.
1	CMD1	0	W	
0	CMD0	0	W	

Notes: 1. Do not execute this command in slave communications. Execute this command after slave communications ends or in master communications. If this command is issued in slave communications, this command is ignored.

2. This command is valid during master communications (MRQ = 1). In other states, this command issuance is ignored. If this command is issued in master communications, the communications controller immediately enters the wait state. At this time, the issued master transmission request ends (MRQ = 0).

3. This command is valid during slave communications (SRQ = 1). In other states, this command issuance is ignored. Once this command was issued in slave transmission, the SRQ flag is 0 before slave transmission. Therefore, a transmit request from the master is not responded. If a transmit request is issued during slave transmission, the transmission stops and the wait state is entered (SRQ = 0).

17.3.3 IEBus Master Control Register (IEMCR)

IEMCR sets communications conditions for master communications (selection of broadcast or normal communications, retransmission counts at arbitration loss, and control bits value). It is not necessary to set this register for slave communications.

Bit	Bit Name	Initial Value	R/W	Description
7	SS	1	R/W	Broadcast/Normal Communications Select Selects broadcast or normal communications for master communications. 0: Broadcast communications 1: Normal communications
6	RN2	0	R/W	Retransmission Counts
5	RN1	0	R/W	Set the number of times retransmission is performed when arbitration is lost in master communications. If arbitration is lost for a specified number of times, the AL flag in IETEF is set and transmission ends with a transmit error. If arbitration is won during retransmission, the retransmission count is automatically restored to the initial setting after master address transfer. 000: 0 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7
4	RN0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	CTL3*1	0	W	Control bits
2	CTL2	0	W	Set the control bits in the control field for master transmission.
1	CTL1	0	W	
0	CTL0	0	W	0000: Reads slave status
				0001: Undefined. Setting prohibited.
				0010: Undefined. Setting prohibited.
				0011: Reads data and locks*2
				0100: Reads locked address (lower 8 bits)
				0101: Reads locked address (upper 4 bits)
				0110: Reads slave status and unlocks*2
				0111: Reads data
				1000: Undefined. Setting prohibited.
				1001: Undefined. Setting prohibited.
				1010: Writes command and locks*2
				1011: Writes data and locks*2
				1100: Undefined. Setting prohibited.
				1101: Undefined. Setting prohibited.
				1110: Writes command
				1111: Writes data

Notes: 1. CTL3 decides the data transfer direction of the message length bits in the message length field and data bits in the data field:
CTL3 = 1: Transfer is performed from master unit to slave unit
CTL3 = 0: Transfer is performed from slave unit to master unit
2. Control bits to lock and unlock

17.3.4 IEBus Master Unit Address Register 1 (IEAR1)

IEAR1 sets the lower 4 bits of the master unit address and communications mode. In master communications, the master unit address becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

Bit	Bit Name	Initial Value	R/W	Description
7	IAR3	0	R/W	Lower 4 Bits of IEBus Master Unit Address
6	IAR2	0	R/W	Set the lower 4 bits of the master unit address.
5	IAR1	0	R/W	
4	IAR0	0	R/W	
3	IMD1	0	R/W	IEBus Communications Mode
2	IMD0	0	R/W	Set IEBus communications mode. 00: Communications mode 0 01: Communications mode 1 10: Communications mode 2 11: Setting prohibited
1	—	0	R/W	Reserved This bit is always read as 0 and cannot be modified.
0	STE	0	R/W	Slave Transmission Setting Sets bit 4 in the slave status register. Transmitting the slave status register informs the master unit that the slave transmission enabled state is entered by setting this bit to 1. Note that this bit only sets the slave status register value and does not affect slave transmission directly. 0: Bit 4 in the slave status register is 0 (slave transmission stop state) 1: Bit 4 in the slave status register is 1 (slave transmission enabled state)

17.3.5 IEBus Master Unit Address Register 2 (IEAR2)

IEAR2 sets the upper 8 bits of the master unit address. In master communications, this register becomes the master address field value. In slave communications, this register is compared with the received slave address field.

Bit	Bit Name	Initial Value	R/W	Description
7	IAR11	0	R/W	Upper 8 Bits of IEBus Master Unit Address
6	IAR10	0	R/W	Set the upper 8 bits of the master unit address.
5	IAR9	0	R/W	
4	IAR8	0	R/W	
3	IAR7	0	R/W	
2	IAR6	0	R/W	
1	IAR5	0	R/W	
0	IAR4	0	R/W	

17.3.6 IEBus Slave Address Setting Register 1 (IESA1)

IESA1 sets the lower 4 bits of the communications destination slave unit address. For slave communications, it is not necessary to set this register.

Bit	Bit Name	Initial Value	R/W	Description
7	ISA3	0	R/W	Lower 4 Bits of IEBus Slave Address
6	ISA2	0	R/W	These bits set the lower 4 bits of the communications destination slave unit address
5	ISA1	0	R/W	
4	ISA0	0	R/W	
3 to 0	—	All 0	R/W	Reserved These bits are always read as 0 and cannot be modified.

17.3.7 IEBus Slave Address Setting Register 2 (IESA2)

IESA2 sets the upper 8 bits of the communications destination slave unit address. For slave communications, it is not necessary to set this register.

Bit	Bit Name	Initial Value	R/W	Description
7	ISA11	0	R/W	Upper 8 Bits of IEBus Slave Address
6	ISA10	0	R/W	Set upper 8 bits of the communications destination slave unit address
5	ISA9	0	R/W	
4	ISA8	0	R/W	
3	ISA7	0	R/W	
2	ISA6	0	R/W	
1	ISA5	0	R/W	
0	ISA4	0	R/W	

17.3.8 IEBus Transmit Message Length Register (IETBFL)

IETBFL sets the message length for master or slave transmission.

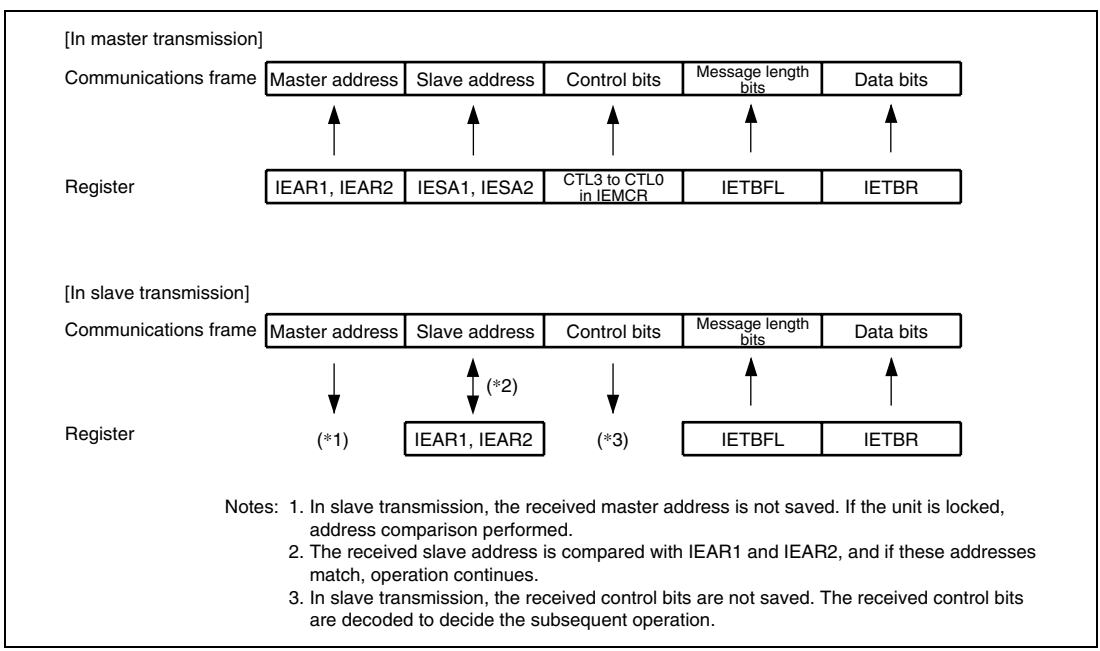
Bit	Bit Name	Initial Value	R/W	Description
7	TBFL7	0	R/W	Transmit Message Length
6	TBFL6	0	R/W	Set the message length for master or slave transmission.
5	TBFL5	0	R/W	
4	TBFL4	0	R/W	If a value exceeding the maximum transmit bytes for one frame is set in IETBFL, communications are performed with two or more frames in some communications modes. In this case, in or after the second frame, the message length value should be the number of bytes of the remaining communications data, however, the initial IETBFL setting remains unchanged. Therefore, for the second frame or after, re-set the number of bytes of the remaining communications data.
3	TBFL3	0	R/W	
2	TBFL2	0	R/W	
1	TBFL1	0	R/W	
0	TBFL0	0	R/W	

17.3.9 IEBus Transmit Buffer Register (IETBR)

IETBR is a 1-byte buffer to which data to be transmitted in master or slave transmission is written. IETBR is empty when the TxRDY flag in IETSR is 1. Check the TxRDY flag before setting transmit data in IETBR.

Data written in IETBR is transmitted in the data field in master or slave transmission. Figure 17.6 shows the correspondence between the communications signal format and registers for IEBus data transfer.

Bit	Bit Name	Initial Value	R/W	Description
7	TBR7	0	R/W	Data to be transmitted is written to this 1-byte buffer.
6	TBR6	0	R/W	
5	TBR5	0	R/W	
4	TBR4	0	R/W	
3	TBR3	0	R/W	
2	TBR2	0	R/W	
1	TBR1	0	R/W	
0	TBR0	0	R/W	



Notes: 1. In slave transmission, the received master address is not saved. If the unit is locked, address comparison performed.
2. The received slave address is compared with IEAR1 and IEAR2, and if these addresses match, operation continues.
3. In slave transmission, the received control bits are not saved. The received control bits are decoded to decide the subsequent operation.

Figure 17.6 Transmission Signal Format and Registers in Data Transfer

17.3.10 IEBus Reception Master Address Register 1 (IEMA1)

IEMA1 indicates the lower four bits of the communications destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the timing of setting the RxS flag in IERSR.

If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state on control field reception, a receive error interrupt is generated and the lower 4 bits of the master address are stored in IEMA1. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	IMA3	0	R	Lower 4 Bits of IEBus Reception Master Address
6	IMA2	0	R	Indicates the lower 4 bits of the communications destination master unit address in slave/broadcast reception.
5	IMA1	0	R	
4	IMA0	0	R	
3 to 0	—	All 0	R	Reserved These bits are always read as 0.

17.3.11 IEBus Reception Master Address Register 2 (IEMA2)

IEMA2 indicates the upper 8 bits of the communications destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the timing of setting the RxS flag in IERSR.

If a broadcast receive error interrupt is selected with the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper 8 bits of the master address are stored in IEMA2. This register cannot be modified by a write.

Bit	Bit Name	Initial Value	R/W	Description
7	IMA11	0	R	Upper 8 Bits of IEBus Reception Master Address
6	IMA10	0	R	Indicates the upper 8 bits of the communications destination master unit address in slave/broadcast reception.
5	IMA9	0	R	
4	IMA8	0	R	
3	IMA7	0	R	
2	IMA6	0	R	
1	IMA5	0	R	
0	IMA4	0	R	

17.3.12 IEBus Receive Control Field Register (IERCTL)

IERCTL indicates the control field value in slave/broadcast reception. This register is enabled when slave/broadcast receive starts, and the contents are changed at the timing of setting the RxS flag in IERSR.

This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0.
3	RCTL3	0	R	IEBus Receive Control Field Indicates the control field value in slave/broadcast reception.
2	RCTL2	0	R	
1	RCTL1	0	R	
0	RCTL0	0	R	

17.3.13 IEBus Receive Message Length Register (IERBFL)

IERBFL indicates the message length field in slave/broadcast reception. This register is enabled when slave/broadcast receive starts, and the contents are changed at the timing of setting the RxS flag in IERSR.

This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	RBFL7	0	R	IEBus Receive Message Length Indicates the contents of message length field in slave/broadcast reception.
6	RBFL6	0	R	
5	RBFL5	0	R	
4	RBFL4	0	R	
3	RBFL3	0	R	
2	RBFL2	0	R	
1	RBFL1	0	R	
0	RBFL0	0	R	

17.3.14 IEBus Receive Buffer Register (IERBR)

IERBR is a 1-byte read-only buffer that stores data received in master or slave reception. This register can be read when the RxRDY flag in IERSR is set to 1. This register indicates the data field value both in master and slave receptions. This register cannot be modified.

Figure 17.7 shows the relationship between transmission signal format and registers in IEBus data reception.

Bit	Bit Name	Initial Value	R/W	Description
7	RBR7	0	R	One-byte read-only buffer that stores data received in master or slave reception
6	RBR6	0	R	
5	RBR5	0	R	
4	RBR4	0	R	
3	RBR3	0	R	
2	RBR2	0	R	
1	RBR1	0	R	
0	RBR0	0	R	

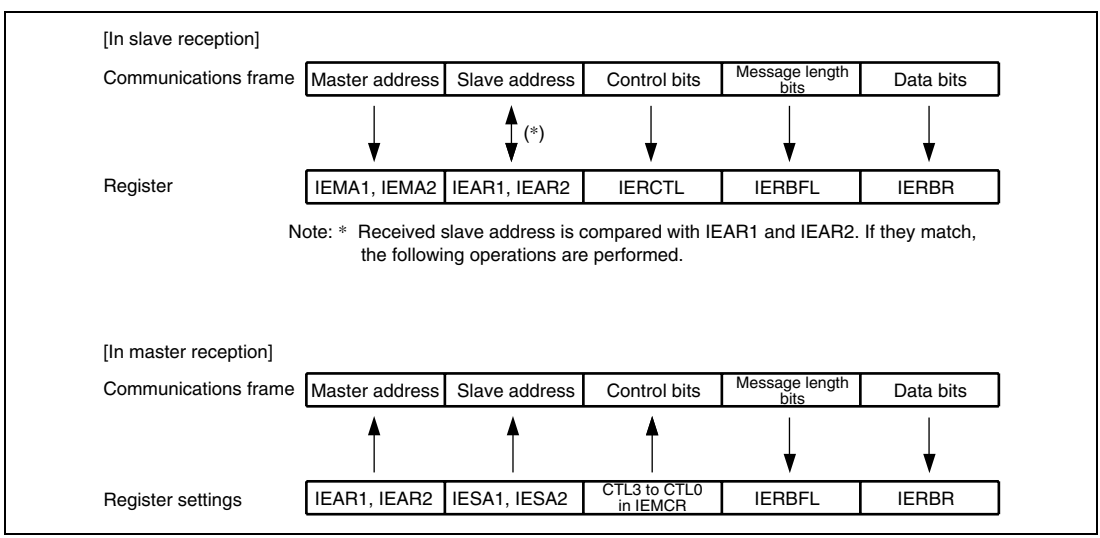


Figure 17.7 Relationship between Transmission Signal Format and Registers in IEBus Data Reception

17.3.15 IEBus Lock Address Register 1 (IELA1)

IELA1 specifies the lower 8 bits of a locked address when a unit is locked. Data in this register is valid when the LCK flag in IEFLG is set to 1. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	ILA7	0	R	Lower 8 Bits of IEBus Lock Address
6	ILA6	0	R	Stores the lower 8 bits of the master unit address when a unit is locked.
5	ILA5	0	R	
4	ILA4	0	R	
3	ILA3	0	R	
2	ILA2	0	R	
1	ILA1	0	R	
0	ILA0	0	R	

17.3.16 IEBus Lock Address Register 2 (IELA2)

IELA2 is an 8-bit read-only register that specifies the upper 4 bits of a locked address when a unit is locked. Data in this register is valid when the LCK flag in IEFLG is set to 1. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0.
3	ILA11	0	R	Upper 4 Bits of IEBus Locked Address
2	ILA10	0	R	Stores the upper 4 bits of the master unit address when a unit is locked.
1	ILA9	0	R	
0	ILA8	0	R	

17.3.17 IEBus General Flag Register (IEFLG)

IEFLG indicates the IEB command execution status, lock status and slave address match, and broadcast reception detection. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	CMX	0	R	Command Execution Status Indicates the command execution status. 1: A command is being executed [Setting condition] • When a master communications request or slave transmit request command is issued while the MRQ, SRQ, or SRE flag is set to 1 0: A command execution is completed [Clearing condition] • When a command execution has been completed
6	MRQ	0	R	Master Communications Request Indicates whether or not the unit is in communications request state as a master unit. 1: The unit is in communications request state as a master unit [Setting condition] • When the CMX flag is cleared to 0 after the master communications request command is issued 0: The unit is not in communications request status as a master unit [Clearing condition] • When the master communications have been completed

Bit	Bit Name	Initial Value	R/W	Description
5	SRQ	0	R	Slave Transmission Request Indicates whether or not the unit is in transmit request status as a slave unit. 1: The unit is in transmit request status as a slave unit [Setting condition] • When the CMX flag is cleared to 0 after the slave transmit request command is issued. 0: The unit is not in transmit request status as a slave unit [Clearing condition] • When a slave transmission has been completed.
4	SRE	0	R	Slave Receive Status Indicates the execution status in slave/broadcast reception. 1: Slave/broadcast reception is being executed [Setting condition] • When the slave/broadcast reception is started while the RE bit in IECTR is set to 1. 0: Slave/broadcast reception is not being executed [Clearing condition] • When the slave/broadcast reception has been completed.
3	LCK	0	R	Lock Status Indication Set to 1 when a unit is locked by a lock request from the master unit. IELA1 and IELA2 values are valid only when this flag is set to 1. 1: A unit is locked [Setting condition] • When data for the number of bytes specified by the message length is not received after the control bits that make the unit locked are received from the master unit. (The LCK flag is set to 1 only when the message length exceeds the maximum number of transfer bytes in one frame. This flag is not set by completion of other errors.) 0: A unit is unlocked [Clearing condition] • When an unlock condition is satisfied or when an unlock command is issued.

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0.
1	RSS	0	R	Receive Broadcast Bit Status Indicates the received broadcast bit value. This flag is valid when the slave/broadcast reception is started. (This flag is changed at the timing of setting the RxS flag in IERSR.) The previous value remains unchanged until the next slave/broadcast reception is started.
0	GG	0	R	General Broadcast Reception Acknowledgement Set to 1 when the slave address is acknowledged as H′FFF in broadcast reception. As well as the receive broadcast bit, this flag is valid when the slave/broadcast reception is started. (This flag is changed at the timing of setting the RxS flag in IERSR.) The previous value remains unchanged until the next slave/broadcast reception is started. This flag is cleared to 0 in slave normal reception. [Setting condition] • When H′FFF is acknowledged in the slave field in broadcast reception [Clearing conditions] • A unit is in slave reception • When H′FFF is not acknowledged in slave field in broadcast reception

17.3.18 IEBus Transmit Status Register (IETSR)

IETSR detects transmit data ready, transmit start, transmit normal completion, transmit completion with an error, or runaway states. Each status flags in IETSR corresponds to a bit in the IEBus transmit interrupt enable register (IEIET) that enables or disables each interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	TxRDY	1	R/W	Transmit Data Ready Indicates that the next data can be written to IETBR since IETBR is empty. This flag is automatically cleared by DTC* data transfer. When data is transmitted by the CPU, this flag must be cleared by software. This flag is cleared by writing 0 after reading a 1 from this flag. [Setting conditions] • Immediately after reset • When data can be written to IETBR (when IEB has loaded data from IETBR to the transmit shift register.) [Clearing conditions] • When writing 0 after reading TxRDY = 1 • When data is written to TBR by the DTC by a TxRDY request. Note: This flag is not cleared on the end byte of DTC transfer.
6 to 4	—	All 0	R/W	Reserved These bits are always read as 0 and cannot be modified.
3	IRA	0	R/W	IEBus Runaway State Indicates that the on-chip microprogram for IEBus control is in the runaway states. This flag is set to 1 when a runaway occurs during either IEBus transmission or reception. (This flag is not a transfer specific flag and is also set for a reception runaway.) [Setting condition] • When the on-chip microprogram is in the runaway states [Clearing condition] • When writing 0 after reading IRA = 1

Bit	Bit Name	Initial Value	R/W	Description
2	TxS	0	R/W	Transmit Start Detection Indicates that the IEB starts transmission. [Setting conditions] • Master transmission: When the arbitration is won and when the master address field transmission is completed • Slave transmission: When the control bits of H′3 (0011) or H′7 (0111) is received from the master unit meaning that data transfer is requested [Clearing condition] • When writing 0 after reading TxS = 1
1	TxF	0	R/W	Transmit Normal Completion Indicates that data for the number of bytes specified by the message length bits has been transmitted with no error. [Setting condition] • When data for the number of bytes specified by the message length bits has been transmitted normally [Clearing condition] • When writing 0 after reading TxF = 1
0	TxE	0	R/W	Transmit Error Completion Indicates that data for the number of bytes specified by the message length bits is not completed and that the data transmission is terminated. The source of this error can be checked by the contents of IETEF. This flag is set at the timing that an error indicated by IETEF occurs. The TxE flag can be cleared even when the error source flag in IETEF is set to 1 because the TxE flag is not logically ORed with the flags in IETEF. In master reception, an error (arbitration loss, timing error, or NAK reception) generated after a master communications command is issued before master reception starts will be detected as a transmit error. [Setting condition] • When the data for the number of bytes specified by the message length bits is not completed and when the transmission is terminated [Clearing condition] • When writing 0 after reading TxE = 1

17.3.19 IEBus Transmit Interrupt Enable Register (IEIET)

IEIET enables/disables IETSR transmit ready, transmit start, transmit normal completion, transmit completion with an error, and runaway interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	TxRDYE	0	R/W	Transmit Data Ready Interrupt Enable Enables/disables a transmit data ready interrupt. 0: Disables a transmit data ready (TxRDY) interrupt 1: Enables a transmit data ready (TxRDY) interrupt
6 to 4	—	All 0	R/W	Reserved These bits are always read as 0 and cannot be modified.
3	IRAE	0	R/W	IEBus Runaway State Interrupt Enable Enables/disables an IEBus runaway state interrupt. 0: Disables an IEBus runaway state interrupt (IRA) 1: Enables an IEBus runaway state interrupt (IRA)
2	TxSE	0	R/W	Transmit Start Interrupt Enable Enables/disables a transmit start (TxS) interrupt. 0: Disables a transmit start (TxS) interrupt 1: Enables a transmit start (TxS) interrupt
1	TxFE	0	R/W	Transmit Normal Completion Interrupt Enable Enables/disables a transmit normal completion (TxF) interrupt. 0: Disables a transmit normal completion (TxF) interrupt 1: Enables a transmit normal completion (TxF) interrupt
0	TxEE	0	R/W	Transmit Error Termination Interrupt Enable Enables/disables a transmit error termination (TxE) interrupt. 0: Disables a transmit error termination (TxE) interrupt 1: Enables a transmit error termination (TxE) interrupt

17.3.20 IEBus Transmit Error Flag Register (IETEF)

IETEF checks the source of a TxE interrupt indicated in IETSR. This register detects an overflow of a maximum number of bytes in one frame, arbitration loss, underrun error, timing error, and NAK reception.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved These bits are always read as 0 and cannot be modified.
4	AL	0	R/W	Arbitration Loss The IEB retransmits from the start bit for the number of times specified by bits RN2 to Rn0 in IEMCR if the arbitration has been lost in master communications. If the arbitration has been lost for the specified number of times, the AL and TxE flags are set to enter the wait state. If the arbitration has been won within retransmit for the specified number of times, this flag is not set to 1. This flag is set only when the arbitration has been lost and the wait state is entered. [Setting condition] • When the arbitration has been lost during data transmission and the transmission has been terminated [Clearing condition] • When writing 0 after reading AL = 1
3	UE	0	R/W	Underrun Error Indicates that an underrun error has occurred during data transmission. The IEB detects an underrun error occurrence when the IEB fetches data from IETBR while the TxRDY flag is set to 1, and the IEB sets the TxE flag and enters the wait state. Accordingly, when the TxRDY flag is not cleared even if data is written to IETBR, an underrun error occurs and data transmission is terminated. Note that the TxRDY flag must be cleared in data transmission by the CPU. [Setting condition] • When the IEB loads data from IETBR to the transmit shift register while the TxRDY flag is set to 1 [Clearing condition] • When writing 0 after reading UE = 1

Bit	Bit Name	Initial Value	R/W	Description
2	TTME	0	R/W	Timing Error Set to 1 if data is not transmitted at the timing specified by the IEBus protocol during data transmission. The IEB sets the TxE flag and enters the wait state. [Setting condition] • When a timing error occurs during data transmission [Clearing condition] • When writing 0 after reading TTME = 1
1	RO	0	R/W	Overflow of Maximum Number of Transmit Bytes in One Frame Indicates that the maximum number of bytes defined by communications mode have been transmitted because a NAK has been received from the receive unit and retransmit has been performed, or that transmission has not been completed because the message length value exceeds the maximum number of transmit bytes in one frame. The IEB sets the TxE flag and enters the wait state. [Setting condition] • When the transmit has not been completed although the maximum number of bytes defined by communications mode have been transmitted [Clearing condition] • When writing 0 after reading RO = 1

Bit	Bit Name	Initial Value	R/W	Description
0	ACK	0	R	Acknowledge bit Status Indicates the data received in the acknowledge bit of the data field. • Acknowledge bit other than in the data field The IEB terminates the transmission and enters the wait state if a NAK is received. In this case, this bit and the TxE flag are set to 1. • Acknowledge bit in the data field The IEB retransmits data up to the maximum number of bytes defined by communications mode until an ACK is received from the receive unit if a NAK is received from the receive unit during data field transmission. In this case, when an ACK is received from the receive unit during retransmission, this flag is not set and transmission will be continued. When transmission is terminated without receiving an ACK, this flag is set to 1. Note: This flag is invalid in broadcast communications. [Setting condition] • When the acknowledge bit of 1 (NAK) is detected [Clearing condition] • When writing 0 after reading ACK = 1

17.3.21 IEBus Receive Status Register (IERSR)

IERSR detects receive data ready, receive start, transmit/receive normal completion, or receive completion with an error. Each status flag in IERSR corresponds to a bit in the IEIER that enables/disables each interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	RxRDY	1	R/W	Receive Data Ready Indicates that the receive data is stored in IERBR and that the receive data can be read. This flag is automatically cleared by DTC* data transfer. When data is transmitted by the CPU, this flag must be cleared by software. [Setting condition] • When data reception has been completed normally and receive data has been loaded to IERBR. [Clearing conditions] • When writing 0 after reading RxRDY = 1 • When IERBR data is read by the DTC by a RxRDY request. Note: This flag cannot be cleared on the end byte of the DTC transfer.
6 to 3	—	All 0	R/W	Reserved These bits are always read as 0 and cannot be modified.
2	RxS	0	R/W	Receive Start Detection Indicates that the IEB starts reception. [Setting conditions] • Master reception: When the message length field has been received from the slave unit correctly after the arbitration is won and the control field transmission is completed • Slave reception: When the message length field has been received from the master unit correctly [Clearing condition] • When writing 0 after reading RxS = 1

Bit	Bit Name	Initial Value	R/W	Description
1	RxF	0	R/W	Receive Normal Completion Indicates that data for the number of bytes specified by the message length bits has been received and with no error. [Setting condition] • When data for the number of bytes specified by the message length bits has been received normally. [Clearing condition] • When writing 0 after reading RxF = 1
0	RxE	0	R/W	Receive Error Completion Indicates that data for the number of bytes specified by the message length bits is not completed and that the data reception is terminated. The source of this error can be checked by the contents of IEREF. This flag is set at the timing that an error indicated by IEREF occurs. The RxE flag can be cleared even when the error source flag in IEREF is set to 1 because the RxE flag is not logically ORed with the flags in IEREF. [Setting condition] • When the data for the number of bytes specified by the message length bits is not completed and when the reception is terminated. [Clearing condition] • When writing 0 after reading RxE = 1

17.3.22 IEBus Receive Interrupt Enable Register (IEIER)

IEIER enables/disables IERSR reception ready, receive start, transmit/receive normal completion, and receive completion with an error interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	RxRDYE	0	R/W	Receive Data Ready Interrupt Enable Enables/disables a receive data ready interrupt. 0: Disables a receive data ready (RxRDY) interrupt 1: Enables a receive data ready (RxRDY) interrupt
6 to 3	—	All 0	R/W	Reserved These bits are always read as 0 and cannot be modified.
2	RxSE	0	R/W	Receive Start Interrupt Enable Enables/disables a receive start (RxS) interrupt. 0: Disables a receive start (RxS) interrupt 1: Enables a receive start (RxS) interrupt
1	RxFE	0	R/W	Receive Normal Completion Enable Enables or disables a receive normal completion (RxF) interrupt. 0: Disables a receive normal completion (RxF)interrupt 1: Enables a receive normal completion (RxF)interrupt
0	RxEE	0	R/W	Receive Error Termination Interrupt Enable Enables or disables a receive error termination (RxE) interrupt. 0: Disables a receive error termination (RxE)interrupt 1: Enables a receive error termination (RxE) interrupt

17.3.23 IEBus Receive Error Flag Register (IEREF)

IEREF checks the source of an RxE interrupt indicated in IERSR. This register detects an overrun error, timing error , overflow of a maximum number of bytes in one frame, and parity error.

These flags become valid when the receive start flag (RxS) is set to 1. If an error occurs before the RxS flag is set to 1, the IEB terminates the communications and enters the wait state. In this case, these flags will not be set and the RxE flag is not set.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved These bits are always read as 0 and cannot be modified.
3	OVE	0	R/W	Overrun Control Flag Used to control the overrun during data reception. The IEB sets the OVE and RxE flags when the IEB receives the next byte data while the receive data has not been read (the RxRDY flag is not cleared) and when the parity bit reception has been started. If this flag remains set until acknowledge bit transfer, the IEB assumes that an overrun error has occurred and returns a NAK to the communications destination unit. The communications destination unit retransmits data up to the maximum number of transmit bytes. The IEB, however, returns a NAK when this flag remains set because the IEB assumes that the overrun error has not been cleared. If this flag is cleared to 0, the IEB decides that the overrun error has been cleared, returns an ACK, and receives the next data. In broadcast reception, if this flag is set during acknowledge bit transmission, the IEB immediately enters the wait state. [Setting condition] • When the next byte data is received while the RxRDY flag is not cleared and when the parity bit of the data is received. [Clearing condition] • When writing 0 after reading OVE = 1
2	RTME	0	R/W	Timing Error Set to 1 if data is not received at the timing specified by the IEBus protocol during data reception. The IEB sets the RxE flag and enters the wait state. [Setting condition] • When a timing error occurs during data reception [Clearing condition] • When writing 0 after reading RTME = 1

Bit	Bit Name	Initial Value	R/W	Description
1	DLE	0	R/W	Overflow of Maximum Number of Receive Bytes in One Frame Indicates that the maximum number of bytes defined by communications mode have been received because a parity error or overrun error occurred, or that the reception has not be completed because the message length value exceeds the maximum number of receive bytes in one frame. The IEB sets the RxE flag and enters the wait state. [Setting condition] • When the reception has not been completed although the maximum number of bytes defined by communications mode have been received. [Clearing condition] • When writing 0 after reading DLE = 1
0	PE	0	R	Parity Error Indicates that a parity error has occurred during data field reception. If a parity error occurs before data field reception, the IEB immediately enters the wait state and the PE flag is not set. If a parity error occurs when the maximum number of receive bytes in one frame has not been received, the PE flag is not set. When a parity error occurs, the IEB returns a NAK to the communications destination unit via the acknowledge bit. In this case, the communications destination unit continues retransfer up to the maximum number of receive bytes in one frame and if the reception has been completed normally by clearing the parity error, the PE flag is not set. If the parity error is not cleared when the reception is terminated before receiving data for the number of bytes specified by the message length, the PE flag is set. In broadcast reception, if a parity error occurs during data field reception, the IEB enters the wait state immediately after setting the PE flag. [Setting condition] • When the parity bit of last data of the data field is not correct after the maximum number of receive bytes has been received [Clearing condition] • When writing 0 after reading PE = 1

17.4 Operation Descriptions

17.4.1 Master Transmit Operation

This section describes an example of master transmission using the DTC after slave reception.

(1) IEB Initialization

(a) Setting the IEBus Control Register (IECTR)

Enable the IEBus pins, select the signal polarity, and select a clock supplied to the IEB. Clear the LUEE bit to 0 since the transfer is performed by the DTC.

(b) Setting the IEBus Master Unit Address Registers 1 and 2 (IEAR1 and IEAR2)

Specify the master unit address and specify the communications mode in IEAR1.

(c) Setting the IEBus Slave Address Setting Registers 1 and 2 (IESA1 and IESA2)

Specify the communications destination slave unit address.

(d) Setting the IEBus Master Control register (IEMCR)

Select broadcast/normal communications, specify the number of retransfer counts at arbitration loss, and specify the control bits.

(e) Setting the IEBus Transmit Message Length Register (IETBFL)

Specify the message length bits.

(f) Setting the IEBus Transmit Interrupt Enable Register (IEIET)

Enable TxRDY (IETxI), TxS, TxF, and TxE (IETSI) interrupts.

The above registers can be specified in any order. (The register specification order does not affect the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for the DTC transfer in the vector address (H′000004D4) to be accessed when a DTC transfer request is generated.
2. Set the following data from the start address of the RAM.
   - Transfer source address (SAR): Start address of the RAM which stores data to be transmitted in the data field.
   - Transfer destination address (DAR): Address (H′FFF808) of the IEBus transmit buffer register (IETBR)
   - Transfer count (CRA): The same value as the IETBFL contents
3. Set DTCEG5 in the DTC enable register G (DTCERG) to enable the TxRDY interrupt (IETxI).

Because the TxRDY flag is retained after a reset, the DTC transfer starts when the IETxI is enabled and the first data for the data field is written to IETBR. The DTC negates the TxRDY flag and the first byte of DTC transfer is completed.

(3) Master Transmission Flow

Figure 17.8 shows the master transmission flow. Numbers in the following description correspond to the number in figure 17.8.

1. After the IEB and DTC have been initialized, a master communications request command is issued from IECMR. During slave reception, the command execution status flag (CMX) in IEFLG is set and the master communications request will not be issued.
2. When the slave reception has been completed, the CMX flag is cleared, the master communications command is executed, and the MRQ flag is set.
3. The transmit start detection flag (TxS) in IETSR is set when arbitration is won and the master address has been transmitted. In this case, one of the transmit status interrupts (IETSI) is requested to the CPU, and the TxS flag is cleared in the interrupt handling routine.
4. The IEB loads data to be transmitted in the data field from IETBR when the control and message length fields have been transmitted and an ACK is received in each field. After that, the TxRDY flag is set. A DTC transfer request is generated by IETxI and the second byte is written to the transmit buffer.
5. Similarly, the data field load and transmission are repeated.
6. The DTC completes the data transfer for the number of specified bytes when data to be transmitted in the last byte is written to. At this time, the DTC does not clear the TxRDY flag. It, however, clears bit DTCEG5 in the DTC enable register G (DTCERG) so as not to generate more DTC transfer request.
7. A TxRDY interrupt (IETxI) is issued to the CPU when the DTC transfer is completed. In this interrupt handling routine, the TxRDY flag can be cleared. However, since a TxRDY interrupt will be generated again after the last byte transfer, the TxRDY flag remains set. (Note that the LUEE bit must be cleared to 0 because an underrun error occurs to terminate the transfer if the LUEE bit in IECTR is set to 1.) Note, however, that the TxRDY interrupt must be disabled because the TxRDY interrupt is always generated.
8. A transmit normal completion (TxF) interrupt (IETSI) occurs after the last data transfer is completed. In this case, the CPU clears the TxF flag and completes the normal completion interrupt and clears the MRQ flag to 0.

Note: As a transmit status interrupt (IETSI), the transmit error termination (TxE) interrupt as well as the transfer start detection (TxS) and transmit normal completion (TxF) interrupts must be enabled. If an error termination interrupt is disabled, no interrupt is generated even if the transmission is terminated by an error.

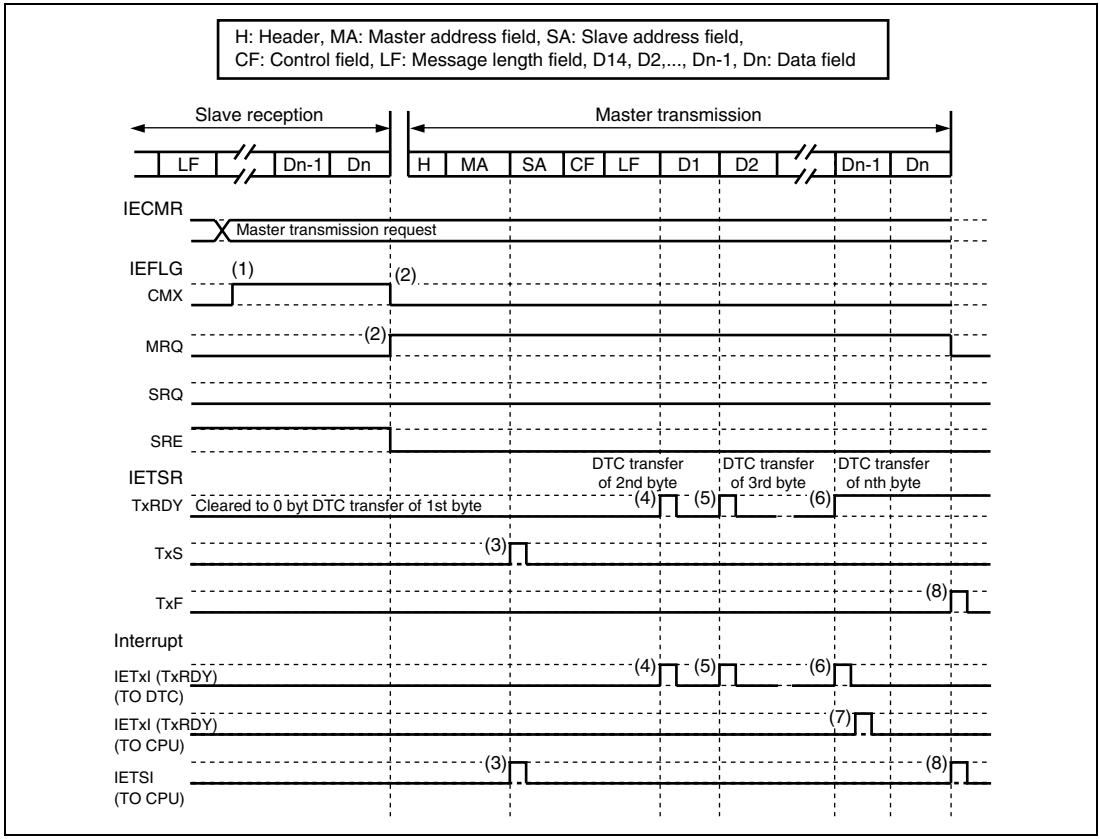


Figure 17.8 Master Transmit Operation Timing

17.4.2 Slave Receive Operation

This section describes an example of performing a slave reception using the DTC.

(1) IEB Initialization

(a) Setting the IEBus Control Register (IECTR)

Enable the IEBus pins, select the signal polarity, and select a clock supplied to the IEB. Set the RE bit to 1 to perform reception. The LUEE bit does not need to be specified.

(b) Setting the IEBus Master Unit Address Registers 1 and 2 (IEAR1 and IEAR2)

Specify the master unit address and specify the communications mode in IEAR1. Compare with the slave address in the communications frame and receive the frame if matched.

(c) Setting the IEBus Receive Interrupt Enable Register (IEIER)

Enable RxRDY (IERxI), RxS, and RxE (IERSI) interrupts.

The above registers can be specified in any order. (The register specification order does not affect the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for the DTC transfer in the vector address (H′000004D2) to be accessed when a DTC transfer request is generated.
2. Specify the following from the start address of the RAM.
   — Transfer source address (SAR): Address (H′FFF80D) of the IEBus receive buffer register (IERBR).
   — Transfer destination address (DAR): Start address of the RAM which stores data received from the data field.
   — Transfer count (CRA): Maximum number of transfer bytes in one frame in the transfer mode.
3. Set DTCEG6 in the DTC enabler register G (DTCERG) to enable the RxRDY interrupt (IETxI).

Because the above settings are performed before the frame reception, the length of data to be received cannot be decided. Accordingly, the maximum number of transfer bytes in one frame is specified as the DTC transfer count.

If the DTC is specified after reception starts, the above settings are performed in the receive start (RxS) interrupt handling routine. In this case, the transfer count must be the same value as the contents of the IEBus receive message length register (IERBFL).

(3) Slave Reception Flow

Figure 17.9 shows the slave reception flow. Numbers in the following description correspond to the number in Figure 17.9. In this example, the DTC is specified when the frame reception starts.

1. After the broadcast reception has been completed, the slave reception is performed. The receive broadcast bit status flag (RSS) in IEFLG retains the previous frame information (set to 1) until the receive start detection flag (RxS) is set to 1. If the RSS flag changes at the timing of header reception, the interrupt handling of the broadcast reception completion must be completed before the header reception. Accordingly, the RSS flag is stipulated that it changes at the timing of starting reception.
2. If data is received up to the message length field, a receive start detection (RxS) interrupt (receive status interrupt (IERSI)) will occur and the SRE flag is set to 1. In this case, the DTC initialization described in (2) is performed. After initialization, the RxS flag is cleared to 0.
3. When the first data is received, the RxRDY flag is set to 1. A DTC transfer request by IERxI occurs, and the DTC loads data from the IEBus receive buffer register (IERBR) and clears the RxRDY flag.
4. Similarly, the data field reception and load are repeated.
5. When the last data is received, the DTC completes the data transfer for the specified number of bytes after loading the receive data to the RAM. In this case, the DTC does not clear the

RxRDY flag. It, however, clears the DTC enable register G (DTCEG). Accordingly, hereafter, no transfer request will be issued to the DTC.

6. When the DTC transfer has been completed, an RxRDY interrupt (IERxI) is issued to the CPU. In this interrupt handling routine, the RxRDY flag is cleared.
7. When the last data is received, a receive normal completion (RxF) interrupt (IERSI) occurs. In this case, the CPU clears the RxF flag in order to complete the normal completion interrupt. The SRE flag is cleared to 0.

Notes: 1. As a receive status interrupt (IERSI), the receive error termination (RxE) interrupt as well as the receive start detection (RxS) and receive normal completion (RxF) interrupts must be enabled. If an error termination interrupt is disabled, no interrupt is generated even if the reception is terminated by an error.

2. The interrupt occurs after the DTC transfer has been completed. Accordingly, the interrupt described in item 6 actually occurs after item 7 above.

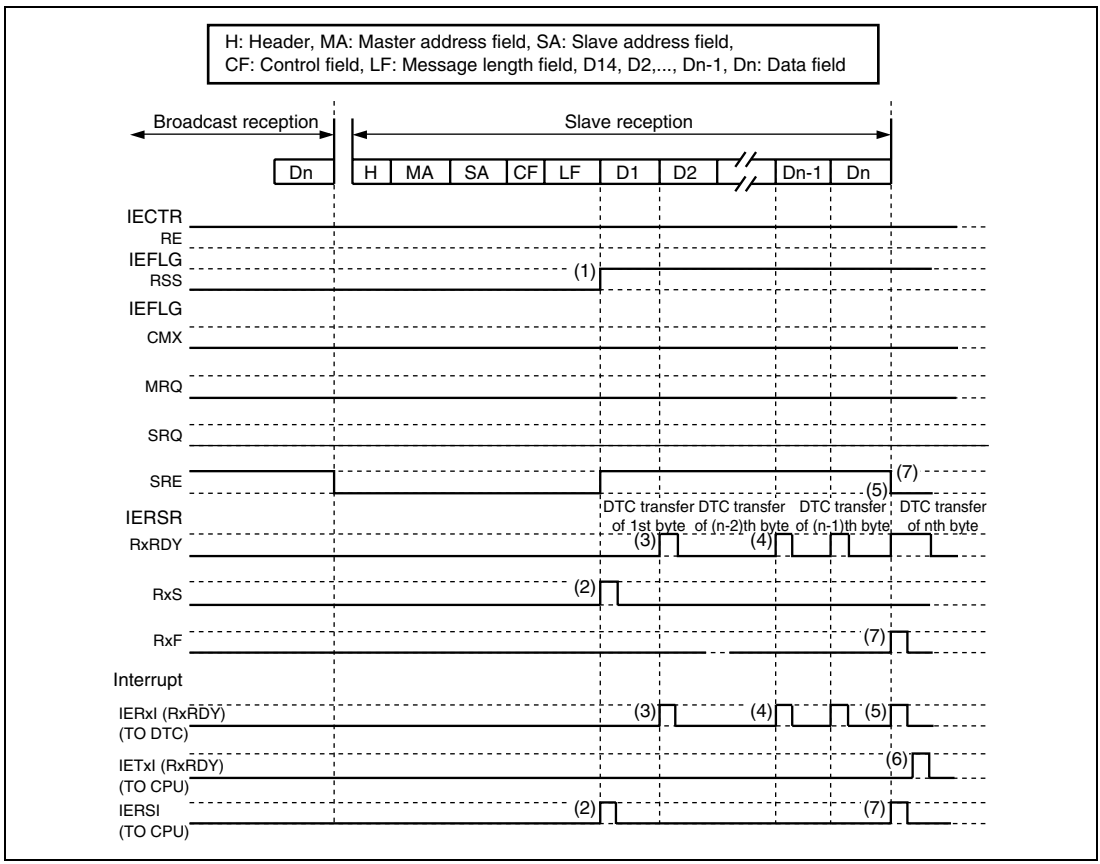


Figure 17.9 Slave Reception Operation Timing

(4) When an Error Occurs in Broadcast Reception (DEE=1)

Figure 17.10 shows an example in which a receive error occurs because the receive preparation cannot be completed (the RxRDY flag is not cleared) until the control field is received in broadcast reception after the slave reception while the DEE bit is set to 1.

Note: The same as the case in which the RE bit is not set before the control field reception.

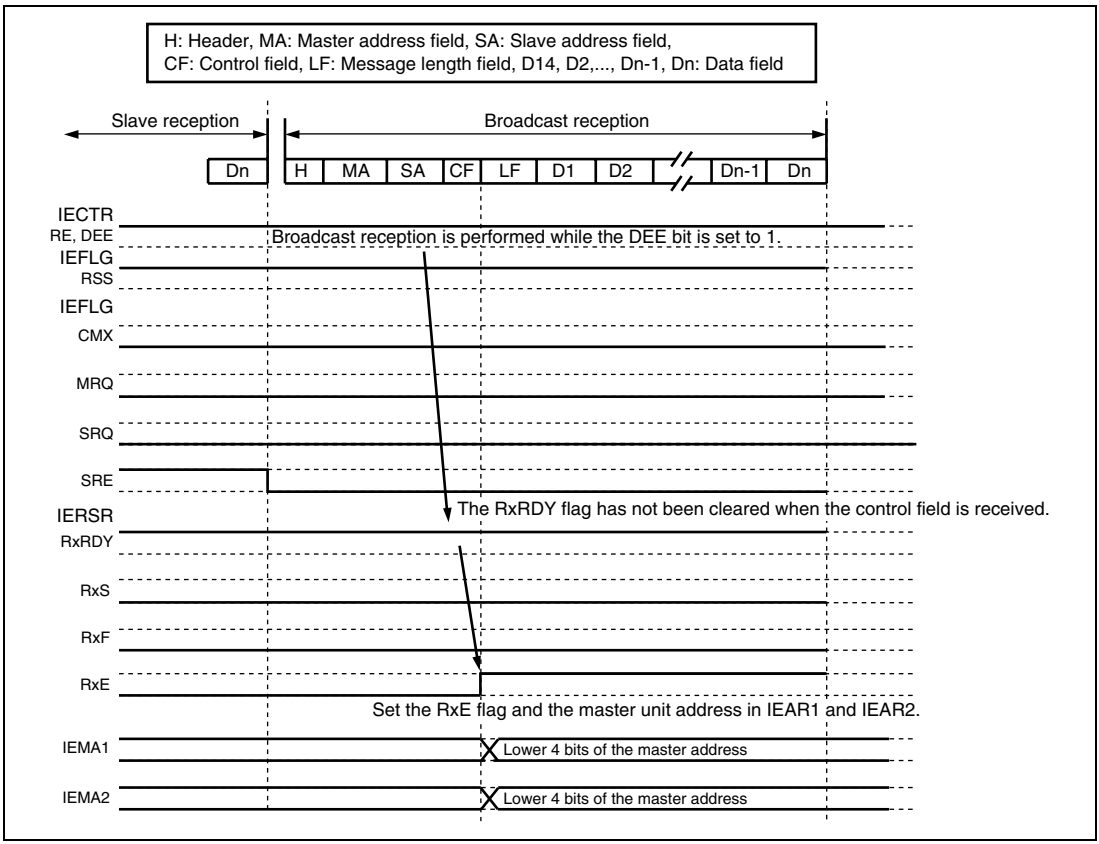


Figure 17.10 Error Occurrence in the Broadcast Reception (DEE=1)

17.4.3 Master Reception

This section shows an example of performing a master reception using the DTC after slave reception.

(1) IEB Initialization

(a) Setting the IEBus Control Register (IECTR)

Enable the IEBus pins, select the signal polarity, and select a clock supplied to the IEB. Set the RE bit to 1 to perform reception. The LUEE bit does not need to be specified.

(b) Setting the IEBus Master Unit Address Registers 1 and 2 (IEAR1 and IEAR2)

Specify the master unit address and specify the communications mode in IEAR1. Compare with the slave address in the communications frame and receive the frame if matched.

(c) Setting the IEBus Slave Address Setting Registers 1 and 2 (IESA1 and IESA2)

Specify the communications destination slave unit address.

(d) Setting the IEBus Master Control Register (IEMCR)

Select broadcast/normal communications, specify the number of retransfer counts at arbitration loss, and specify the control bits.

(e) Setting the IEBus Receive Interrupt Enable Register (IEIER)

Enable the RxRDY (IERxI), RxS, RxF, and RxE (IERSI) interrupts.

The above registers can be specified in any order. (The register specification order does not affect the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for the DTC transfer in the vector address (H′000004D2) to be accessed when a DTC transfer request is generated.
2. Set the following data from the start address of the RAM.
   - Transfer source address (SAR): Address (H′FFF80D) of the IEBus receive buffer register (IERBR).
   - Transfer destination address (DAR): Start address of the RAM which stores data to be received from the data field.
   - Transfer count (CRA): Maximum number of transfer bytes in one frame in the transfer mode.
3. Set bit DTCEG6 in the DTC enabler register G (DTCERG), and enable the RxRDY interrupt (IERxI).

Because the above settings are performed before frame reception, the length of data to be received cannot be determined. Accordingly, the maximum number of transfer bytes in one frame is specified as the DTC transfer count.

If the DTC is specified after reception starts, the above settings are performed in the receive start detection (RxS) interrupt handling routine. In this case, the transfer count must be the same value as the contents of the IEBus receive message length register (IERBFL).

(3) Master Reception Flow

Figure 17.11 shows the master reception flow. Numbers in the following description correspond to the number in figure 17.11. In this example, the DTC is specified when the frame reception starts.

1. After the IEB has been initialized, a master communications request command is issued from IECMR. During slave reception, the command execution status flag (CMX) in IEFLG is set and the master communications request will not be issued.
2. The CMX flag is cleared when the slave reception is completed, the master communications command is executed, and the MRQ flag is set.
3. If the arbitration is won, the master address, slave address, and control field will be transmitted. An error generated before the control field transmission will be handled as a transmission error. In this case, the TxE flag is set and the error contents will be reflected in IETEF.
4. The message length field is received from the slave unit. If no parity error is detected and reception is performed correctly, the receive start detection flag (RxS) is set to 1. If a parity error occurs, it is handled as a receive error. A receive start detection (RxS) interrupt (receive status interrupt (IERSI)) occurs and the DTC initialization described in (2) is performed. After DTC initialization, the RxS flag is cleared to 0.
5. When the first data is received, the RxRDY flag is set to 1. A DTC transfer request by IERxI occurs and the DTC loads data from the IEBus receive buffer register (IERBR) and clears the RxRDY flag.
6. Similarly, the above data field receive and load operations are repeated.
7. When the last data is received, the DTC completes the data transfer for the specified number of bytes after loading the receive data to the RAM. In this case, the DTC does not clear the RxRDY flag. It, however, clears the DTC enable register G (DTCEG). Accordingly, hereafter, no transfer request will be issued to the DTC.
8. When the DTC transfer has been completed, an RxRDY interrupt (IERxI) is issued to the CPU. In this interrupt handling routine, the RxRDY flag is cleared.
9. When the last data is received, a receive normal completion (RxF) interrupt (IERSI) occurs. In this case, the CPU clears the RxF flag to complete the receive normal completion interrupt. The MRQ flag is cleared to 0.

Notes: 1. As a receive status interrupt (IERSI), an receive error completion (RxE) interrupt as well as the receive start detection (RxS) and receive normal completion (RxF) interrupts must be enabled. If a receive error completion interrupt is disabled, no interrupt is generated even if the reception is terminated by an error.

2. The interrupt occurs after the DTC transfer has been completed. Accordingly, the interrupt described in item 8 actually occurs after item 9 above.

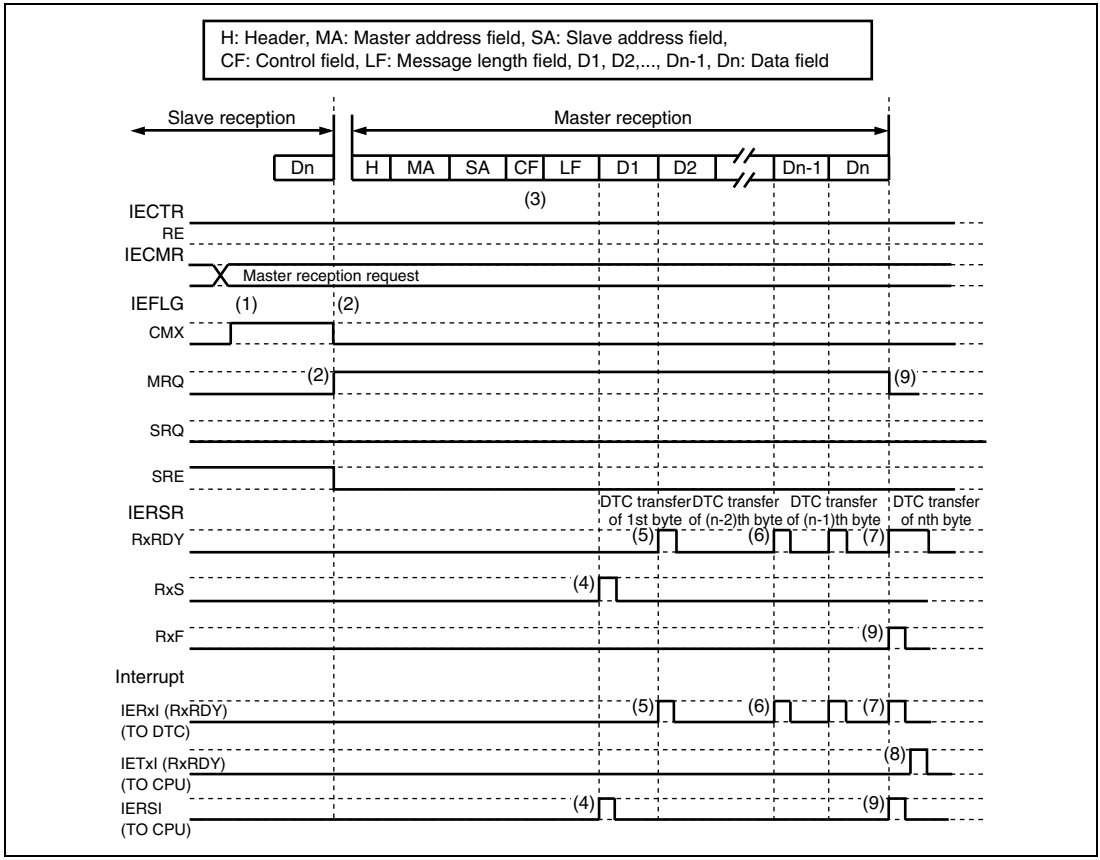


Figure 17.11 Master Receive Operation Timing

17.4.4 Slave Transmission

This section shows an example of performing a slave transmission using the DTC after slave reception.

(1) IEB Initialization

(a) Setting the IEBus Control Register (IECTR)

Enable the IEBus pins, select the signal polarity, and select a clock supplied to the IEB. Clear the LUEE bit to 0 because transfer by the DTC is performed.

(b) Setting the IEBus Master Unit Address Registers 1 and 2 (IEAR1 and IEAR2)

Specify the master unit address and specify the communications mode in IEAR1. Compare with the slave address in the communications frame and receive the frame if matched.

(c) Setting the IEBus Slave Address Setting Registers 1 and 2 (IESA1 and IESA2)

Specify the message length bits.

(d) Setting the IEBus Transmit Interrupt Enable Register (IEIET)

Enable the TxRDY (IETxI), TxS, and TxE (IETSI) interrupts.

The above registers can be specified in any order. (The register specification order does not affect the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for the DTC transfer in the vector address (H′000004D4) to be accessed a DTC transfer request is generated.
2. Set the following data from the start address of the RAM.
   - Transfer source address (SAR): Start address of the RAM which stores data to be transmitted from the data field.
   - Transfer destination address (DAR): Address (H′FFF808) of the IEBus transmit buffer register (IETBR)
   - Transfer count (CRA): The same value as IETBFL
3. Set bit DTCEG5 in the DTC enabler register G (DTCERG), and enable the TxRDY interrupt (IETxI).

   Because the TxRDY flag is retained after reset, the DTC transfer is executed when the IETxI is enabled and the first data field data is written to IETBR. The DTC negates the TxRDY flag and the DTC transfer of the first byte is completed.

(3) Slave Transmission Flow

Figure 17.12 shows the slave transmission flow. Numbers in the following description correspond to the numbers in Figure 17.12.

1. After the IEB and DTC have been initialized, a slave communications request command is issued from IECMR. During slave reception, the command execution status flag (CMX) in IEFLG is set and the slave communications request will not be issued.
2. The CMX flag is cleared when the slave reception is completed, the slave communications command is executed, and the SRQ flag is set.
3. If data up to the control field has been received correctly and if the contents of the control bits is H′3 or H′7, the transmit start detection flag (TxS) in IETSR register is set to 1. In this case, the TxS flag is cleared in the TxS interrupt handling routine.
4. The slave then transmits the message length field, and the IEB loads the transmit data in the data field from IETBR when the ACK is received. Then the TxRDY flag is set to 1. A DTC transfer request by IETxI is generated and the second byte data is written to the transmit buffer.
5. Similarly, the above data field load and transmission operations are repeated.
6. The DTC completes the data transfer for the number of specified bytes when data to be transmitted in the last byte is written to. At this time, the DTC does not clear the TxRDY flag.

It, however, clears bit DTCEG5 in the DTC enable register G (DTCERG) not to generate more DTC transfer request.

7. A TxRDY interrupt (IETxI) is issued to the CPU when the DTC transfer is completed. In this interrupt handling routine, the TxRDY flag can be cleared. However, since the TxRDY interrupt will be generated again after the last byte transfer, the TxRDY flag remains set. (Note that the LUEE bit should be cleared to 0 because an underrun error occurs to terminate the transfer if the LUEE bit in IECTR is set to 1.) Note, however, that the TxRDY interrupt should be disabled because the TxRDY interrupt is always generated.
8. After the last data transfer has been completed, a transmit normal completion (TxF) interrupt occurs. In this case, the CPU clears the TxF flag and completes the normal completion interrupt and clears the SRQ flag to 0.

Notes: 1. As a transmit status interrupt (IETSI), a transmit error termination (TxE) interrupt as well as the transmit start detection (TxS) and transmit normal completion (TxF) interrupts must be enabled. If a transmit error completion interrupt is disabled, no interrupt is generated even if the transfer is terminated by an error.

2. If the control bits sent from the master unit is H′0, H′4, H′5, or H′6 in slave transmission, the IEB automatically performs processing and the TxS and TxF flags are not set.

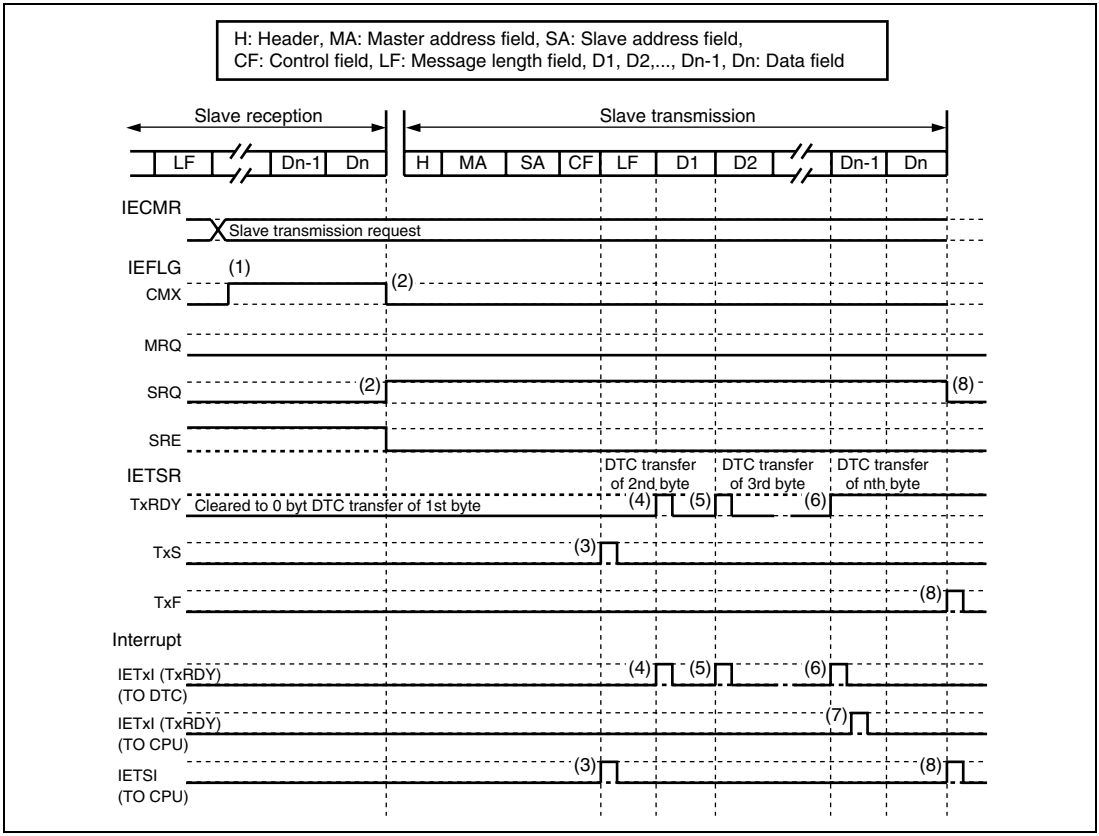
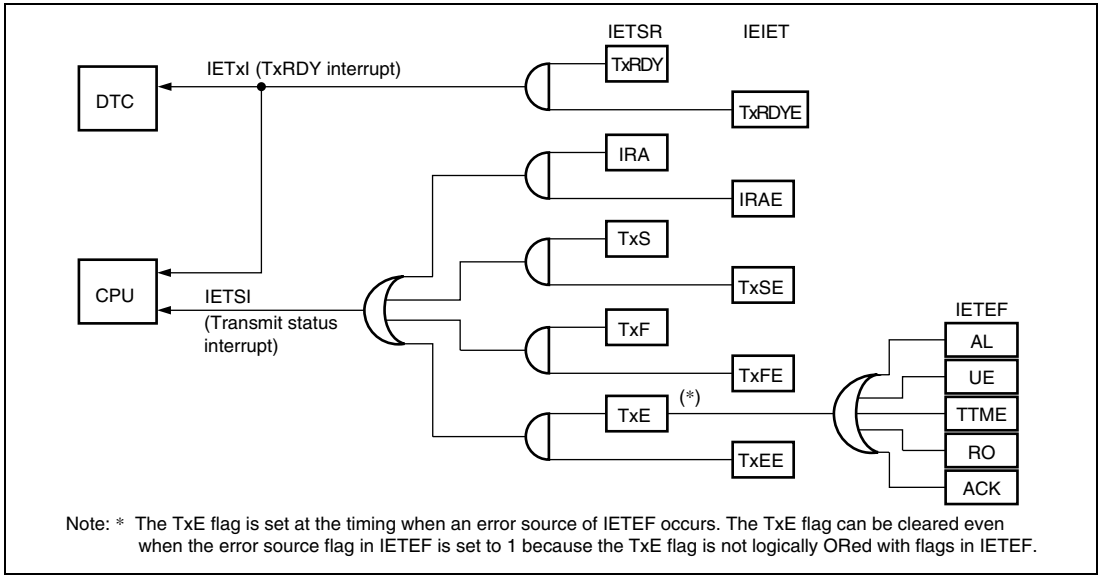


Figure 17.12 Slave Transmit Operation Timing

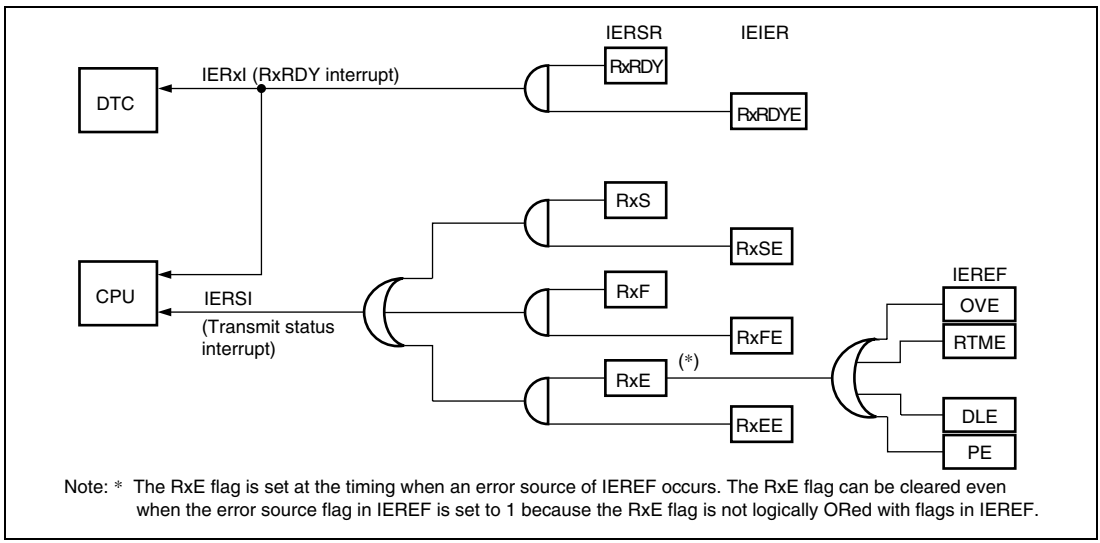
17.5 Interrupt Sources

Figures 17.13 and 17.14 show the transmit and receive interrupt sources, respectively.



Note: * The TxE flag is set at the timing when an error source of IETEF occurs. The TxE flag can be cleared even when the error source flag in IETEF is set to 1 because the TxE flag is not logically ORed with flags in IETEF.

Figure 17.13 Relationships among Transfer Interrupt Sources



Note: * The RxE flag is set at the timing when an error source of IEREF occurs. The RxE flag can be cleared even when the error source flag in IEREF is set to 1 because the RxE flag is not logically ORed with flags in IEREF.

Figure 17.14 Relationships among Receive Interrupt Sources

17.6 Usage Notes

17.6.1 TxRDY Flag and Underrun Error

1. The TxRDY flag indicates that IETBR is empty. Writing to IETBR by the DTC clears the TxRDY flag. Meanwhile, the TxRDY flag must be cleared by software since writing to IETBR by the CPU does not clear the TxRDY flag.
2. If the CPU fails to write to IETBR by the timing of the frame transmission or if the number of transfer words is less than the length specified by the message length bits, an underrun error occurs.
3. The IEB decides that an underrun error occurred when the data is loaded from IETBR to the transmit shift register while the TxRDY flag is set to 1. In this case, the IEB sets the TxE flag in IETSR and enters the wait state. The UE flag in IETEF is also set to 1.
4. On the receive side, the unit decides that a timing error has occurred because the communications are terminated.
5. In data transfer using the DTC, the TxRDY flag in IETSR is not cleared after the last byte data is transferred to IETBR and a CPU interrupt caused by the DTC interrupt will occur.

   If the TxRDY flag is not cleared in this CPU interrupt handling routine, an underrun error will occur when the last byte data is loaded from IETBR to the transmit shift register. In this case, if the LUEE bit is cleared to 0 (initial value), no underrun error occurs and the last byte of the data field is transmitted correctly. (if the LUEE bit is set to 1, an underrun error occurs.)
6. Although the DTC is used as described in item 5, if the number of DTC transfer words is less than the length specified by the message length bits, the LUEE bit setting is invalid. (The LUEE bit is valid only when data is transmitted for the number of bytes specified by the message length bits has been transmitted.) In this case, an underrun error occurs, data is transmitted for one byte less than the DTC transfer words, and the transfer is terminated by a transmit error.

17.6.2 RxRDY Flag and Overrun Error

1. The RxRDY flag indicates that IERBR stores data. Reading from IERBR by the DTC clears the RxRDY flag. Meanwhile, the RxRDY flag must be cleared by software since reading from IERBR by the CPU does not clear the RxRDY flag.
2. If the CPU fails to read from IERBR by the timing of the frame reception or if the number of transfer words is less than the length specified by the message length bits, an overrun error occurs.
3. The IEB receives data while the RxRDY flag is set and sets the OVE flag when the parity bit reception starts. If the OVE flag is set when the acknowledge bit is transmitted, the IEB assumes that an overrun error has occurred, returns a NAK, and discards the data in the receive shift register.

4. On the transmit side, the unit continues retransfer until an ACK is received because it receives a NAK.
5. If the OVE flag is cleared without loading the receive data from IERBR in the RxE interrupt handling routine caused when the OVE flag is set to 1, the IEB decides that the overrun error has been cleared and sends an ACK to other units. In this case, the transmit unit completes the communications correctly. However, no receive data is loaded from the IERBR and the receive unit continues reception. Accordingly, in an interrupt handling routine caused by the OVE flag, receive data must be loaded from IERBR, the RxRDY flag must be cleared. The DTC, thus, should be ready to receive the next byte, and then the OVE flag must be cleared.
6. Item 5 above will not occur when the DTC transfer words is specified as the IERBFL value.

17.6.3 Error Flag s in the IETEF

(1) AL Flag

The AL Flag is set to 1 when arbitration is lost even if retransfer is performed for the number of times specified by IEMCR after arbitration has been lost. The AL flag is not set when arbitration is won during retransfer. If the AL flag is set to 1, the TxE flag is set and the wait state is entered.

(2) UE Flag

If the UE flag is set to 1, the TxE flag is set and the wait state is entered. For details, see section 17.6.1, TxRDY Flag and Underrun Error.

(3) TTME Flag

If a timing error occurs during data transfer, the TTME and TxE flags are set, and the wait state is entered.

(4) RO Flag

When retransfer is performed up to the maximum number of transfer bytes defined by the protocol because of reception of a NAK from the receive side during data field transmission, the number of transferred bytes may be less than that of bytes specified by the message length. At this time the RO flag is set. Moreover, when the value of the message length bits is greater than the maximum number of transfer bytes, the RO flag is also set. The RO flag is not set if the maximum number of transfer bytes defined by the protocol is specified (for example, 32-byte message length is specified in mode 1) and the transfer is performed correctly.

If the RO flag is set to 1, the TxE flag is set to 1 and the wait state is entered.

(5) ACK Flag

- If a NAK is received in an acknowledge bit before the message length field transmission, the ACK flag is set, the TxE flag is set, and then the wait state is entered.

- If a NAK is received in an acknowledge bit of the data field, data is automatically retransmitted up to the maximum number of transfer bytes defined by the protocol. If an ACK is received in an acknowledge bit during retransfer and the following data is transmitted correctly, the ACK flag is not set. If a NAK is received in the last data transfer during the retransfer for the maximum number of transfer bytes, the ACK flag is set to 1 and the wait state is entered.

Note: Even if a NAK is received from the receive side during the data field transmission, retransfer is performed up to the maximum number of transfer bytes defined by the protocol, and the number of transferred bytes is less than that of bytes specified by the message length bits, an ACK may be received in the acknowledge bit in the last data transfer. In this case, the ACK flag is not set although the RO flag is set.

17.6.4 Error Flags in IEREF

(1) OVE Flag

When the OVE flag is set, the RxE flag is also set. If an overrun error is cleared and the OVE flag is also cleared, the IEBus receive operation is continued. For details, see section 17.6.2, RxRDY Flag and Overrun Error.

(2) RTME Flag

If a timing error occurs during data reception after reception starts (the RxS flag is set to 1), the RTME flag is set to 1, RxE flag is set to 1, and the wait state is entered. When a timing error occurs before reception starts, this flag is not set and the reception frame is discarded.

(3) DLE Flag

When retransfer is performed up to the maximum number of transfer bytes defined by the protocol because of reception of a NAK caused by a parity or an overrun error during data field reception, the number of transferred bytes may be greater than that of bytes specified by the message length. At this time the DLE flag is set. Moreover, when the value of the message length bits is greater than the maximum number of transfer bytes, the DLE flag is also set. The DLE flag is not set if the maximum number of transfer bytes defined by the protocol is specified and the transfer is performed correctly.

If the DLE flag is set to 1, the RxE flag is set to 1 and the wait state is entered.

(4) PE Flag

If a parity error occurs after reception starts (the RxS flag is set to 1), a NAK is sent to perform re-reception.

If a parity error is not cleared when the maximum number of transfer bytes specified by the protocol is received, the PE flag is set to 1, the RxE flag is set to 1 and the wait state is entered. If

a parity error is cleared during the rereception and if the following data is received correctly, the PE flag is not set.

Notes: 1. If the reception is performed up to the maximum number of transfer bytes defined by the protocol because of a parity or an overrun error during data field reception, the number of receive bytes is less than that of bytes specified by the message length bits, no parity error or overrun error may occur at the last byte reception. In this case, the DLE flag is set. However, the OVE and PE flags are not set.

2. The flags in IEREF are set after reception starts. Accordingly, the RxE flag is valid and set after the RxS flag has been set. If an error occurs before reception starts, the frame is discarded and no interrupt occurs.

17.6.5 Notes on Slave Transmission

When the slave unit transmits the slave status and upper and lower locked addresses, a parity or an overrun error occurs in the master reception side and the data cannot be received. Accordingly, even if a NAK is returned, the slave unit is not capable of retransfer.

In this case, the master unit must discard the frame in which an error occurred and request the above operation in the master reception to receive the correct frame.

17.6.6 Notes on DTC Specification

When transmit or receive data is transferred by the DTC, bit 5 (for transmission) or bit 6 (for reception) in DTCERG must be set by the bit manipulation instruction (such as BSET or BCLR). In this case, other bits (bits 7 and 4 to 0) in DTCERG must not be set to 1.

17.6.7 Error Handling in Transmission

Figure 17.15 shows the operation when a timing error occurs.

When a timing error occurs in data transmission (1), there is a possibility that the next data is already transferred to the transmit buffer by the DTC and the TxRDY flag that is the DTC initiation source is already cleared to 0 (2).

In this case, if retransfer is performed, data remained in the transmit buffer (previous frame data) is transmitted as the first byte data of the data field (3).

To avoid this error, in master transmission, the first byte data in the data field should be written to the transmit buffer by software instead of using the DTC. After that, data can be transferred by the DTC. In this case, the SAR (transfer source address) and CRA (transfer counter) should be specified as follows.

- An address of the on-chip memory that stores the second byte data → SAR
- The number of bytes specified by message length –1 → CRA

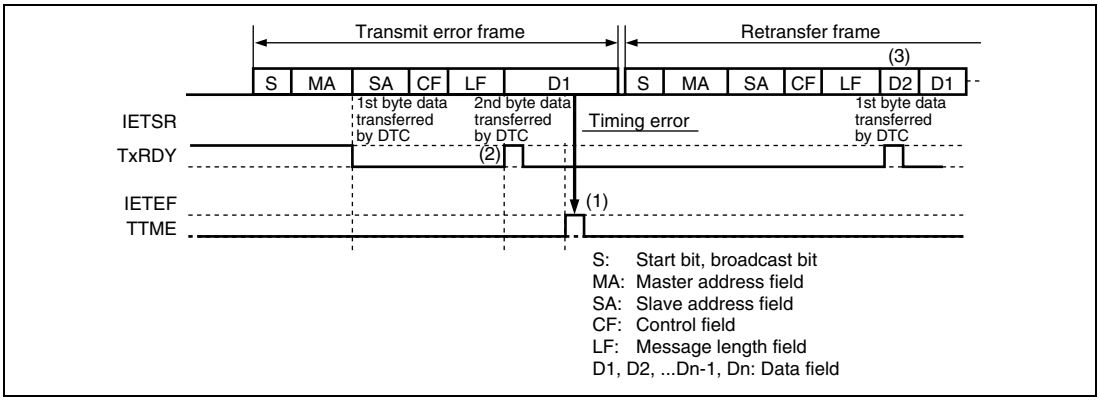


Figure 17.15 Error Processing in Transfer

17.6.8 Power-Down Mode Operation

The IEB stops operation and is initialized in power-down modes such as module stop, watch, software standby and hardware standby modes.

To initialize the IEB, the module stop mode must be specified. To reduce power consumption during IEB operation, the sleep mode must be used.

17.6.9 Notes on Middle-Speed Mode

In middle-speed mode, the IEB registers must not be read from or written to.

17.6.10 Notes on Register Access

The IEB registers can be accessed in bytes. The IEB registers must not be accessed in words or longwords.

Section 18 Hitachi Controller Area Network (HCAN) [H8S/2556 Series]

The HCAN controls a controller area network (CAN) for realtime communication in vehicular and industrial equipment systems, etc. For details on CAN specification, refer to Bosch CAN Specification Version 2.0 1991, Robert Bosch GmbH.

The block diagram of the HCAN is shown in figure 18.1.

18.1 Features

- CAN version: Conforming to Bosch 2.0B active
  Communication systems: NRZ (Non-Return to Zero) system (with bit-stuffing function)
  Broadcast communication system
  Transmission path: Bidirectional 2-wire serial communication
  Communication speed: Max. 1 Mbps
  Data length: 0 to 8 bytes
- Number of channels: 1
- Data buffers: 16 (one receive-only buffer and 15 buffers settable for transmission/reception)
- Data transmission: Two methods
  Mailbox (buffer) number order (low-to-high)
  Message priority (identifier) reverse-order (high-to-low)
- Data reception: Two methods
  Message identifier match (transmit/receive-setting buffers)
  Reception with message identifier masked (receive-only)
- CPU interrupts: 12
  Error interrupt
  Reset processing interrupt
  Message reception interrupt
  Message transmission interrupt
- HCAN operating modes
- Support for various modes
  Hardware reset
  Software reset
  Normal status (error-active, error-passive)
  Bus off status
  HCAN configuration mode
  HCAN sleep mode

HCAN halt mode

- Module stop mode can be set

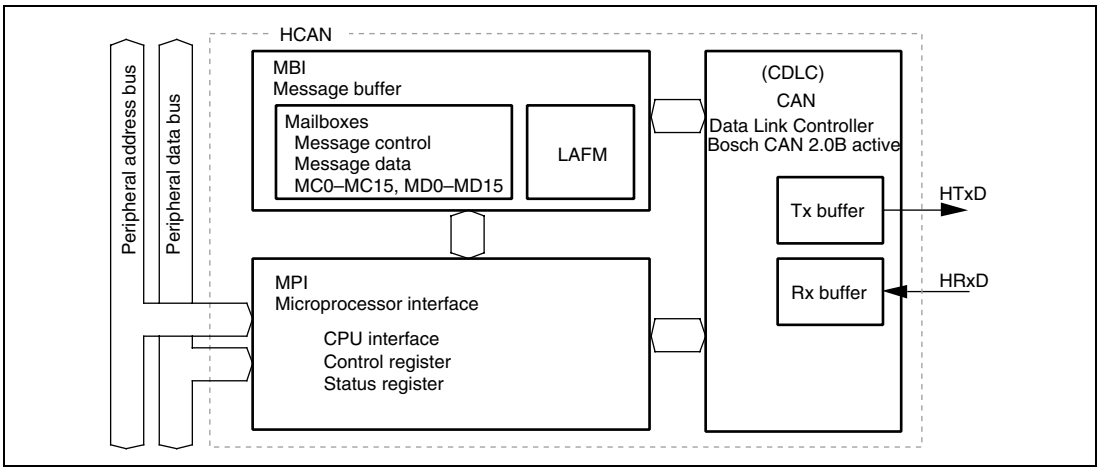


Figure 18.1 HCAN Block Diagram

- Message Buffer Interface (MBI)

  The MBI, consisting of mailboxes and a local acceptance filter mask (LAFM), stores CAN transmit/receive messages (identifiers, data, etc.) Transmit messages are written by the CPU. For receive messages, the data received by the CDLC is stored automatically.

- Microprocessor Interface (MPI)

  The MPI, consisting of a bus interface, control register, status register, etc., controls HCAN internal data, status, and so forth.

- CAN Data Link Controller (CDLC)

  The CDLC transmits and receives of messages conforming to the Bosch CAN Ver. 2.0B active standard (data frames, remote frames, error frames, overload frames, inter-frame spacing), as well as CRC checking, bus arbitration, and other functions.

18.2 Input/Output Pins

Table 18.1 shows the HCAN's pins.

When using HCAN pins, settings must be made in the HCAN configuration mode (during initialization: MCR0 = 1 and GSR3 = 1).

Table 18.1 HCAN Pins

Name	Abbreviation	Input/Output	Function
HCAN transmit data pin	HTxD	Output	CAN bus transmission pin
HCAN receive data pin	HRxD	Input	CAN bus reception pin

A bus driver is necessary for the interface between the pins and the CAN bus. A Philips PCA82C250 compatible model is recommended.

18.3 Register Descriptions

The HCAN has the following registers.

- Master control register (MCR)
- General status register (GSR)
- Bit configuration register (BCR)
- Mailbox configuration register (MBCR)
- Transmit wait register (TXPR)
- Transmit wait cancel register (TXCR)
- Transmit acknowledge register (TXACK)
- Abort acknowledge register (ABACK)
- Receive complete register (RXPR)
- Remote request register (RFPR)
- Interrupt register (IRR)
- Mailbox interrupt mask register (MBIMR)
- Interrupt mask register (IMR)
- Receive error counter (REC)
- Transmit error counter (TEC)
- Unread message status register (UMSR)
- Local acceptance filter mask H (LAFMH)
- Local acceptance filter mask L (LAFML)
- Message control (8-bit × 8 registers × 16 sets) (MC0 to MC15)
- Message data (8-bit × 8 registers × 16 sets) (MD0 to MD15)

18.3.1 Master Control Register (MCR)

MCR controls the HCAN.

Bit	Bit Name	Initial Value	R/W	Description
7	MCR7	0	R/W	HCAN Sleep Mode Release When this bit is set to 1, the HCAN automatically exits HCAN sleep mode on detection of CAN bus operation.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	MCR5	0	R/W	HCAN Sleep Mode When this bit is set to 1, the HCAN transits to HCAN sleep mode. When this bit is cleared to 0, HCAN sleep mode is released.
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	MCR2	0	R/W	Message Transmission Method 0: Transmission order determined by message identifier priority 1: Transmission order determined by mailbox number priority (TXPR1 > TXPR15)
1	MCR1	0	R/W	Halt Request When this bit is set to 1, the HCAN transits to HCAN HALT mode. When this bit is cleared to 0, HCAN HALT mode is released.
0	MCR0	1	R/W	Reset Request When this bit is set to 1, the HCAN transits to reset mode. For details, refer to section 18.4.1, Hardware and Software Reset. [Setting conditions] • Power-on reset • Hardware standby • Software standby • Watch mode • Module stop mode • 1-write (software reset) [Clearing condition] • When 0 is written to this bit while the GSR3 bit in GSR is 1

18.3.2 General Status Register (GSR)

GSR indicates the status of the HCAN.

Bit	Bit Name	Initial Value	R/W	Description
7 to4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	GSR3	1	R	Reset Status Bit Indicates whether the HCAN module is in the normal operating state or the reset state. This bit cannot be modified. [Setting conditions] • When entering configuration mode after the HCAN internal reset has finished • Sleep mode [Clearing condition] • When entering normal operation mode after the MCR0 bit in MCR is cleared to 0 (Note that there is a delay between clearing of the MCR0 bit and the GSR3 bit.)
2	GSR2	1	R	Message Transmission Status Flag Flag that indicates whether the module is currently in the message transmission period. This bit cannot be modified. [Setting condition] • Start of message transmission (SOF) [Clearing condition] • Interval of three bits after EOF (End of Frame)
1	GSR1	0	R	Transmit/Receive Warning Flag This bit cannot be modified. [Clearing conditions] • When TEC < 96 and REC < 96 • When TEC ≥ 256 [Setting condition] • When TEC ≥ 96 or REC ≥ 96

Bit	Bit Name	Initial Value	R/W	Description
0	GSR0	0	R	Bus Off Flag This bit cannot be modified. [Setting condition] • When TEC ≥ 256 (bus off state) [Clearing condition] • Recovery from bus off state

18.3.3 Bit Configuration Register (BCR)

BCR is used to set HCAN bit timing parameters and the baud rate prescaler. For details on parameters, refer to section 18.4.2, Initialization after Hardware Reset.

Bit	Bit Name	Initial Value	R/W	Description
15	BCR7	0	R/W	Re-Synchronization Jump Width (SJW)
14	BCR6	0	R/W	Set the maximum bit synchronization width. 00: 1 time quantum 01: 2 time quanta 10: 3 time quanta 11: 4 time quanta
13	BCR5	0	R/W	Baud Rate Prescaler (BRP)
12	BCR4	0	R/W	Set the length of time quantum.
11	BCR3	0	R/W	000000: 2 × system clock
10	BCR2	0	R/W	000001: 4 × system clock
9	BCR1	0	R/W	000010: 6 × system clock
8	BCR0	0	R/W	: 111111: 128 × system clock
7	BCR15	0	R/W	Bit Sample Point (BSP) Sets the point at which data is sampled. 0: Bit sampling at one point (end of time segment 1 (TSEG1)) 1: Bit sampling at three points (end of TSEG1 and preceding and following one time quantum)

Bit	Bit Name	Initial Value	R/W	Description
6	BCR14	0	R/W	Time Segment 2 (TSEG2)
5	BCR13	0	R/W	Set the TSEG2 width within a range of 2 to 8 time quanta.
4	BCR12	0	R/W	000: Setting prohibited
				001: 2 time quanta
				010: 3 time quanta
				011: 4 time quanta
				100: 5 time quanta
				101: 6 time quanta
				110: 7 time quanta
				111: 8 time quanta
3	BCR11	0	R/W	Time Segment 1 (TSEG1)
2	BCR10	0	R/W	Set the TSEG1 (PRSEG + PHSEG1) width to between 4 and 16 time quanta.
1	BCR9	0	R/W	
0	BCR8	0	R/W	0000: Setting prohibited
				0001: Setting prohibited
				0010: Setting prohibited
				0011: 4 time quanta
				0100: 5 time quanta
				0101: 6 time quanta
				0110: 7 time quanta
				0111: 8 time quanta
				1000: 9 time quanta
				1001: 10 time quanta
				1010: 11 time quanta
				1011: 12 time quanta
				1100: 13 time quanta
				1101: 14 time quanta
				1110: 15 time quanta
				1111: 16 time quanta

18.3.4 Mailbox Configuration Register (MBCR)

MBCR is used to set the transfer direction for each mailbox.

Bit	Bit Name	Initial Value	R/W	Description
15	MBCR7	0	R/W	These bits set the transfer direction for the corresponding mailboxes from 1 to 15. MBCRn determines the transfer direction for mailbox n (n =1 to 15). 0: Corresponding mailbox is set for transmission 1: Corresponding mailbox is set for reception Bit 8 is reserved. This bit is always read as 1 and the write value should always be 1.
14	MBCR6	0	R/W	
13	MBCR5	0	R/W	
12	MBCR4	0	R/W	
11	MBCR3	0	R/W	
10	MBCR2	0	R/W	
9	MBCR1	0	R/W	
8	—	1	R	
7	MBCR15	0	R/W	
6	MBCR14	0	R/W	
5	MBCR13	0	R/W	
4	MBCR12	0	R/W	
3	MBCR11	0	R/W	
2	MBCR10	0	R/W	
1	MBCR9	0	R/W	
0	MBCR8	0	R/W	

18.3.5 Transmit Wait Register (TXPR)

TXPR is used to set a transmit wait after a transmit message is stored in a mailbox (buffer) (CAN bus arbitration wait).

Bit	Bit Name	Initial Value	R/W	Description
15	TXPR7	0	R/W	These bits set a transmit wait (CAN bus arbitration wait) for the corresponding mailboxes 1 to 15. When TXPRn (n = 1 to 15) is set to 1, the message in mailbox n becomes the transmit wait state. [Clearing conditions] • Completion of message transmission • Completion of transmission cancellation
14	TXPR6	0	R/W	
13	TXPR5	0	R/W	
12	TXPR4	0	R/W	
11	TXPR3	0	R/W	
10	TXPR2	0	R/W	
9	TXPR1	0	R/W	
8	—	0	R	Bit 8 is reserved. This bit is always read as 0 and the write value should always be 0.
7	TXPR15	0	R/W	
6	TXPR14	0	R/W	
5	TXPR13	0	R/W	
4	TXPR12	0	R/W	
3	TXPR11	0	R/W	
2	TXPR10	0	R/W	
1	TXPR9	0	R/W	
0	TXPR8	0	R/W	

18.3.6 Transmit Wait Cancel Register (TXCR)

TXCR controls the cancellation of transmit wait messages in mailboxes (buffers).

Bit	Bit Name	Initial Value	R/W	Description
15	TXCR7	0	R/W	These bits cancel the transmit wait message in the corresponding mailboxes 1 to 15. When TXCRn (n = 1 to 15) is set to 1, the transmit wait message in mailbox n is canceled. [Clearing condition] • Completion of TXPR clearing when transmit message is canceled normally
14	TXCR6	0	R/W	
13	TXCR5	0	R/W	
12	TXCR4	0	R/W	
11	TXCR3	0	R/W	
10	TXCR2	0	R/W	
9	TXCR1	0	R/W	
8	—	0	R	Bit 8 is reserved. This bit is always read as 0 and the write value should always be 0.
7	TXCR15	0	R/W	
6	TXCR14	0	R/W	
5	TXCR13	0	R/W	
4	TXCR12	0	R/W	
3	TXCR11	0	R/W	
2	TXCR10	0	R/W	
1	TXCR9	0	R/W	
0	TXCR8	0	R/W	

18.3.7 Transmit Acknowledge Register (TXACK)

TXACK is a status register that indicates the normal transmission of mailbox (buffer) transmit messages.

Bit	Bit Name	Initial Value	R/W	Description
15	TXACK7	0	R/(W)*	These bits are status flags that indicate error-free transmission of the transmit message in the corresponding mailboxes 1 to 15. When the message in mailbox n (n = 1 to 15) has been transmitted error-free, TXACKn is set to 1. [Setting condition] • Completion of message transmission for corresponding mailbox [Clearing condition] • Writing 1
14	TXACK6	0	R/(W)*	
13	TXACK5	0	R/(W)*	
12	TXACK4	0	R/(W)*	
11	TXACK3	0	R/(W)*	
10	TXACK2	0	R/(W)*	
9	TXACK1	0	R/(W)*	
8	—	0	R	Bit 8 is reserved. This bit is always read as 0 and the write value should always be 0.
7	TXACK15	0	R/(W)*	
6	TXACK14	0	R/(W)*	
5	TXACK13	0	R/(W)*	
4	TXACK12	0	R/(W)*	
3	TXACK11	0	R/(W)*	
2	TXACK10	0	R/(W)*	
1	TXACK9	0	R/(W)*	
0	TXACK8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

18.3.8 Abort Acknowledge Register (ABACK)

ABACK is a status register that indicates the normal cancellation (aborting) of mailbox (buffer) transmit messages.

Bit	Bit Name	Initial Value	R/W	Description
15	ABACK7	0	R/(W)*	These bits are status flags that indicate error-free cancellation (abortion) of the transmit message in the corresponding mailboxes 1 to 15. When the message in mailbox n (n = 1 to 15) has been canceled error-free, ABACKn is set to 1. [Setting condition] • Completion of transmit message cancellation for corresponding mailbox [Clearing condition] • Writing 1
14	ABACK6	0	R/(W)*	
13	ABACK5	0	R/(W)*	
12	ABACK4	0	R/(W)*	
11	ABACK3	0	R/(W)*	
10	ABACK2	0	R/(W)*	
9	ABACK1	0	R/(W)*	
8	—	0	R	Bit 8 is reserved. This bit is always read as 0. The write value should always be 0.
7	ABACK15	0	R/(W)*	
6	ABACK14	0	R/(W)*	
5	ABACK13	0	R/(W)*	
4	ABACK12	0	R/(W)*	
3	ABACK11	0	R/(W)*	
2	ABACK10	0	R/(W)*	
1	ABACK9	0	R/(W)*	
0	ABACK8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

18.3.9 Receive Complete Register (RXPR)

RXPR is a status register that indicates the normal reception of messages (data frame or remote frame) in mailboxes. For reception of a remote frame, when a bit in this register is set to 1, the corresponding remote request register (RFPR) bit is also set to 1 simultaneously.

Bit	Bit Name	Initial Value	R/W	Description
15	RXPR7	0	R/(W)*	When the message in mailbox n (n = 0 to 15) has been received error-free, RXPRn is set to 1. [Setting condition] • Completion of message (data frame or remote frame) reception in corresponding mailbox [Clearing condition] • Writing 1
14	RXPR6	0	R/(W)*	
13	RXPR5	0	R/(W)*	
12	RXPR4	0	R/(W)*	
11	RXPR3	0	R/(W)*	
10	RXPR2	0	R/(W)*	
9	RXPR1	0	R/(W)*	
8	RXPR0	0	R/(W)*	
7	RXPR15	0	R/(W)*	
6	RXPR14	0	R/(W)*	
5	RXPR13	0	R/(W)*	
4	RXPR12	0	R/(W)*	
3	RXPR11	0	R/(W)*	
2	RXPR10	0	R/(W)*	
1	RXPR9	0	R/(W)*	
0	RXPR8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

18.3.10 Remote Request Register (RFPR)

RFPR is a status register that indicates normal reception of remote frames in mailboxes (buffers). When a bit in this register is set to 1, the corresponding receive complete register (RXPR) bit is also set to 1 simultaneously.

Bit	Bit Name	Initial Value	R/W	Description
15	RFPR7	0	R/(W)*	When mailbox n (n = 0 to 15) has received the remote frame error-free, RFPRn (n = 0 to 15) is set to 1. [Setting condition] • Completion of remote frame reception in corresponding mailbox [Clearing condition] • Writing 1
14	RFPR6	0	R/(W)*	
13	RFPR5	0	R/(W)*	
12	RFPR4	0	R/(W)*	
11	RFPR3	0	R/(W)*	
10	RFPR2	0	R/(W)*	
9	RFPR1	0	R/(W)*	
8	RFPR0	0	R/(W)*	
7	RFPR15	0	R/(W)*	
6	RFPR14	0	R/(W)*	
5	RFPR13	0	R/(W)*	
4	RFPR12	0	R/(W)*	
3	RFPR11	0	R/(W)*	
2	RFPR10	0	R/(W)*	
1	RFPR9	0	R/(W)*	
0	RFPR8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

18.3.11 Interrupt Register (IRR)

IRR is an interrupt status flag register.

Bit	Bit Name	Initial Value	R/W	Description
15	IRR7	0	R/(W)*	Overload Frame/Buss OFF Recovery Interrupt Flag [Setting conditions] • When an overload frame is transmitted in error active/passive state • When eleven recessive bits are received 128 times in buss off state [Clearing condition] • Writing 1
14	IRR6	0	R/(W)*	Bus Off Interrupt Flag Status flag indicating the bus off state caused by the transmit error counter. [Setting condition] • When TEC ≥ 256 [Clearing condition] • Writing 1
13	IRR5	0	R/(W)*	Error Passive Interrupt Flag Status flag indicating the error passive state caused by the transmit/receive error counter. [Setting condition] When TEC ≥ 128 or REC ≥ 128 [Clearing condition] • Writing 1
12	IRR4	0	R/(W)*	Receive Overload Warning Interrupt Flag Status flag indicating the error warning state caused by the receive error counter. [Setting condition] When REC ≥ 96 [Clearing condition] • Writing 1

Bit	Bit Name	Initial Value	R/W	Description
11	IRR3	0	R/(W)*	Transmit Overload Warning Interrupt Flag Status flag indicating the error warning state caused by the transmit error counter. [Setting condition] • When TEC ≥ 96 [Clearing condition] • Writing 1
10	IRR2	0	R	Remote Frame Request Interrupt Flag Status flag indicating that a remote frame has been received in a mailbox. [Setting condition] • When remote frame reception is completed, when corresponding MBIMR = 0 [Clearing condition] • Clearing of all bits in RFPR (remote request register)
9	IRR1	0	R	Receive Message Interrupt Flag Status flag indicating that a mailbox receive message has been received normally. [Setting condition] • When data frame or remote frame reception is completed, when corresponding MBIMR = 0 [Clearing condition] • Clearing of all bits in RXPR (receive complete register)
8	IRR0	1	R/(W)*	Reset Interrupt Flag Status flag indicating that the HCAN module has been reset. This bit cannot be masked by the interrupt mask register (IMR). If this bit is not cleared to 0 after entering power-on reset or returning from software standby mode, watch mode, or module stop mode, interrupt processing will start immediately when the interrupt controller enables interrupts. [Setting condition] • When the reset operation has finished after entering power-on reset or software standby mode, watch mode, or module stop mode. [Clearing condition] • Writing 1

Bit	Bit Name	Initial Value	R/W	Description
7 to5	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
4	IRR12	0	R/(W)*	Bus Operation Interrupt Flag Status flag indicating detection of a dominant bit due to bus operation when the HCAN module is in HCAN sleep mode. [Setting condition] • Bus operation (dominant bit) detection in HCAN sleep mode [Clearing condition] • Writing 1
3, 2	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
1	IRR9	0	R	Unread Interrupt Flag Status flag indicating that a receive message has been overwritten before being read. [Setting condition] • When UMSR (unread message status register) is set [Clearing condition] • Clearing of all bits in UMSR (unread message status register)
0	IRR8	0	R/(W)*	Mailbox Empty Interrupt Flag Status flag indicating that the next transmit message can be stored in the mailbox. [Setting condition] • When TXPR (transmit wait register) is cleared by completion of transmission or completion of transmission abort [Clearing condition] • Writing 1

Note: * Only 1 can be written to this bit for clearing the flag.

18.3.12 Mailbox Interrupt Mask Register (MBIMR)

MBIMR controls the enabling or disabling of individual mailbox interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
15	MBIMR7	1	R/W	Mailbox Interrupt Mask (MBIMRx)
14	MBIMR6	1	R/W	When MBIMRn (n = 0 to 15) is cleared to 0, the interrupt request in mailbox n is enabled. When set to 1, the interrupt request is masked.
13	MBIMR5	1	R/W	
12	MBIMR4	1	R/W	The interrupt source in a transmit mailbox is TXPR clearing caused by transmission end or transmission cancellation. The interrupt source in a receive mailbox is RXPR setting on reception end.
11	MBIMR3	1	R/W	
10	MBIMR2	1	R/W	
9	MBIMR1	1	R/W	
8	MBIMR0	1	R/W	
7	MBIMR15	1	R/W	
6	MBIMR14	1	R/W	
5	MBIMR13	1	R/W	
4	MBIMR12	1	R/W	
3	MBIMR11	1	R/W	
2	MBIMR10	1	R/W	
1	MBIMR9	1	R/W	
0	MBIMR8	1	R/W	

18.3.13 Interrupt Mask Register (IMR)

IMR enables or disables requests by individual interrupt sources of IRR. The interrupt flag cannot be masked.

Bit	Bit Name	Initial Value	R/W	Description
15	IMR7	1	R/W	Overload Frame/Bus Off Recovery Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR7) is enabled. When set to 1, OVR0 is masked.
14	IMR6	1	R/W	Bus Off Interrupt Mask When this bit is cleared to 0, ERS0 (interrupt request by IRR6) is enabled. When set to 1, ERS0 is masked.
13	IMR5	1	R/W	Error Passive Interrupt Mask When this bit is cleared to 0, ERS0 (interrupt request by IRR5) is enabled. When set to 1, ERS0 is masked.

Bit	Bit Name	Initial Value	R/W	Description
12	IMR4	1	R/W	Receive Overload Warning Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR4) is enabled. When set to 1, OVR0 is masked.
11	IMR3	1	R/W	Transmit Overload Warning Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR3) is enabled. When set to 1, OVR0 is masked.
10	IMR2	1	R/W	Remote Frame Request Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR2) is enabled. When set to 1, OVR0is masked.
9	IMR1	1	R/W	Receive Message Interrupt Mask When this bit is cleared to 0, RM1 (interrupt request by IRR1) is enabled. When set to 1, RMI is masked.
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7 to 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
4	IMR12	1	R/W	Bus Operation Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR12) is enabled. When set to 1, OVR0 is masked.
3, 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	IMR9	1	R/W	Unread Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt request by IRR9) is enabled. When set to 1, OVR0 is masked.
0	IMR8	1	R/W	Mailbox Empty Interrupt Mask When this bit is cleared to 0, SLE0 (interrupt request by IRR8) is enabled. When set to 1, SLE0 is masked.

18.3.14 Receive Error Counter (REC)

REC is an 8-bit read-only register that functions as a counter indicating the number of receive message errors on the CAN bus. The count value is stipulated in the CAN protocol.

18.3.15 Transmit Error Counter (TEC)

TEC is an 8-bit read-only register that functions as a counter indicating the number of transmit message errors on the CAN bus. The count value is stipulated in the CAN protocol.

18.3.16 Unread Message Status Register (UMSR)

UMSR is a status register that indicates, for individual mailboxes, that a received message has been overwritten by a new receive message before being read. When overwritten by a new message, data in the unread receive message is lost.

Bit	Bit Name	Initial Value	R/W	Description
15	UMSR7	0	R/W	Indicates that a received massage has been overwritten by a new message before being read. [Setting condition] When a new message is received before RXPR is cleared [Clearing condition] Writing 1
14	UMSR6	0	R/W	
13	UMSR5	0	R/W	
12	UMSR4	0	R/W	
11	UMSR3	0	R/W	
10	UMSR2	0	R/W	
9	UMSR1	0	R/W	
8	UMSR0	0	R/W	
7	UMSR15	0	R/W	
6	UMSR14	0	R/W	
5	UMSR13	0	R/W	
4	UMSR12	0	R/W	
3	UMSR11	0	R/W	
2	UMSR10	0	R/W	
1	UMSR9	0	R/W	
0	UMSR8	0	R/W	

18.3.17 Local Acceptance Filter Masks (LAFML, LAFMH)

LAFML and LAFMH individually set the identifier bits of the message to be stored in mailbox 0 as Don't Care. For details, refer to section 18.4.4, Massage Reception. The relationship between the identifier bits and mask bits are shown in the following.

- LAFML

Bit	Bit Name	Initial Value	R/W	Description
15	LAFML7	0	R/W	When this bit is set to 1, ID-7 of the receive message identifier is not compared.
14	LAFML6	0	R/W	When this bit is set to 1, ID-6 of the receive message identifier is not compared.
13	LAFML5	0	R/W	When this bit is set to 1, ID-5 of the receive message identifier is not compared.
12	LAFML4	0	R/W	When this bit is set to 1, ID-4 of the receive message identifier is not compared.
11	LAFML3	0	R/W	When this bit is set to 1, ID-3 of the receive message identifier is not compared.
10	LAFML2	0	R/W	When this bit is set to 1, ID-2 of the receive message identifier is not compared.
9	LAFML1	0	R/W	When this bit is set to 1, ID-1 of the receive message identifier is not compared.
8	LAFML0	0	R/W	When this bit is set to 1, ID-0 of the receive message identifier is not compared.
7	LAFML15	0	R/W	When this bit is set to 1, ID-15 of the receive message identifier is not compared.
6	LAFML14	0	R/W	When this bit is set to 1, ID-14 of the receive message identifier is not compared.
5	LAFML13	0	R/W	When this bit is set to 1, ID-13 of the receive message identifier is not compared.
4	LAFML12	0	R/W	When this bit is set to 1, ID-12 of the receive message identifier is not compared.
3	LAFML11	0	R/W	When this bit is set to 1, ID-11 of the receive message identifier is not compared.
2	LAFML10	0	R/W	When this bit is set to 1, ID-10 of the receive message identifier is not compared.
1	LAFML9	0	R/W	When this bit is set to 1, ID-9 of the receive message identifier is not compared.
0	LAFML8	0	R/W	When this bit is set to 1, ID-8 of the receive message identifier is not compared.

- LAFMH

Bit	Bit Name	Initial Value	R/W	Description
15	LAFMH7	0	R/W	When this bit is set to 1, ID-20 of the receive message identifier is not compared.
14	LAFMH6	0	R/W	When this bit is set to 1, ID-19 of the receive message identifier is not compared.
13	LAFMH5	0	R/W	When this bit is set to 1, ID-18 of the receive message identifier is not compared.
12 to10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	LAFMH1	0	R/W	When this bit is set to 1, ID-17 of the receive message identifier is not compared.
8	LAFMH0	0	R/W	When this bit is set to 1, ID-16 of the receive message identifier is not compared.
7	LAFMH15	0	R/W	When this bit is set to 1, ID-28 of the receive message identifier is not compared.
6	LAFMH14	0	R/W	When this bit is set to 1, ID-27 of the receive message identifier is not compared.
5	LAFMH13	0	R/W	When this bit is set to 1, ID-26 of the receive message identifier is not compared.
4	LAFMH12	0	R/W	When this bit is set to 1, ID-25 of the receive message identifier is not compared.
3	LAFMH11	0	R/W	When this bit is set to 1, ID-24 of the receive message identifier is not compared.
2	LAFMH10	0	R/W	When this bit is set to 1, ID-23 of the receive message identifier is not compared.
1	LAFMH9	0	R/W	When this bit is set to 1, ID-22 of the receive message identifier is not compared.
0	LAFMH8	0	R/W	When this bit is set to 1, ID-21 of the receive message identifier is not compared.

18.3.18 Message Control (MC0 to MC15)

The message control register sets consist of eight 8-bit registers for one mailbox. The HCAN has 16 sets of these registers. Because message control registers are in RAM, their initial values after power-on are undefined. Be sure to initialize them by writing 0 or 1. Figure 18.2 shows the register names for each mailbox.

Mail box 0	MC0[1]	MC0[2]	MC0[3]	MC0[4]	MC0[5]	MC0[6]	MC0[7]	MC0[8]
Mail box 1	MC1[1]	MC1[2]	MC1[3]	MC1[4]	MC1[5]	MC1[6]	MC1[7]	MC1[8]
Mail box 2	MC2[1]	MC2[2]	MC2[3]	MC2[4]	MC2[5]	MC2[6]	MC2[7]	MC2[8]
Mail box 3	MC3[1]	MC3[2]	MC3[3]	MC3[4]	MC3[5]	MC3[6]	MC3[7]	MC3[8]
Mail box 15	MC15[1]	MC15[2]	MC15[3]	MC15[4]	MC15[5]	MC15[6]	MC15[7]	MC15[8]

Figure 18.2 Message Control Register Configuration

The setting of message control registers are shown in the following. Figures 18.3 and 18.4 show the correspondence between the identifiers and register bit names.

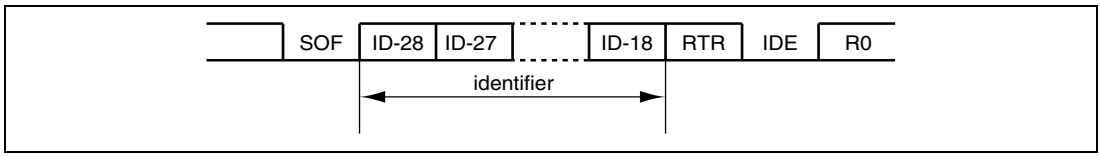


Figure 18.3 Standard Format

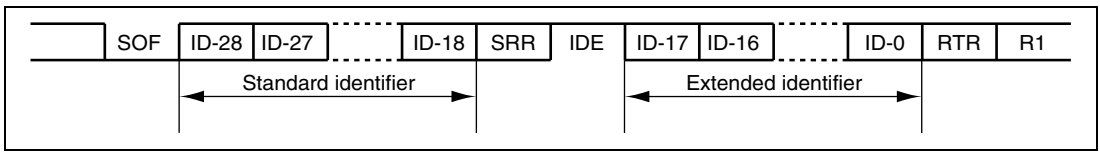


Figure 18.4 Extended Format

Register Name	Bit	Bit Name	R/W	Description
MCx[1]	7 to 4	—	R/W	The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).
	3 to 0	DLC3 to DLC0	R/W	Data Length Code Set the data length of a data frame or the data length requested in a remote frame within the range of 0 to 8 bytes. 0000: 0 byte 0001: 1 byte 0010: 2 bytes 0011: 3 bytes 0100: 4 bytes 0101: 5 bytes 0110: 6 bytes 0111: 7 bytes 1000: 8 bytes : : 1111: 8 bytes
MCx[2]	7 to 0	—	R/W	The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).
MCx[3]	7 to 0	—	R/W	
MCx[4]	7 to 0	—	R/W	
MCx[5]	7 to 5	ID-20 to ID-18	R/W	Sets ID-20 to ID-18 in the identifier.
	4	RTR	R/W	Remote Transmission Request Used to distinguish between data frames and remote frames. 0: Data frame 1: Remote frame
	3	IDE	R/W	Identifier Extension Used to distinguish between the standard format and extended format of data frames and remote frames. 0: Standard format 1: Extended format
	2	—	R/W	The initial value of this bit is undefined. It must be initialized by writing 0 or 1.
	1 to 0	ID-17 to ID-16	R/W	Sets ID-17 and ID-16 in the identifier.
MCx[6]	7 to 0	ID-28 to ID-21	R/W	Sets ID-28 to ID-21 in the identifier.
MCx[7]	7 to 0	ID-7 to ID-0	R/W	Sets ID-7 to ID-0 in the identifier.
MCx[8]	7 to 0	ID-15 to ID-8	R/W	Sets ID-15 to ID-8 in the identifier.

Note: x: Mailbox number

18.3.19 Message Data (MD0 to MD15)

The message data register sets consist of eight 8-bit registers for one mailbox. The HCAN has 16 sets of these registers. Because message data registers are in RAM, their initial values after power-on are undefined. Be sure to initialize them by writing 0 or 1. Figure 18.5 shows the register names for each mailbox.

Mail box 0	MD0[1]	MD0[2]	MD0[3]	MD0[4]	MD0[5]	MD0[6]	MD0[7]	MD0[8]
Mail box 1	MD1[1]	MD1[2]	MD1[3]	MD1[4]	MD1[5]	MD1[6]	MD1[7]	MD1[8]
Mail box 2	MD2[1]	MD2[2]	MD2[3]	MD2[4]	MD2[5]	MD2[6]	MD2[7]	MD2[8]
Mail box 3	MD3[1]	MD3[2]	MD3[3]	MD3[4]	MD3[5]	MD3[6]	MD3[7]	MD3[8]
Mail box 15	MD15[1]	MD15[2]	MD15[3]	MD15[4]	MD15[5]	MD15[6]	MD15[7]	MD15[8]

Figure 18.5 Message Data Configuration

18.4 Operation

18.4.1 Hardware and Software Resets

The HCAN can be reset by a hardware reset or software reset.

- Hardware Reset

  At power-on reset, or in hardware standby mode, software standby mode, watch mode, or module stop mode, the HCAN is initialized by automatically setting the MCR reset request bit (MCR0) in MCR and the reset state bit (GSR3) in GSR. At the same time, all internal registers, except for message control and message data registers, are initialized by a hardware reset.
- Software Reset

  The HCAN can be reset by setting the MCR reset request bit (MCR0) in MCR via software. In a software reset, the error counters (TEC and REC) are initialized, however other registers are not. If the MCR0 bit is set while the CAN controller is performing a communication operation (transmission or reception), the initialization state is not entered until message transfer has been completed. The reset status bit (GSR3) in GSR is set on completion of initialization.

18.4.2 Initialization after Hardware Reset

After a hardware reset, the following initialization processing should be carried out:

1. Clearing of IRR0 bit in the interrupt register (IRR)
2. Bit rate setting

3. Mailbox transmit/receive settings
4. Mailbox (RAM) initialization
5. Message transmission method setting

These initial settings must be made while the HCAN is in bit configuration mode. Configuration mode is a state in which the GSR3 bit in GSR is set to 1 by a reset. Configuration mode is exited by clearing the MCR0 bit in MCR to 0; when the MCR0 bit is cleared to 0, the HCAN automatically clears the GSR3 bit in GSR. There is a delay between clearing the MCR0 bit and clearing the GSR3 bit because the HCAN needs time to be internally reset, there is a delay between clearing of the MCR0 bit and GSR3 bit. After the HCAN exits configuration mode, the power-up sequence begins, and communication with the CAN bus is possible as soon as 11 consecutive recessive bits have been detected.

IRR0 Clearing: The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode and watch mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.

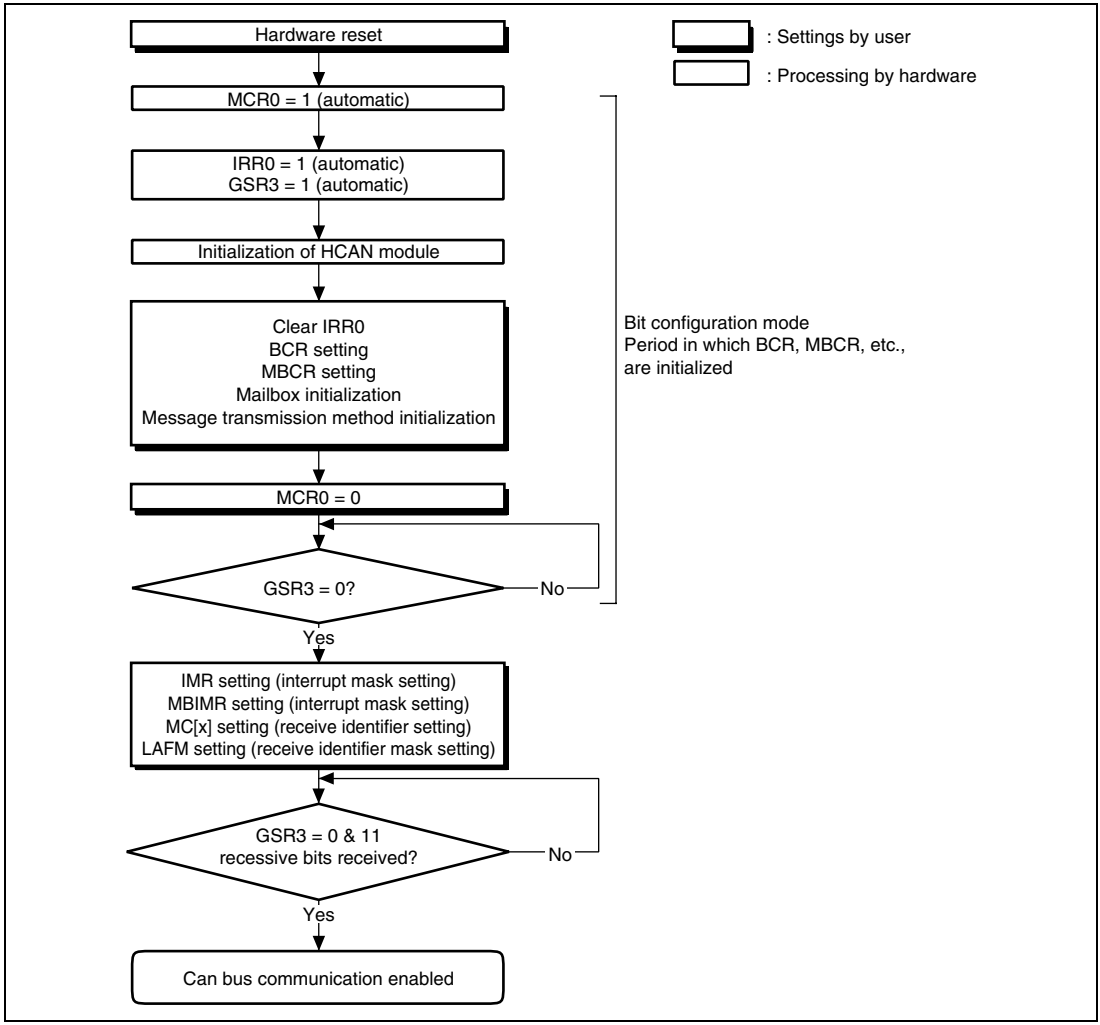


Figure 18.6 Hardware Reset Flowchart

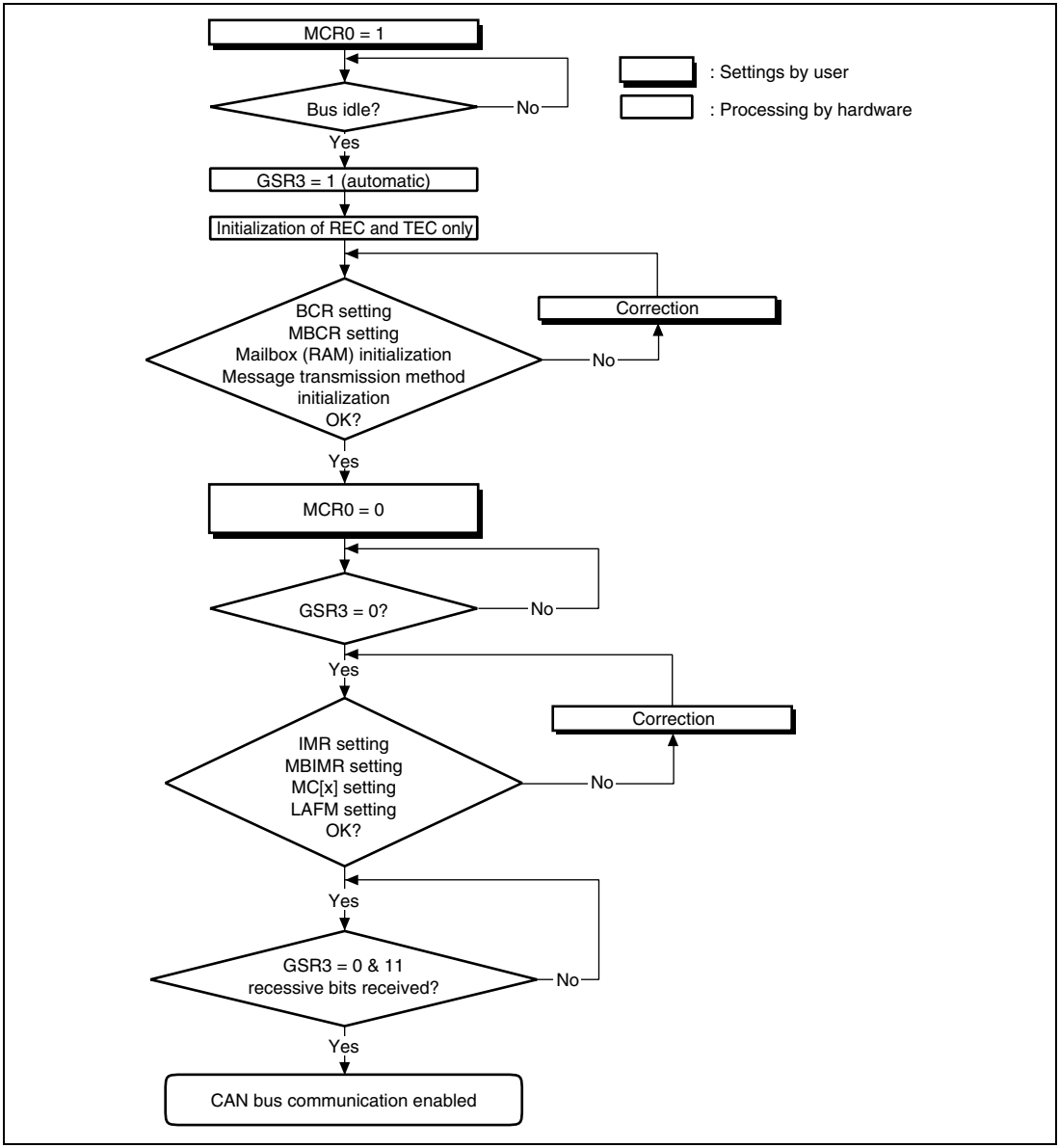


Figure 18.7 Software Reset Flowchart

Bit Rate and Bit Timing Settings: The bit rate and bit timing settings are made in the bit configuration register (BCR). Settings should be made such that all CAN controllers connected to the CAN bus have the same baud rate and bit width. The 1-bit time consists of the total of the settable time quantum (tq).

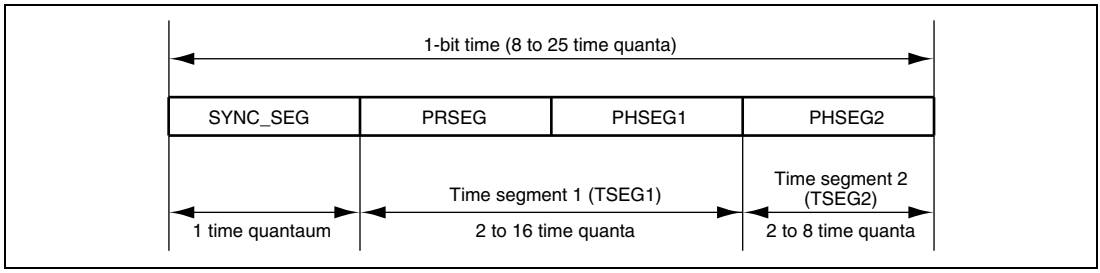


Figure 18.8 Detailed Description of One Bit

SYNC_SEG is a segment for establishing the synchronization of nodes on the CAN bus. Normal bit edge transitions occur in this segment. PRSEG is a segment for compensating for the physical delay between networks. PHSEG1 is a buffer segment for correcting phase drift (positive). This segment is extended when synchronization (resynchronization) is established. PHSEG2 is a buffer segment for correcting phase drift (negative). This segment is shortened when synchronization (resynchronization) is established. Limits on the settable value (TSEG1, TSEG2, BRP, sample point, and SJW) are shown in table 18.2.

Table 18.2 Limits for Settable Value

Name	Abbreviation	Min. Value	Max. Value
Time segment 1	TSEG1	B′0011*2	B′1111
Time segment 2	TSEG2	B′001*3	B′111
Baud rate prescaler	BRP	B′000000	B′111111
Bit sample point	BSP	B′0	B′1
Re-synchronization jump width	SJW*1	B′00	B′11

Notes: 1. SJW is stipulated in the CAN specifications:
$4 \geq SJW \geq 1$
2. The minimum value of TSEG2 is stipulated in the CAN specifications:
$TSEG2 \geq 2 + SJW$
3. The minimum value of TSEG1 is stipulated in the CAN specifications:
$TSEG1 > TSEG2$

Time quanta (TQ) is an integer multiple of the number of system clocks, and is determined by the baud rate prescaler (BRP) as follows. f_{CLK} is the system clock frequency.

$TQ = 2 \times (BPR\ setting + 1)/f_{CLK}$

The following formula is used to calculate the 1-bit time and bit rate.

$1\text{-bit time} = TQ \times (3 + TSEG1 + TSEG2)$

Bit rate = 1/Bit time

$= f_{CLK}/\{2 \times (BPR\ setting + 1) \times (3 + TSEG1 + TSEG2)\}$

Note: $f_{CLK} = \phi$ (system clock)
A BCR value is used for BRP, TSEG1, and TSEG2.

Example: With a system clock of 20 MHz, a BRP setting of B′000000, a TSEG1 setting of B′0100, and a TSEG2 setting of B′011:

Bit rate = $20/\{2 \times (0 + 1) \times (3 + 4 + 3)\} = 1$ Mbps

Table 18.3 Setting Range for TSEG1 and TSEG2 in BCR

		TSEG2 (BCR[14:12])						
		001	010	011	100	101	110	111
		2	3	4	5	6	7	8
TSEG1 (BCR[11:8])	0011	No	Yes	No	No	No	No	No
	0100	Yes*	Yes	Yes	No	No	No	No
	0101	Yes*	Yes	Yes	Yes	No	No	No
	0110	Yes*	Yes	Yes	Yes	Yes	No	No
	0111	Yes*	Yes	Yes	Yes	Yes	Yes	No
	1000	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1001	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1010	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1011	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1100	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1101	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1110	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1111	Yes*	Yes	Yes	Yes	Yes	Yes	Yes

Note: The time quantum value for TSEG1 and TSEG2 is the TSEG value + 1.
* When baud rate prescaler (BRP) is not B′000000 (2 × system clock), this can be set.

Mailbox Transmit/Receive Settings: The HCAN has 16 mailboxes. Mailbox 0 is receive-only, while mailboxes 1 to 15 can be set for transmission or reception. The Initial status of mailboxes 1 to 15 is for transmission. Mailbox transmit/receive settings are not initialized by a software reset.

Clearing a bit to 0 in the mailbox configuration register (MBCR) designates the corresponding mailbox for transmission use, whereas a setting of 1 in MBCR designates the corresponding mailbox for reception use. When setting mailboxes for reception, in order to improve message reception efficiency, high-priority messages should be set in low-to-high mailbox order.

Mailbox (Message Control/Data) Initial Settings: Message control/data are held in RAM, and so their initial values are undefined after power is supplied. Initial values must therefore be set in all the mailboxes (by writing 0s or 1s).

Setting the Message Transmission Method: The following two kinds of message transmission methods are available.

- Transmission order determined by message identifier priority
- Transmission order determined by mailbox number priority

Either of the message transmission methods can be selected with the message transmission method bit (MCR2) in the master control register (MCR): When messages are set to be transmitted according to the message identifier priority, if several messages are designated as waiting for transmission (TXPR = 1), the message with the highest priority in the message identifier is stored in the transmit buffer. CAN bus arbitration is then carried out for the message stored in the transmit buffer, and the message is transmitted when the transmission right is acquired. When the TXPR bit is set, the highest-priority message is found and stored in the transmit buffer.

When messages are set to be transmitted according to the mailbox number priority, if several messages are designated as waiting for transmission (TXPR = 1), messages are stored in the transmit buffer in low-to-high mailbox order. CAN bus arbitration is then carried out for the message stored in the transmit buffer, and the message is transmitted when the transmission right is acquired.

18.4.3 Message Transmission

Messages are transmitted using mailboxes 1 to 15. The transmission procedure after initial settings is described below, and a transmission flowchart is shown in figure 18.9.

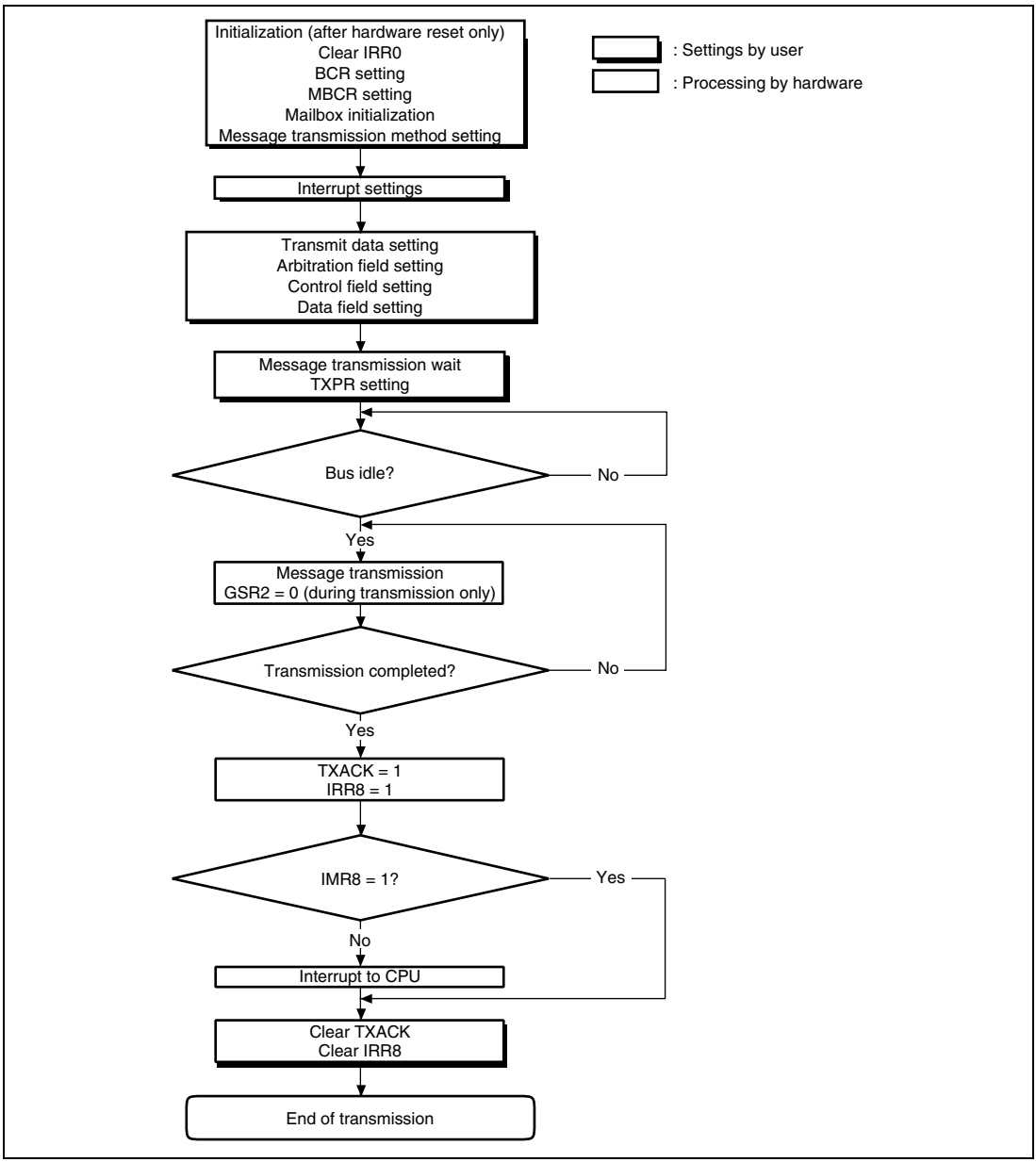


Figure 18.9 Transmission Flowchart

CPU interrupt source settings: The CPU interrupt source is set by the interrupt mask register (IMR) and mailbox interrupt mask register (MBIMR). Transmission acknowledge and transmission abort acknowledge interrupts can be generated for individual mailboxes in the mailbox interrupt mask register (MBIMR). Interrupt sources of the interrupt register (IRR) can be masked by IMR.

Arbitration field setting: The arbitration field is set by message control registers MCx[5]–MCx[8] in a transmit mailbox. For a standard format, an 11-bit identifier (ID-28 to ID-18) and the RTR bit are set, and the IDE bit is cleared to 0. For an extended format, a 29-bit identifier (ID-28 to ID-0) and the RTR bit are set, and the IDE bit is set to 1.

Control field setting: In the control field, the byte length of the data to be transmitted is set within the range of zero to eight bytes. The register to be set is the message control register MCx[1] in a transmit mailbox.

Data field setting: In the data field, the data to be transmitted is set within the range zero to eight. The registers to be set are the message data registers MDx[1]–MDx[8]. The byte length of the data to be transmitted is determined by the data length code in the control field. Even if data exceeding the value set in the control field is set in the data field, up to the byte length set in the control field will actually be transmitted.

Message transmission: If the corresponding mailbox transmit wait bit (TXPR1–TXPR15) in the transmit wait register (TXPR) is set to 1 after message control and message data registers have been set, the message enters transmit wait state. If the message is transmitted error-free, the corresponding acknowledge bit (TXACK1–TXACK15) in the transmit acknowledge register (TXACK) is set to 1, and the corresponding transmit wait bit (TXPR1–TXPR15) in the transmit wait register (TXPR) is automatically cleared to 0. Also, if the corresponding bit (MBIMR1-MBIMR15) in the mailbox interrupt mask register (MBIMR) and the mailbox empty interrupt bit (IRR8) in the interrupt mask register (IMR) are both simultaneously set to enable interrupts, interrupts may be sent to the CPU.

If transmission of a transmit message is aborted in the following cases, the message is retransmitted automatically:

- CAN bus arbitration failure (failure to acquire the bus)
- Error during transmission (bit error, stuff error, CRC error, frame error, or ACK error)

Message transmission cancellation: Transmission cancellation can be specified for a message stored in a mailbox as a transmit wait message. A transmit wait message is canceled by setting the bit for the corresponding mailbox (TXCR1–TXCR15) to 1 in the transmit cancel register (TXCR). Clearing the transmit wait register (TXPR) does not cancel transmission. When cancellation is executed, the transmit wait register (TXPR) is automatically reset, and the corresponding bit is set to 1 in the abort acknowledge register (ABACK). An interrupt to the CPU can be requested, and if the mailbox empty interrupt (IRR8) is enabled for the bits (MBIMR1-MBIMR15) corresponding

to the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR), interrupts may be sent to the CPU.

However, a transmit wait message cannot be canceled at the following times:

- During internal arbitration or CAN bus arbitration
- During data frame or remote frame transmission

Figure 18.10 shows a flowchart for transmit message cancellation.

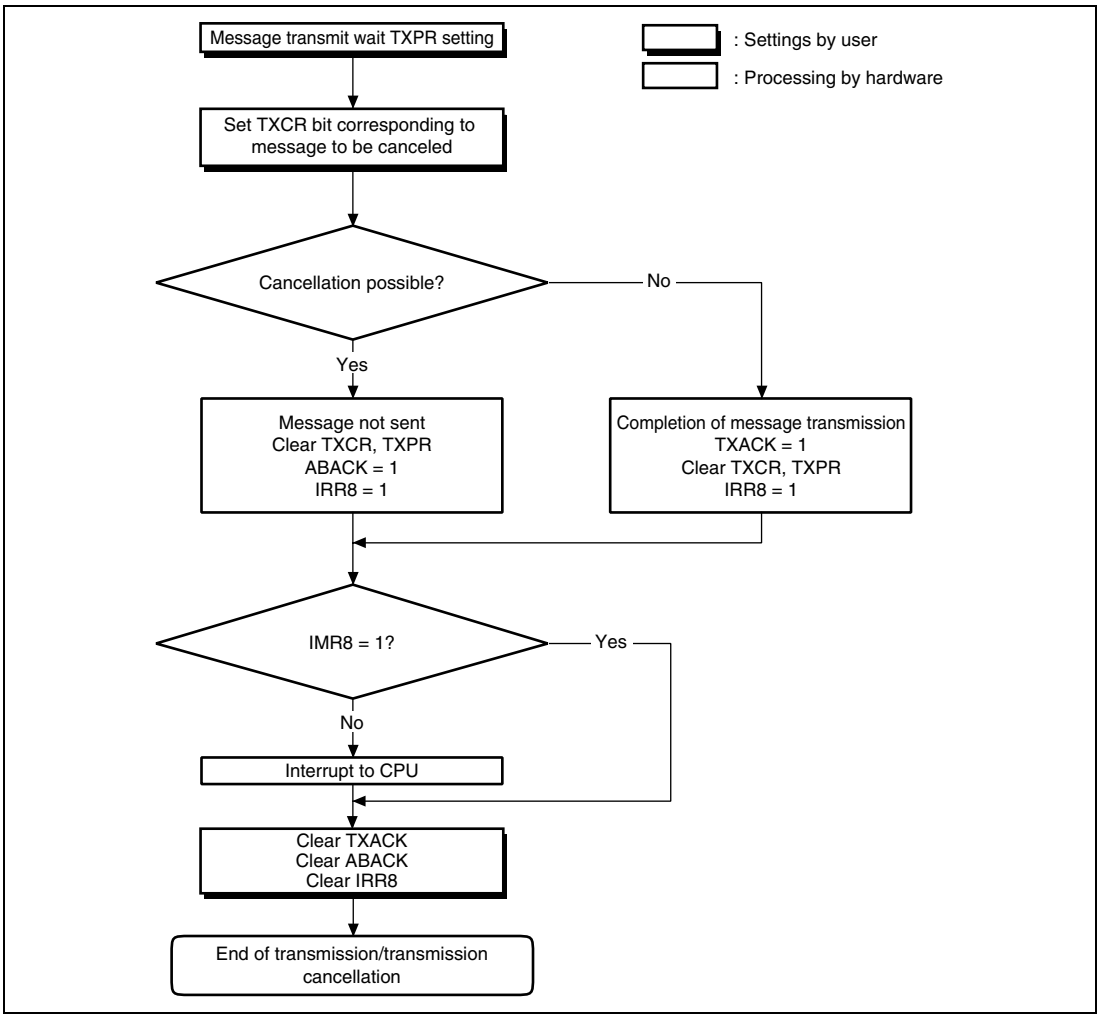


Figure 18.10 Transmit Message Cancellation Flowchart

18.4.4 Message Reception

The reception procedure after initial settings is described below. A reception flowchart is shown in Figure 18.11.

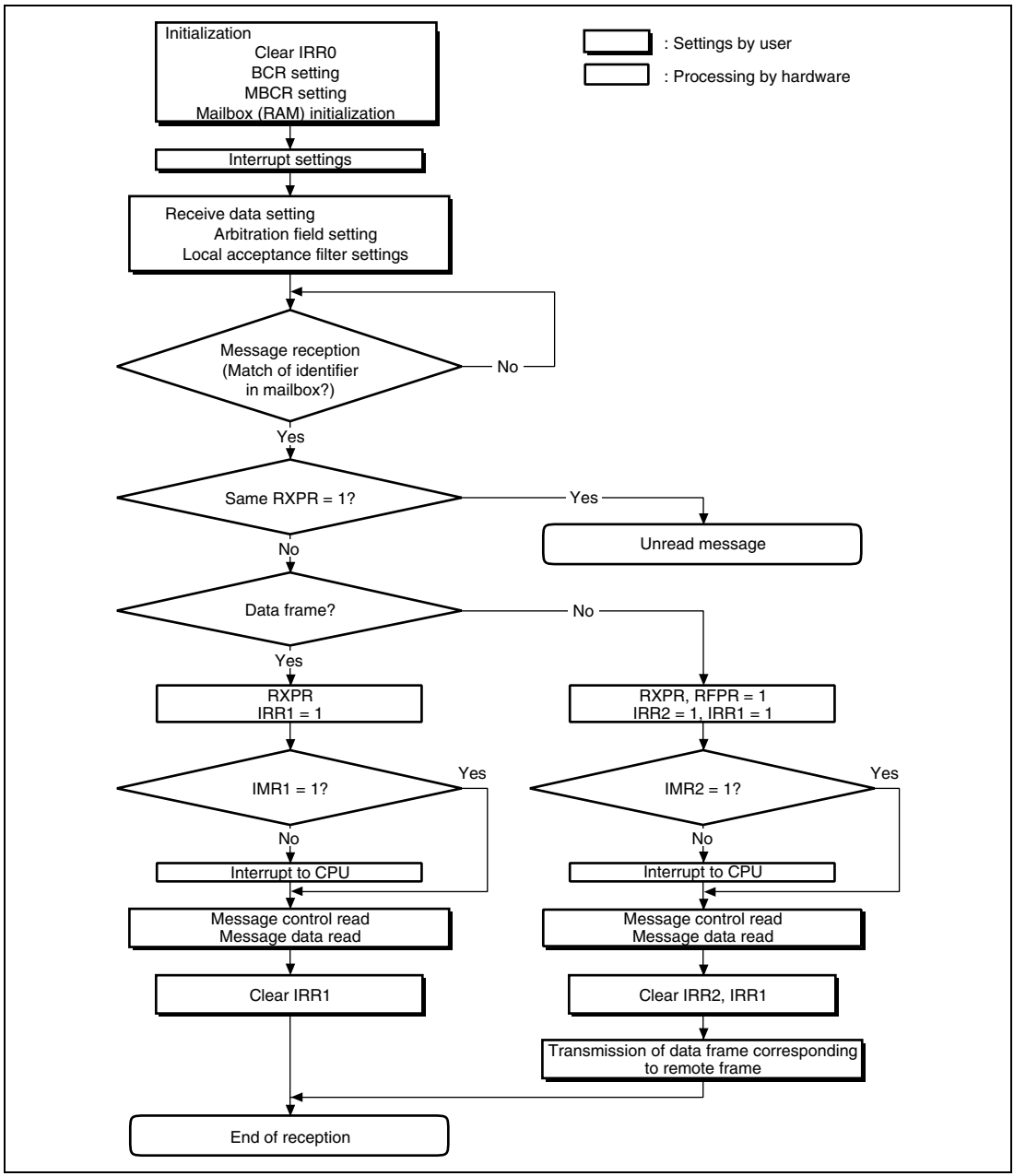


Figure 18.11 Reception Flowchart

CPU interrupt source settings: CPU interrupt source settings are made in the interrupt mask register (IMR) and mailbox interrupt register (MBIMR). The message to be received is also specified. Data frame and remote frame receive wait interrupt requests can be generated for individual mailboxes in the MBIMR. Interrupt sources of the interrupt register (IRR) are enabled by IMR.

Arbitration field setting: To receive a message, the message identifier must be set in advance in the message control registers (MCx[1]–MCx[8]) for the receiving mailbox. When a message is received, all the bits in the receive message identifier are compared with those in each message control register identifier, and if a 100% match is found, the message is stored in the matching mailbox. Mailbox 0 has a local acceptance filter mask (LAFM) that allows Don't Care settings to be made. The LAFM setting can be made only for mailbox 0. By making the Don't Care setting for all the bits in the receive message identifier, messages of multiple identifiers can be received.

Examples:

- When the identifier of mailbox 1 is 010_1010_1010 (standard format), only one kind of message identifier can be received by mailbox 1:

  Identifier 1: 010_1010_1010
- When the identifier of mailbox 0 is 010_1010_1010 (standard format) and the LAFM setting is 000_0000_0011 (0: Care, 1: Don't Care), a total of four kinds of message identifiers can be received by mailbox 0:

  Identifier 1: 010_1010_1000

  Identifier 2: 010_1010_1001

  Identifier 3: 010_1010_1010

  Identifier 4: 010_1010_1011

Message reception: When a message is received, a CRC check is performed automatically. If the result of the CRC check is normal, ACK is transmitted in the ACK field irrespective of whether the message can be received or not.

- Data frame reception

  If the received message is confirmed to be error-free by the CRC check, the identifier in the mailbox (and also LAFM in the case of mailbox 0 only) and the identifier of the receive message, are compared. If a complete match is found, the message is stored in the mailbox. The message identifier comparison is carried out on each mailbox in turn, starting with mailbox 0 and ending with mailbox 15. If a complete match is found, the comparison ends at that point, the message is stored in the matching mailbox, and the corresponding receive complete bit (RXPR0–RXPR15) is set in the receive complete register (RXPR). However, when a mailbox 0 LAFM comparison is carried out, even if the identifier matches, the mailbox comparison sequence does not end at that point, but continues with mailbox 1 and then the remaining mailboxes. It is therefore possible for a message matching mailbox 0 to be received by another mailbox. Note that the same message cannot be stored in more than one of

mailboxes 1 to 15. On receiving a message, a CPU interrupt request may be generated depending on the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR) settings.

- Remote frame reception

  Two kinds of messages—data frames and remote frames—can be stored in mailboxes. A remote frame differs from a data frame in that the remote transmission request bit (RTR) in the message control register and the data field are 0 bytes long. The data length to be returned in a data frame must be stored in the data length code (DLC) in the control field.

  When a remote frame (RTR = recessive) is received, the corresponding bit is set in the remote request wait register (RFPR). If the corresponding bit (MBIMR0–MBIMR15) in the mailbox interrupt mask register (MBIMR) and the remote frame request interrupt mask (IRR2) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

Unread message overwrite: If the receive message identifier matches the mailbox identifier, the receive message is stored in the mailbox regardless of whether the mailbox contains an unread message or not. If a message overwrite occurs, the corresponding bit (UMSR0–UMSR15) is set in the unread message register (UMSR). In overwriting an unread message, when a new message is received before the corresponding bit in the receive complete register (RXPR) has been cleared, the unread message register (UMSR) is set. If the unread interrupt flag (IRR9) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU. Figure 18.12 shows a flowchart for unread message overwriting.

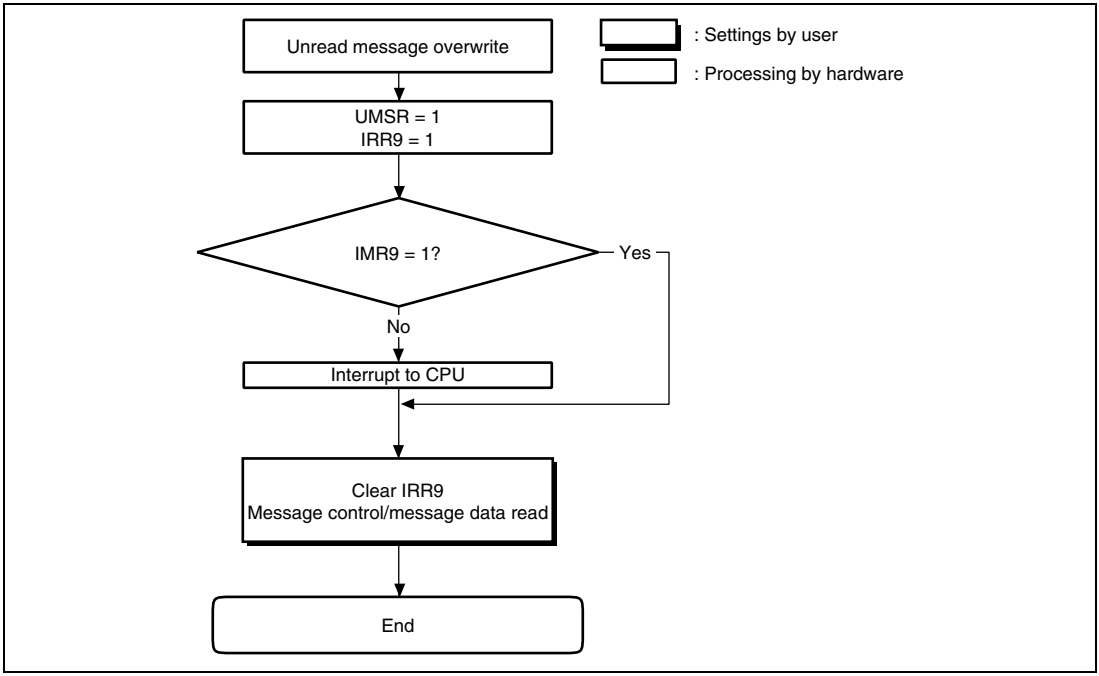


Figure 18.12 Unread Message Overwrite Flowchart

18.4.5 HCAN Sleep Mode

The HCAN is provided with an HCAN sleep mode that places the HCAN module in the sleep state in order to reduce current dissipation. Figure 18.13 shows a flowchart of the HCAN sleep mode.

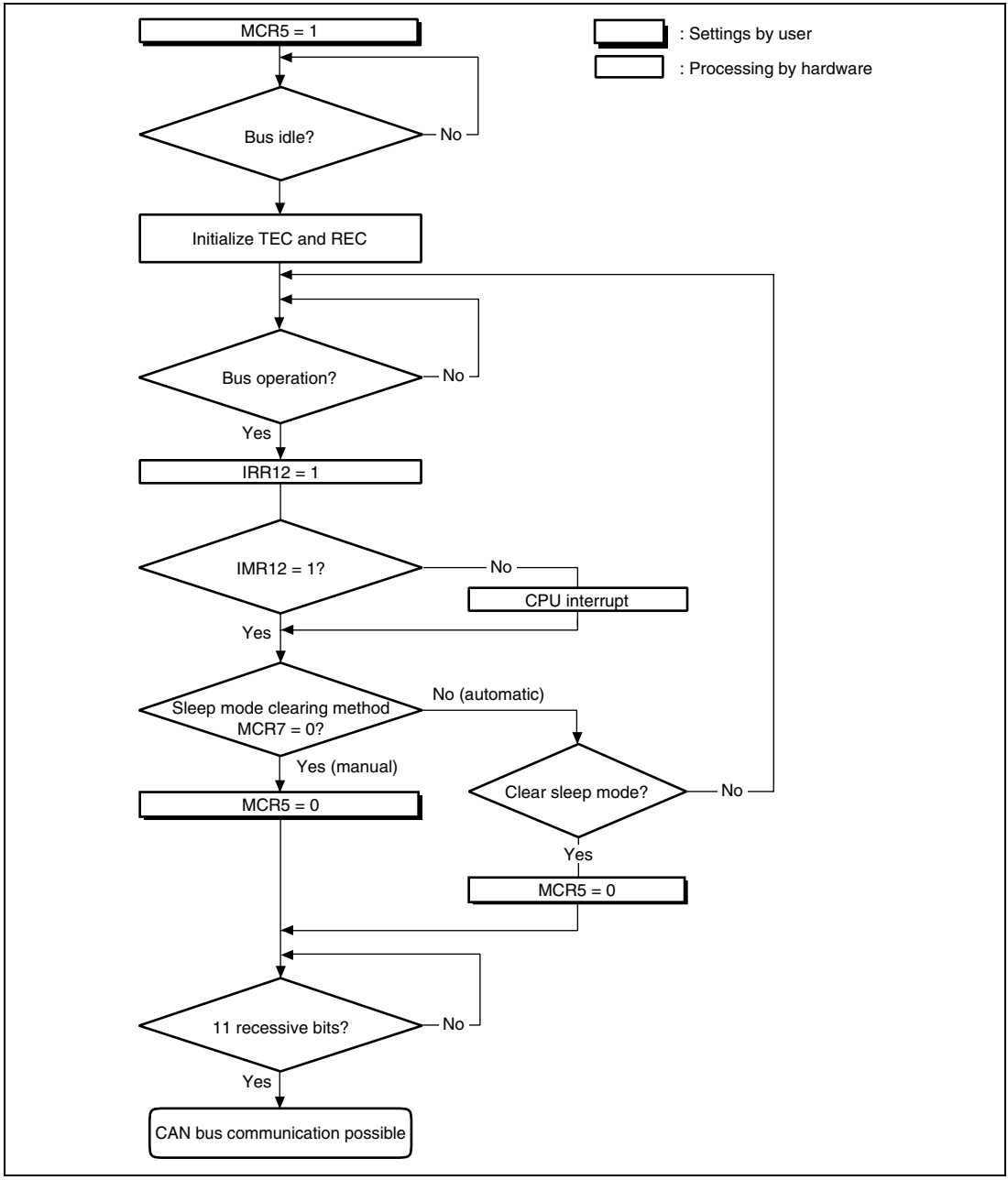


Figure 18.13 HCAN Sleep Mode Flowchart

HCAN sleep mode is entered by setting the HCAN sleep mode bit (MCR5) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN sleep mode is delayed until the bus becomes idle.

Either of the following methods of clearing HCAN sleep mode can be selected:

- Clearing by software
- Clearing by CAN bus operation

Eleven recessive bits must be received after HCAN sleep mode is cleared before CAN bus communication is re-enabled.

Clearing by software: HCAN sleep mode is cleared by writing a 0 to MCR5 from the CPU.

Clearing by CAN bus operation: The cancellation method is selected by the MCR7 bit setting in MCR. Clearing by CAN bus operation occurs automatically when the CAN bus performs an operation and this change is detected. In this case, the first message is not stored in a mailbox; messages will be received normally from the second message onward. When a change is detected on the CAN bus in HCAN sleep mode, the bus operation interrupt flag (IRR12) is set in the interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

18.4.6 HCAN Halt Mode

The HCAN halt mode is provided to enable mailbox settings to be changed without performing an HCAN hardware or software reset. Figure 18.14 shows a flowchart of the HCAN halt mode.

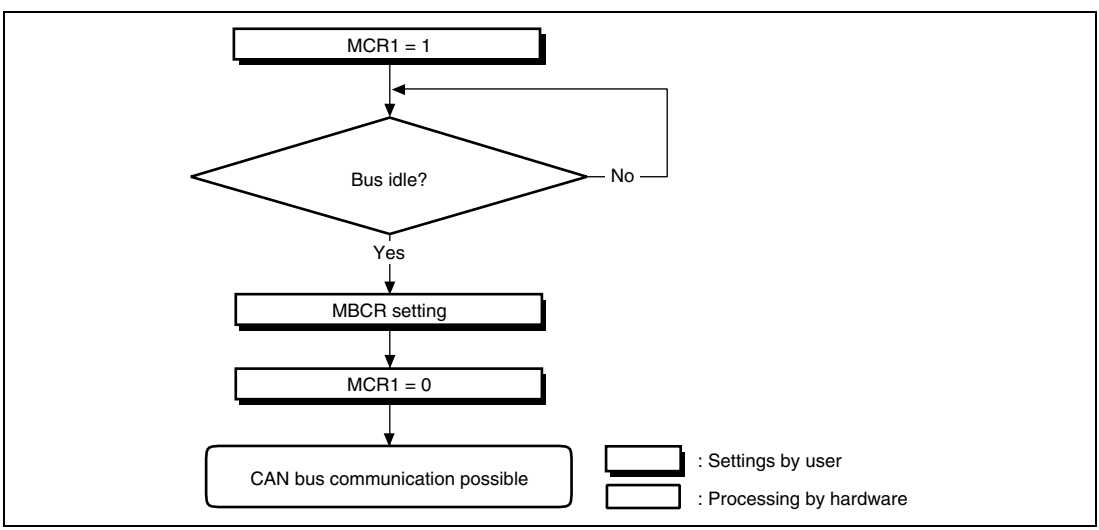


Figure 18.14 HCAN Halt Mode Flowchart

HCAN halt mode is entered by setting the halt request bit (MCR1) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN halt mode is delayed until the bus becomes idle.

HCAN halt mode is cleared by clearing MCR1 to 0.

18.5 Interrupts

Table 18.4 lists the HCAN interrupt sources. With the exception of the reset processing vector (IRR0), these sources can be masked. Masking is implemented using the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, refer to section 5, Interrupt Controller.

Table 18.4 HCAN Interrupt Sources

Name	Description	Interrupt Flag	DTC Activation
ERS0/OVR0	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
	Bus off interrupt (TEC ≥ 256)	IRR6	
	Reset process interrupt by power-on reset	IRR0	
	Remote frame reception interrupt	IRR2	
	Error warning interrupt (TEC ≥ 96)	IRR3	
	Error warning interrupt (REC ≥ 96)	IRR4	
	Overload frame transmission interrupt/bus off recovery interrupt (11 recessive bits × 128 times)	IRR7	
	Unread message overwrite interrupt	IRR9	
	CAN bus operation in HCAN sleep mode interrupt	IRR12	
RM0	Mailbox 0 message reception interrupt	IRR1	Possible
RM1	Mailbox 1-15 message reception interrupt	IRR1	Not possible
SLE0	Message transmission/cancellation interrupt	IRR8	

18.6 DTC Interface

The DTC can be activated by the reception of a message in HCAN mailbox 0. When DTC transfer ends after DTC activation has been set, the RXPR0 and RFPR0 flags are cleared automatically. An interrupt request due to a receive interrupt from the HCAN cannot be sent to the CPU in this case. Figure 18.15 shows a DTC transfer flowchart.

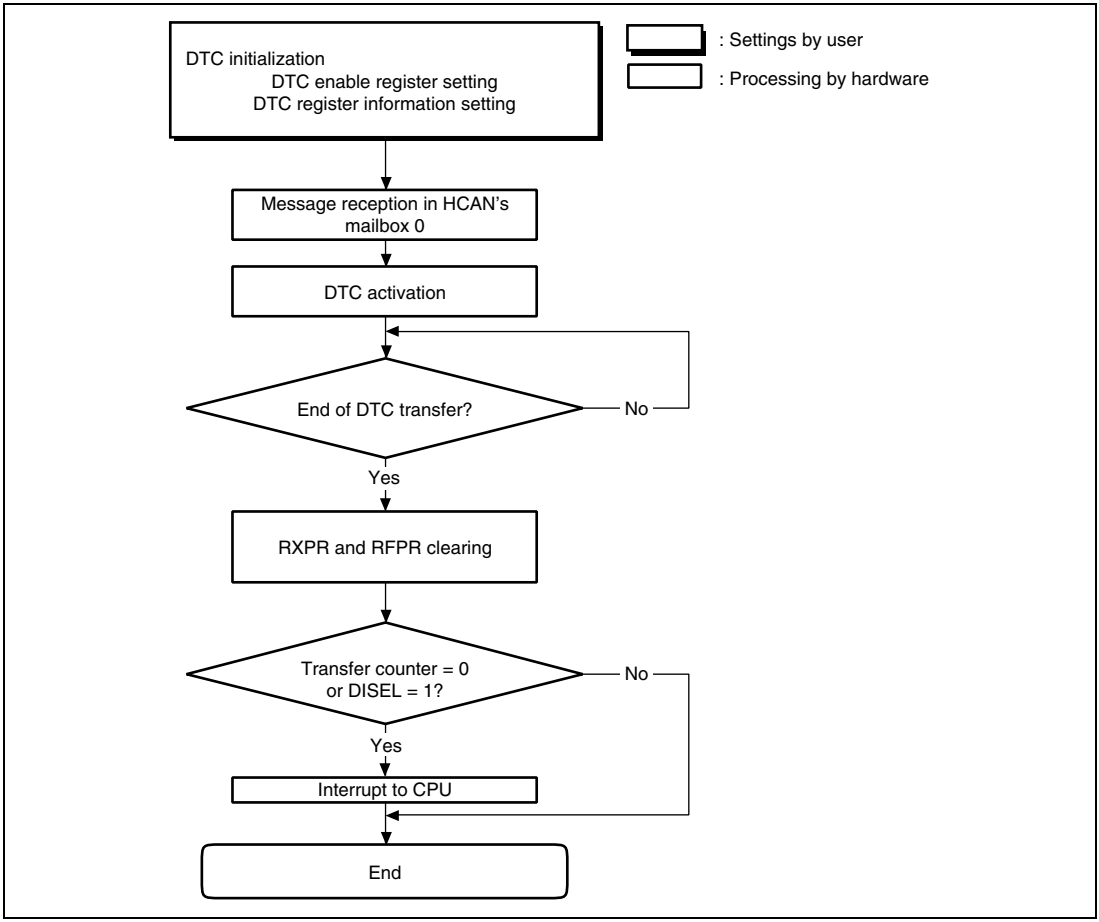


Figure 18.15 DTC Transfer Flowchart

18.7 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Philips PCA82C250 transceiver IC is recommended. Any other product must be compatible with the PCA82C250. Figure 18.16 shows a sample connection diagram.

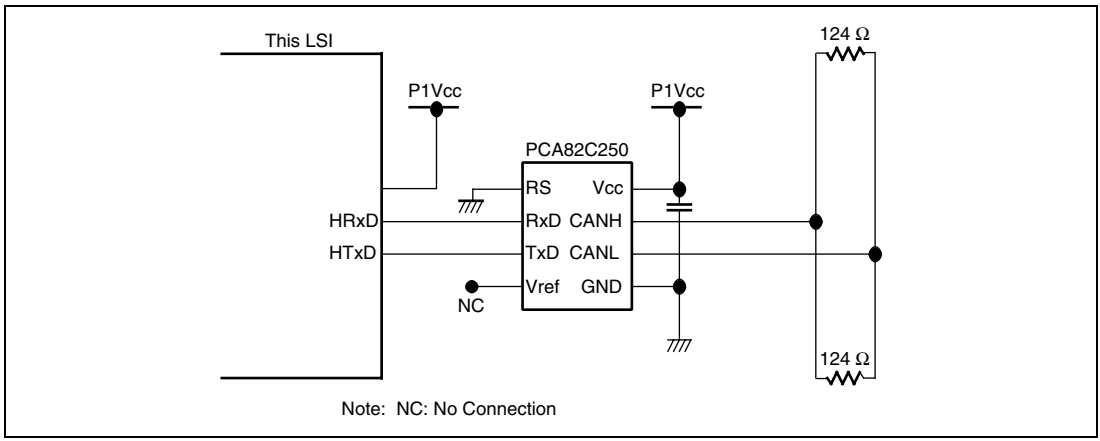


Figure 18.16 High-Speed Interface Using PCA82C250

18.8 Usage Notes

18.8.1 Module Stop Mode Setting

HCAN operation can be disabled or enabled using the module stop control register. The initial setting is for HCAN operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

18.8.2 Reset

The HCAN is reset by a power-on reset or in hardware standby mode, software standby mode, watch mode, or module stop mode. All the registers are initialized in a reset, however mailboxes (message control (MCx[x])/message data (MDx[x])) are not. After power-on, mailboxes (message control (MCx[x])/message data (MDx[x])) are initialized, and their values are undefined. Therefore, mailbox initialization must always be carried out after a power-on reset or recovery from hardware standby mode, software standby mode, watch mode, or module stop mode. The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode, watch mode, or module stop mode. As this bit cannot be masked in the interrupt mask register (IMR), if HCAN interrupt enabling is set in the interrupt controller without clearing the flag, an HCAN interrupt will be initiated immediately. IRR0 should therefore be cleared during initialization.

18.8.3 HCAN Sleep Mode

The bus operation interrupt flag (IRR12) in the interrupt register (IRR) is set by CAN bus operation in HCAN sleep mode. Therefore, this flag is not used by the HCAN to indicate sleep mode release. Note that the reset status bit (GSR3) in the general status register (GSR) is set in sleep mode.

18.8.4 Interrupts

When the mailbox interrupt mask register (MBIMR) is set, the interrupt register (IRR8, 2, 1) is not set by reception completion, transmission completion, or transmission cancellation for the set mailboxes.

18.8.5 Error Counters

In the case of error active and error passive, REC and TEC normally count up and down. In the bus-off state, 11-bit recessive sequences are counted (REC + 1) using REC. If REC reaches 96 during the count, IRR4 and GSR1 are set. If REC reaches 128 during the count, IRR7 is set.

18.8.6 Register Access

Byte or word access can be used on all HCAN registers. Longword access cannot be used.

18.8.7 HCAN Medium-Speed Mode

In medium-speed mode, neither read nor write is possible for the HCAN registers.

18.8.8 Register Hold in Standby Modes

All HCAN registers are initialized in hardware standby mode and software standby mode.

18.8.9 Usage of Bit Change Instructions

Do not use bit change instructions to clear flags, because the status flags of HCAN is cleared by writing 1. To clear a flag, use MOV instruction to write 1 to only the bits to be cleared.

Section 19 RAM

This LSI includes high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data.

The RAM can be enabled or disabled using the RAME bit in the system control register (SYSCR). For details on SYSCR, refer to section 3.2.2, System Control Register (SYSCR).

Section 20 Flash Memory (0.18-µm F-ZTAT Version)

This LSI has 512-kbyte of on-chip flash memory. Features of flash memory are shown below.

20.1 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
  - 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.
- One off-board programming mode
  - Programmer 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. The user branch is also supported. In addition, this LSI supports user branch.

  User Branch*

  Programming is carried out in 128-byte units through several steps such as applying programming pulse and reading verify. Erasing is carried out in 1-divided-block unit through several processing steps. Between these steps, a setting can be created to execute user process routines. This setting is called "with user branch."

Note: * Not available in this LSI.

- 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.
- Protect Mode

  Software protect by register setting is provided to set the protect status of programming/erasing flash memory

  When an error is detected, the mode is changed to error protect mode to abort programming/erasing processing.
- Programming/erasing time

  The flash memory programming time is TBD ms (typ) in 128-byte simultaneous programming and TBD μs per byte. The erasing time is TBD ms (typ) per block.
- Number of programming

  The number of flash memory programming can be up to 100 times.

20.1.1 Block Diagram

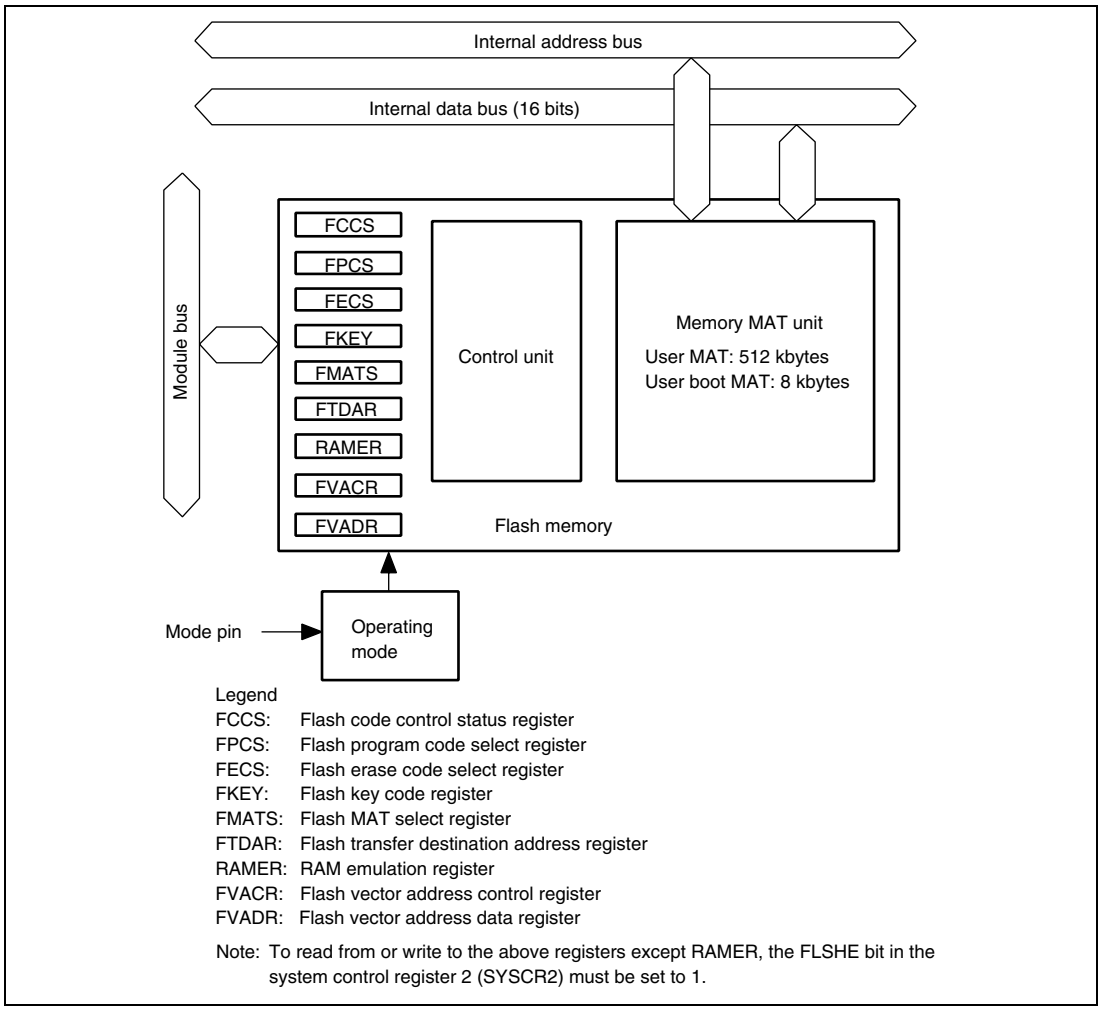


Figure 20.1 Block Diagram of Flash Memory

20.1.2 Operating Mode

When each mode pin is set in the reset state and reset start is performed, the microcomputer enters each operating mode as shown in figure 20.2. For the setting of each pin, refer to table 20.1.

- 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 programmer mode.

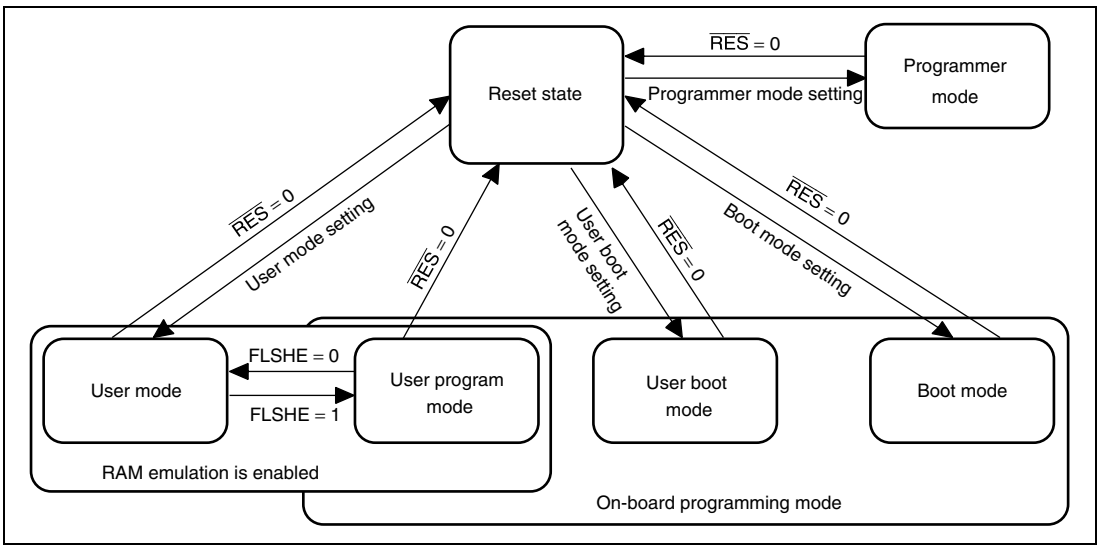


Figure 20.2 Mode Transition of Flash Memory

Table 20.1 MD Pin Setting and Operating Mode

Pin	Reset state	On-chip ROM valid mode*[1]	User program mode*[2]	User boot mode	Boot mode	Programmer mode
$\overline{\text{RES}}$	0	1	1	1	1	1
MD0	0/1	0/1	0/1	1	0/1	0
MD1	0/1	1	1	0	1	0
MD2	0/1	1	1	0	0	0

Notes: 1 On-chip ROM valid mode indicates mode 6 and mode 7. For details, refer to section 3, MCU Operating Modes.

2 To transit to User program mode, set FLSHE bit in SYSCR2 to 1.

20.1.3 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 20.2.

Table 20.2 Comparison of Programming Modes

	Boot mode	User program mode	User boot mode	Programmer 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)	○	○	○ (Automatic)
Block division erasure	○ *1	○	○	×
Program data transfer	From host via SCI	From optional device via RAM	From optional device via RAM	Via programmer
User branch	×	×	×	×
RAM emulation	×	○	×	×
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*2	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE setting	Changing mode setting and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. Firstly, the reset vector is fetched from the embedded program storage MAT. After the flash memory related registers are checked, the reset vector is fetched from the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode and programmer 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.

20.1.4 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 register.

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 programmer mode.

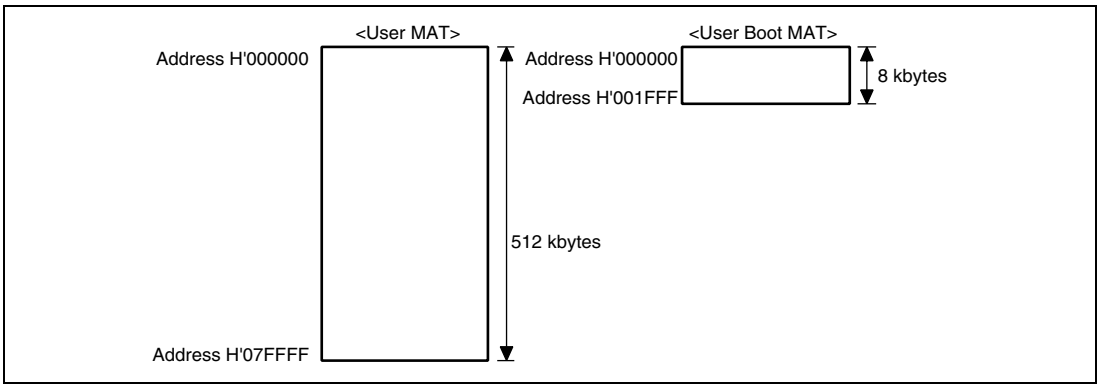


Figure 20.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address which exceeds the size of the 8-kbyte user boot MAT should not be accessed. If the attempt is made, data is read as undefined value.

20.1.5 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 20.4. 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.

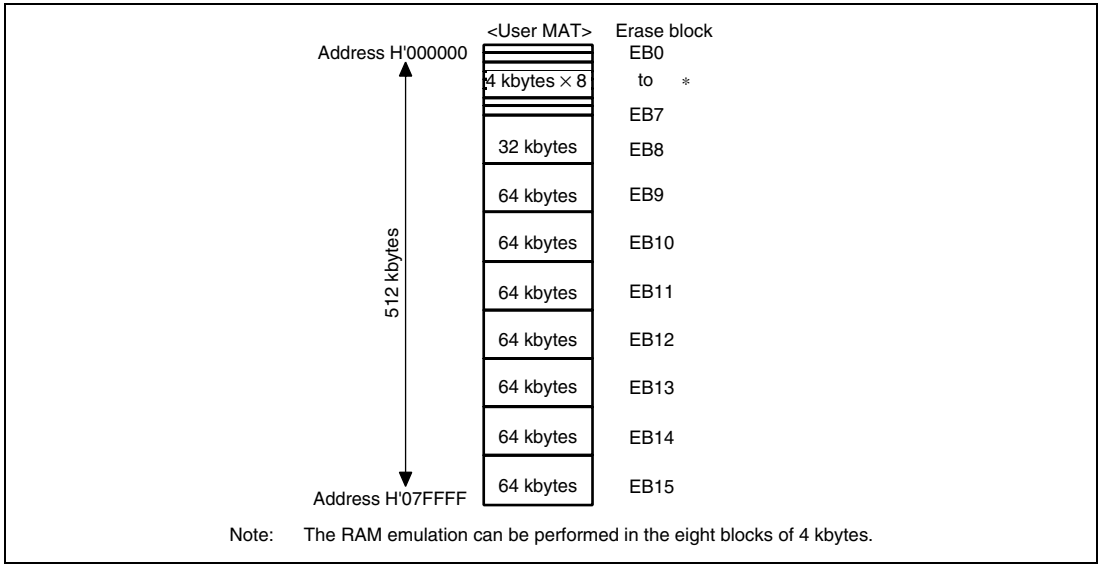


Figure 20.4 Block Division of User MAT

20.1.6 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. An overview of the procedure is given as follows. For details, see section 20.4.2, User Program Mode.

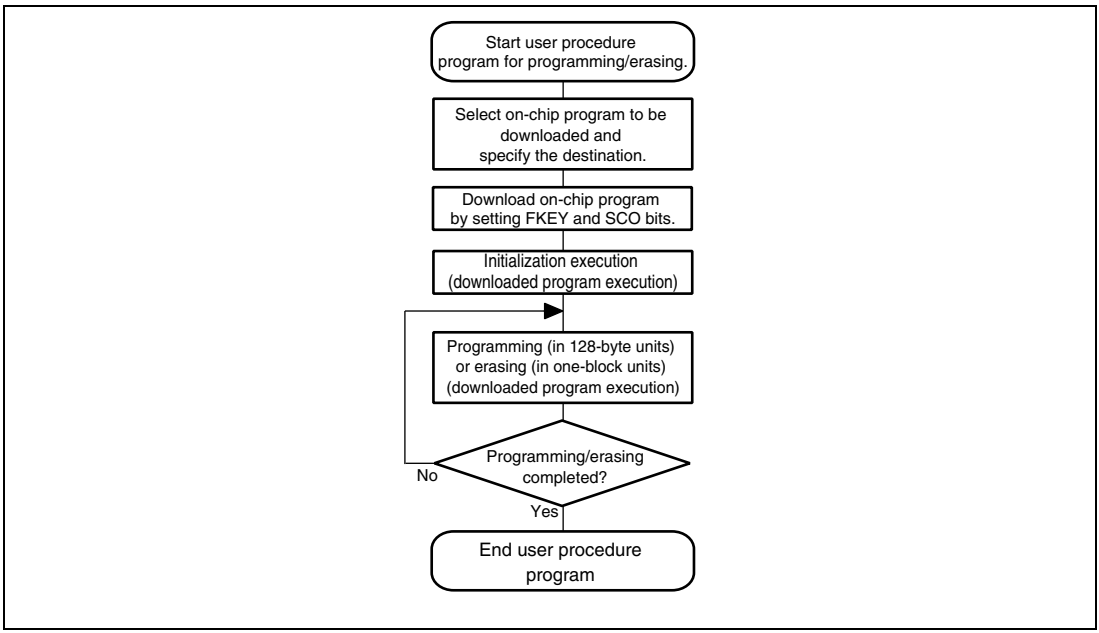


Figure 20.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded

   This LSI has programming/erasing programs which 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 address of the programming destination is specified by the FTDAR.

2. Download of on-chip program

   The on-chip program is automatically downloaded by setting the SCO bit in the flash key register (FKEY) and the flash control register (FCCS) of the programming/erasing interface register.

   The flash memory 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 the 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 parameter, 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 area should be outside programming-prohibited areas such as user MAT area during programming process or on-chip program area. This setting is performed by using the programming/erasing interface parameter.

4. Programming/erasing execution

   To execute programming/erasing, it is necessary to enter user program mode by setting FLSHE bit in SYSCR2 to 1.

   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 parameter and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction and performing the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameter.

   The area to be programmed must be erased in advance when programming flash memory. It is, however, impossible to download at the same time the erasing program and the programming program. Therefore, execute the above procedures of 1 to 4 in the order of erasing first and programming next.

   All interrupts are prohibited during programming and erasing. Interrupts must be masked within the user system.

   Access in the flash memory space during programming/erasing is not guaranteed. Accordingly, when the interrupt vector or the interrupt handler is in the flash memory, interrupt processing is not guaranteed. When NMI interrupt is inevitable during overprogramming/erasing system such as in system error processing, set FVACR and FVADR to set the interrupt vector and the interrupt processing routine in the on-chip RAM or the external space.

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.

20.2 Pin Configuration

Table 20.3 shows the flash memory pin configuration.

Table 20.3 Pin Configuration

Pin Name	Abbreviation	Input/Output	Function
Reset	$\overline{\text{RES}}$	Input	Reset
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
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 programmer mode, see section 20.9, Programmer Mode.

20.3 Register Descriptions

The registers/parameters which control flash memory are shown as follows. To access registers other than RAMER that control flash memory, set the FLSHE bit in SYSCR2 in mode which makes flash memory valid.

- Flash code control status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- System control register 2 (SYSCR2)
- Flash pass and fail result (FPFR)
- Download pass and fail result (DPFP)
- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase Block select (FEBS)
- Flash program and erase frequency control (FPEFEQ)
- Flash user branch address set parameter (FUBRA)
- RAM emulation register (RAMER)
- Flash vector address control register (FVACR)
- Flash vector address data register R (FVADRR)
- Flash vector address data register E (FVADRE)

- Flash vector address data register H (FVADRH)
- Flash vector address data register L (FVADRL)

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 20.4.

Table 20.4 Register/Parameter and Target Mode

		Download	Initialization	Programming	Erasure	Read	RAM Emulation
Programming/ Erasing Interface Register	FCCS	○	—	—	—	—	—
	FPCS	○	—	—	—	—	—
	FECS	○	—	—	—	—	—
	FKEY	○	—	○	○	—	—
	FMATS	—	—	○ *1	○ *1	○ *2	—
	FTDAR	○	—	—	—	—	—
Programming/ Erasing Interface Parameter	DPFR	○	—	—	—	—	—
	FPFR	—	○	○	○	—	—
	FPEFEQ	—	○	—	—	—	—
	FUBRA	—	○	—	—	—	—
	FMPAR	—	—	○	—	—	—
	FMPDR	—	—	○	—	—	—
	FEBS	—	—	—	○	—	—
RAM Emulation	RAMER	—	—	—	—	—	○

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.

20.3.1 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.

Flash Code Control and Status Register (FCCS): FCCS is configured by bits which request the monitor of error occurrence during programming or erasing flash memory and the download of on-chip program.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R	Reserved This bit is always read as 0. The write value should always be 1.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FLER	0	R	Flash Memory Error 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 transition to a power-on reset or 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 μs which is longer than normal. 0: Flash memory operates normally 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 20.5.3, Error Protection
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	SCO	0	(R)/W	Source Program Copy Operation 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 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 executed 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. 0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed [Clear condition] When download is completed 1: Request that the on-chip programming/erasing program is downloaded to the on-chip RAM is occurred [Set conditions] When all of the following conditions are satisfied and 1 is written to this bit • H'A5 is written to FKEY • During execution in the on-chip RAM • Not in RAM emulation mode (RAMS in RAMER = 0)

Flash Program Code Select Register (FPCS): FPCS selects the on-chip programming program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program. 0: On-chip programming program is not selected [Clear condition] When transfer is completed 1: On-chip programming program is selected

Flash Erase Code Select Register (FECS): FECS selects download of the on-chip erasing program.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program. 0: On-chip erasing program is not selected [Clear condition] When transfer is completed 1: On-chip erasing program is selected

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	Bit Name	Initial Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	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.
5	K5	0	R/W	
4	K4	0	R/W	
3	K3	0	R/W	Only when H′5A is written, programming/erasing can be executed. Even if the on-chip programming/erasing program is executed, the flash memory cannot be programmed or erased when the value other than H′5A is written to FKEY. 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′A5 is in software protection state.) H'00: Initial value
2	K2	0	R/W	
1	K1	0	R/W	
0	K0	0	R/W	

Flash MAT Select Register (FMATS): FMATS specifies whether user MAT or user boot MAT is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	MAT Select 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 20.7, 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 programmer mode.) 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	MS6	0	R/W	
5	MS5	0/1*	R/W	
4	MS4	0	R/W	
3	MS3	0/1*	R/W	
2	MS2	0	R/W	
1	MS1	0/1*	R/W	
0	MS0	0	R/W	

Note: Set to 1 when in user boot mode, otherwise set to 0.

Flash Transfer Destination Address Register (FTDAR): FTDAR is a register that specify the address to download an on-chip program. This register must be specified before setting the SCO bit in FCCS to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when the address specified by bits TDA6 to TDA0, which is the start address to download an on-chip program, is over the range. Whether or not the range specified by bits TDA6 to TDA0 is within the range of H′00 to H′07 is determined when an on-chip program is downloaded by setting the SCO bit in FCCS. Make sure that this bit is cleared to 0 before setting the SCO bit to 1 and the value specified by TDA6 to TDA0 is within the range of H′00 to H′07. 0: The value specified by bits TDA6 to TDA0 is within the range. 1: The value specified by is TDA6 to TDA0 is over the range (H′08 to H′FF) and the download is stopped.
6	TDA6	0	R/W	Transfer Destination Address Specifies the start address to download an on-chip program. H′00 to H′07 can be specified meaning that the start address in the on-chip RAM space can be specified in units of 4 kbytes. H′00: H′FF9000 is specified as a start address to download an on-chip program. H′01: H′FFA000 is specified as a start address to download an on-chip program. H′02: H′FFB000 is specified as a start address to download an on-chip program. H′03: H′FFC000 is specified as a start address to download an on-chip program. H′04: H′FFD000 is specified as a start address to download an on-chip program. H′05: H′FFE000 is specified as a start address to download an on-chip program. H′06: H′FF8000 is specified as a start address to download an on-chip program. H′07: H′FF7000 is specified as a start address to download an on-chip program. H′08 to H′FF: Setting prohibited. Specifying this value sets the TDRE bit to 1 and stops the download.
5	TDA5	0	R/W	
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	
0	TDA0	0	R/W	

System Control Register 2 (SYSCR2): SYSCR2 controls register accesses.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved The write value should always be 0.
3	FLSHE	0	R/W	Flash memory control register enable The access of the flash memory control register to the CPU is controlled by writing 0. Setting 1 to FLSHE bit enables reading/programming the flash memory control register. When this bit is cleared to 0, the flash memory control register is not selected. In this case, the content of the flash memory control register is retained. 0: Flash control logic unit which controls H′FFFFA4 to H′FFFFAF is disabled. 1: Flash control logic unit which controls H′FFFFA4 to H′FFFFAF is enabled.
2	—	Undefined	—	Reserved The write value should always be 0.
1, 0	—	All 0	R/W	Reserved The write value should always be 0.

20.3.2 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 20.5. The meaning of the bits in FPFR varies in each processing program: initialization, programming, or erasure. For details, see descriptions of FPFR for each process.

Table 20.5 Parameters and Target Modes

Name of Parameter	Abbreviation	Down Load	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
Download pass and fail result	DPFR	○	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass and fail result	FPFR	—	○	○	○	R/W	Undefined	R0L of CPU
Flash programming/ erasing frequency control	FPEFEQ	—	○	—	—	R/W	Undefined	ER0 of CPU
Flash user branch address set parameter	FUBRA	—	○	—	—	R/W	Undefined	ER1 of CPU
Flash multi-purpose address area	FMPAR	—	—	○	—	R/W	Undefined	ER1 of CPU
Flash multi-purpose data destination area	FMPDR	—	—	○	—	R/W	Undefined	ER0 of CPU
Flash erase block select	FEBS	—	—	—	○	R/W	Undefined	ER0 of CPU

Note: A single byte of the start address to download an on-chip program, which is specified by FTDAR.

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area to be downloaded is the 4-kbyte area starting from the address specified by FTDAR. For the address map of the on-chip RAM, see figure 20.10.

Download control is set by the programming/erasing interface registers, and the DPFR parameter indicates the return value.

(a) Download pass/fail result parameter (DPFR: single byte of start address 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 writing the single byte of the start address specified by FTDAR to the value other than the return value of download (for example, H′FF) before the download start (before setting the SCO bit to 1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved Return 0
2	SS	—	R/W	Source Select Error Detect Only one type for the on-chip program which can be downloaded can be specified. 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. 0: Download program can be selected normally 1: Download error is occurred (multi-selection or program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect (FK) Returns the check result whether the value of FKEY register is set to H′A5. 0: FKEY setting is normal (FKEY = H′A5) 1: Setting value of FKEY becomes error (FKEY = value other than H′A5)
0	SF	—	R/W	Success/Fail Returns the result whether download is ended normally or not. The determination result whether program that is downloaded to the on-chip RAM is read back and then transferred to the on-chip RAM is returned. 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 settable range of operating frequency in this LSI, refer to section 24.4.2, Clock Timing.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	F31 to F16	—	R/W	Reserved This bit should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set 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 value multiplied by 100 is converted to the binary digit and is written to the FPEFEQ parameter (general register ER0). For example, when the operating frequency of the CPU is 25.000 MHz, the value is as follows. 1. The number to three decimal places of 25.000 is rounded and the value is thus 25.00. 2. The formula that 33.00 × 100 = 3300 is converted to the binary digit and B′0000,1001,1100,0100 (H′09C4) is set to ER0.

(b) Flash user branch address setting parameter (FUBRA: General register ER1 of CPU)

This parameter sets the user branch destination. Set user program is executed in a specified unit during programming/erasure.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	UA31 to UA0	—	R/W	User Branch Destination Address When no user branch is required, set address 0 (H′00000000). The user branch destination should be the RAM space or external bus space to which on-chip program is not transferred. Be careful not to branch to the area without execution codes, or runaway or destruction of the on-chip program area or the stack area is caused. In case of runaway, the value of flash memory is not guaranteed. During the processing in the user branch destination, do not download or initialize the on-chip program or initiate programming/erasing program. Programming/erasing at the return from the user branch destination is not guaranteed. In addition, do not modify the data prepared to be programmed. Also during the processing in the user branch destination, do not modify the programming/erasing interface register or make transition to RAM emulation mode. After the user branch processing has completed, use the RTS instruction to return to programming/erasing program.

(c) Flash pass/fail parameter (FPFR: general register R0L of CPU)

This parameter indicates FPFR as the return value of the initialization result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved Return 0
2	BR	—	R/W	User Branch Error Detect Returns the check result whether the specified user branch destination address is in the storage area for the downloaded programming/erasing-related programs. 0: Setting of user branch address is normal 1: Setting of user branch address is abnormal
1	FQ	—	R/W	Frequency Error Detect Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization is completed normally. 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.

1. The start address of the programming destination on the user MAT must be stored in a general register ER1. This parameter is called as FMPAR (flash multipurpose address area parameter).

   Since the program data is always in units of 128 bytes, the lower eight bits (A7 to A0) 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 in other than the flash memory space.

   When data to be programmed does not satisfy 128 bytes, the 128-byte program data must be prepared by filling with the dummy code H′FF.

   The start address of the area in which the prepared program data is stored must be stored in a general register ER0. This parameter is called as FMPDR (flash multipurpose data destination area parameter).

For details on the program processing procedure, see section 20.4.2, User Program Mode.

(a) Flash multipurpose address area parameter (FMPAR: general register ER1 of CPU)

This parameter stores the start address of the programming destination on the user MAT.

When the address in the area other than 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 in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	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 stores 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 in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	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)

This parameter indicates the return value of the program processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved Return 0.
6	MD	—	R/W	Programming Mode Related Setting Error Detect Returns the check result that the error protection state is not entered. For conditions to enter the error protection state, see section 20.5.3, Error Protection. 0: FLER setting is normal (FLER = 0) 1: Programming cannot be performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect 1 is returned to this bit when the specified data could not be written or some flash memory related registers are rewritten at return from the user brunch processing because the user MAT was not erased. 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 boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (programming result is not guaranteed)

Bit	Bit Name	Initial Value	R/W	Description
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY before the start of the programming processing. 0: FKEY setting is normal (FKEY = H′5A) 1: FKEY setting is error (FKEY = value other than H′5A)
3	—	—	—	Reserved Returns 0.
2	WD	—	R/W	Write Data Address Detect When the following address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs. • The address in the on-chip RAM where programming/erasing program is downloaded. • The address in the flash memory area 0: Setting of write data address is normal 1: Setting of write data address is abnormal
1	WA	—	R/W	Write Address Error Detect When the following items are specified as the start address of the programming destination, an error occurs. • When the programming destination address in the area other than flash memory is specified • At the dual bank programming, both of the destination address is specified in the same bank. 0: Setting of programming destination address is normal 1: Setting of programming destination address is abnormal
0	SF	—	R/W	Success/Fail Indicates whether the program processing is ended normally or not. 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 20.4.2, 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	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	—	Reserved These bits should be cleared to 0.
7	EB7	—	R/W	Erase Block 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.
6	EB6	—	R/W	
5	EB5	—	R/W	
4	EB4	—	R/W	
3	EB3	—	R/W	
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

(b) Flash pass/fail parameter (FPFR: general register R0L of CPU)

This parameter returns value of the erasing processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved Return 0.
6	MD	—	—	Erasure Mode Related Setting Error Detect Returns the check result of whether the error protection state is entered. For conditions to enter the error protection state, see section 20.5.3, Error Protection. 0: FLER setting is normal (FLER = 0) 1: FLER = 1 and erasure cannot be performed
5	EE	—	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed at the return from the user branch processing. 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 boot mode or programming mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before start of the erasing processing. 0: FKEY setting is normal (FKEY = H′5A) 1: FKEY setting is error (FKEY = value other than H′5A)
3	EB	—	R/W	Erase Block Select Error Detect Returns the check result whether the specified erase-block number is in the block range of the user MAT. 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2	—	—	—	Reserved
1	—	—	—	Return 0.

Bit	Bit Name	Initial Value	R/W	Description
0	SF	—	R/W	Success/Fail Indicates whether the erasing processing is ended normally or not. 0: Erasure is ended normally (no error) 1: Erasure is ended abnormally (error occurs)

20.3.3 RAM Emulation Register (RAMER)

When the real-time programming of the user MAT is emulated, RAMER sets the area of the user MAT which is overlapped with a part of the on-chip RAM. RAMER is initialized to H′00 at a power-on reset or in hardware standby mode and is not initialized in software standby mode. The RAMER setting must be executed in user mode or in user program mode.

For the division method of the user-MAT area, see table 20.6. In order to operate the emulation function certainly, the target MAT of the RAM emulation must not be accessed immediately after RAMER is programmed. If it is accessed, the normal access is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RAMS	—	R/W	RAM Select Sets whether the user MAT is emulated or not. When RAMS = 1, all blocks of the user boot MAT are in the programming/erasing protection state. 0: RAM emulation function is invalid All blocks of the user boot MAT are not in the programming/erasing protection state 1: RAM emulation function is valid All blocks of the user boot MAT are in the programming/erasing protection state
2	RAM2	0	R/W	User MAT Area Select
1	RAM1	0	R/W	These bits are used with bit 3 and select the user-MAT area to be overlapped with the on-chip RAM. (See table 20.6.)
0	RAM0	0	R/W	

Table 20.6 Division of User MAT Area

RAM Area	Block Name	RAMS	RAM2	RAM1	RAM0
H′FFD000 to H′FFDFFF	RAM area (4 kbytes)	0	*	*	*
H′000000 to H′000FFF	EB0 (4 kbytes)	1	0	0	0
H′001000 to H′001FFF	EB1 (4 kbytes)	1	0	0	1
H′002000 to H′002FFF	EB2 (4 kbytes)	1	0	1	0
H′003000 to H′003FFF	EB3 (4 kbytes)	1	0	1	1
H′004000 to H′004FFF	EB4 (4 kbytes)	1	1	0	0
H′005000 to H′005FFF	EB5 (4 kbytes)	1	1	0	1
H′006000 to H′006FFF	EB6 (4 kbytes)	1	1	1	0
H′007000 to H′007FFF	EB7 (4 kbytes)	1	1	1	1

Note: Don’t care

20.3.4 Flash Vector Address Control Register (FVACR)

FVACR modifies the space from which the vector table data of the NMI interrupts is read. Normally the vector table data is read from the address spaces from H′00001C to H′00001F. However, the vector table can be read from the internal I/O register (FVADDR 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 the NMI interrupt is necessary such as in the system error processing, FVACR and FVADDR to FVADRL must be set and the interrupt exception processing routine must be set in the on-chip RAM space or in the external space.

Bit	Bit Name	Initial Value	R/W	Description
7	FVCHGE	0	R/W	Vector Switch Function Valid Selects whether the function for modifying the space from which the vector table data is read is valid or invalid. When FVCHGE = 1, the vector table data can be read from the internal I/O register (FVADDR to FVADRL). 0: Function for modifying the space from which the vector table data is read is invalid (Initial value) 1: Function for modifying the space from which the vector table data is read is valid
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.5 Flash Vector Address Data Register (FVADR)

This is a register to store the vector data when the flash vector address control register (FVACR) is used to enable the function to select the space where the vector table data is read. This register consists of four 8-bit registers: FVADRR, FVADRE, FVADRH, and FVARL. This register is initialized to H′00000000 at a power-on reset or in hardware standby mode.

- FVADRR

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R/W	Set the vector address.

- FVADRE

Bit	Bit Name	Initial Value	R/W	Description
23 to 16	—	All 0	R/W	Set the vector address.

- FVADRH

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R/W	Set the vector address.

- FVADRL

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R/W	Set the vector address.

20.4 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 of the pin setting for entering each mode, see table 20.1. For details of the state transition of each mode for flash memory, see figure 20.2.

20.4.1 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 20.6. For details on the pin setting in boot mode, see table 20.1. The NMI and other interrupts are ignored in boot mode. However, the NMI and other interrupts should be disabled in the user system.

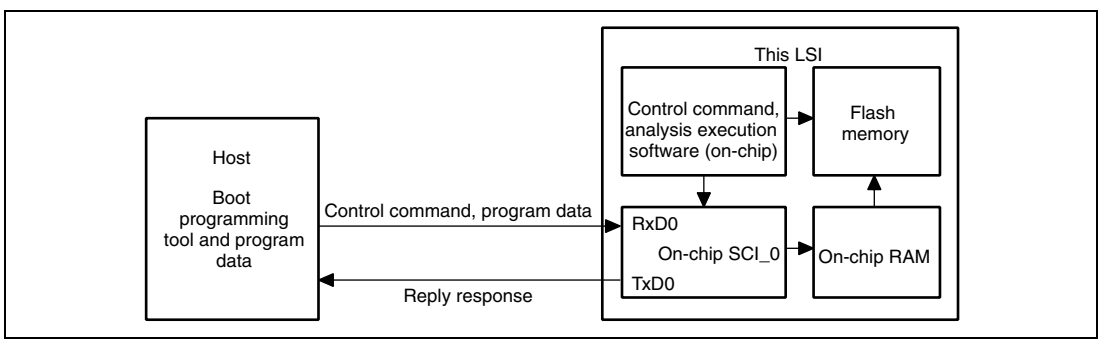


Figure 20.6 System Configuration in Boot Mode

(1) 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 20.7. Boot mode must be initiated in the range of this system clock.

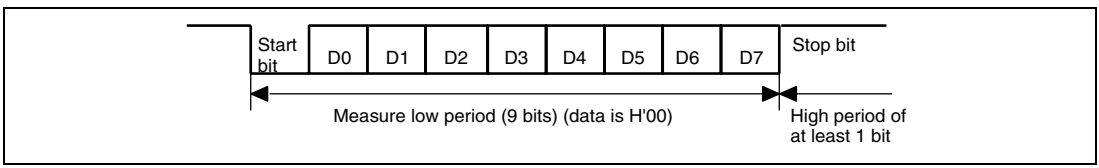


Figure 20.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 20.7 System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI

Bit Rate of Host	System Clock Frequency
9,600 bps	10 to 25 MHz
19,200 bps	16 to 25 MHz

(2) State Transition Diagram

The overview of the state transition diagram after boot mode is initiated is shown in figure 20.8.

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.
3. 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′FFFFFFFF 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 H′FF and transmitted. Then the state for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be used when the specified block is programmed without a reset start 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 programmed data after all user MAT/user boot MAT has automatically been erased.

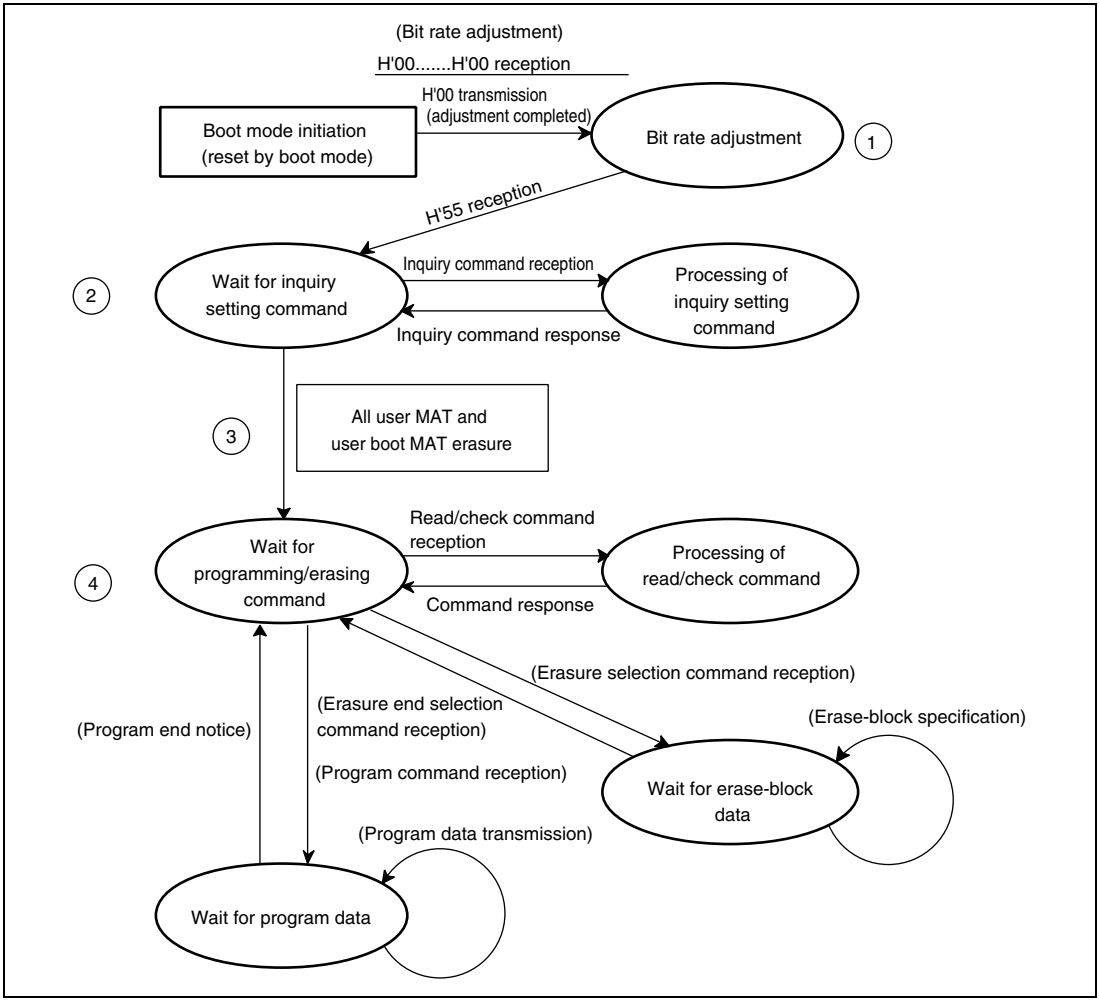


Figure 20.8 Overview of Boot Mode State Transition Diagram

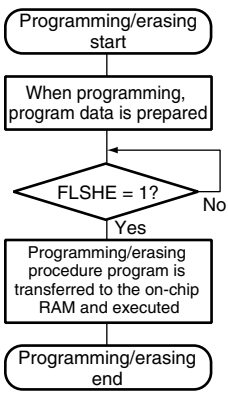
20.4.2 User Program Mode

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 20.9.

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 µs.



1. RAM emulation mode must be canceled in advance. Download cannot be executed in emulation mode.
2. When the program data is made by means of emulation, change the download destination by the FTDAR register. Note that the download area and the emulation area will overlap if FTDAR is H'04.
3. Set FLSHE bit in SYSCR2 register 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 or ROM, the program data can be in an external space.
5. After programming/erasing is finished, the FLSHE bit is cleared to 0 and protected.

Figure 20.9 Programming/Erasing Overview Flow

(1) 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 determination 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 20.10 shows the program area to be downloaded.

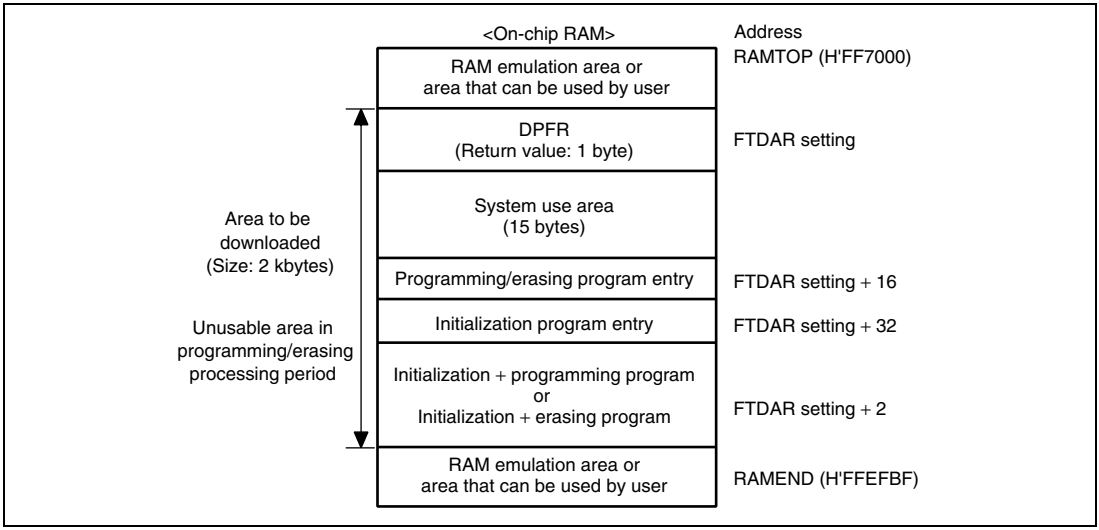


Figure 20.10 RAM Map When Programming/Erasing is Executed

(2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 20.11.

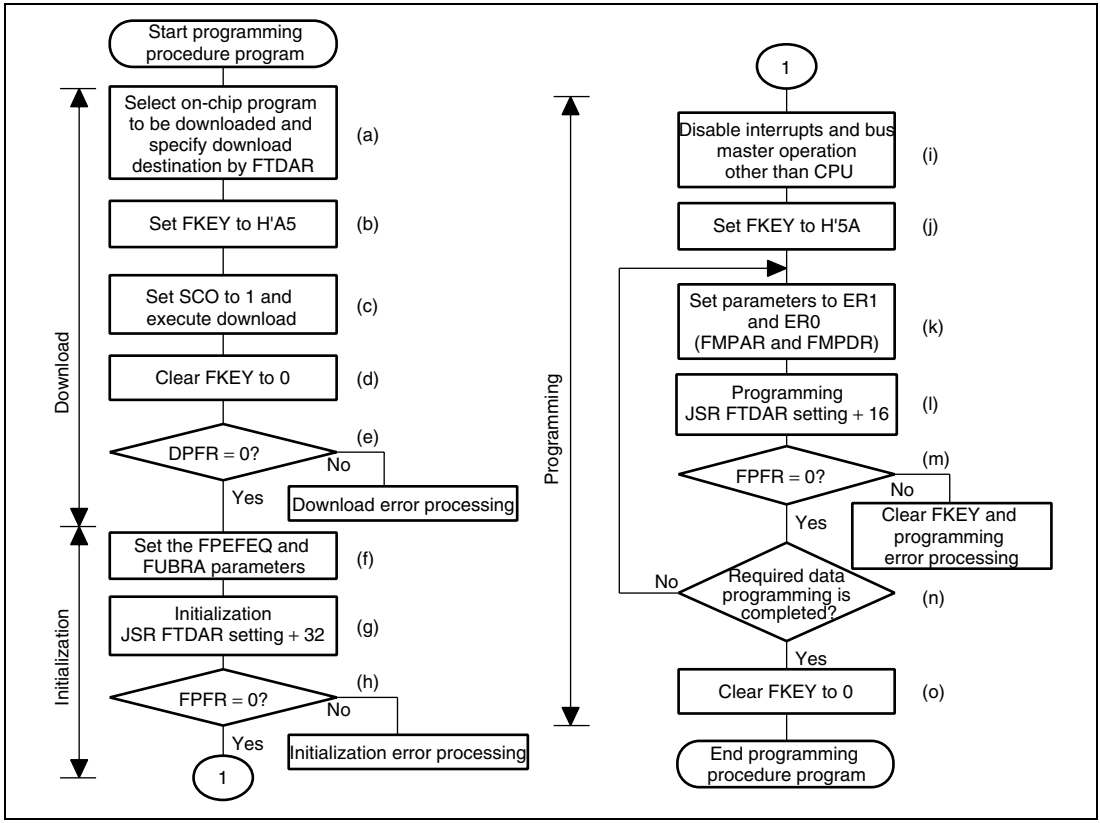


Figure 20.11 Programming Procedure

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 20.4.4, 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 dummy data to be added is H′FF, the program processing period can be shortened.

(a) Select the on-chip program to be downloaded and specify a 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 SS bit in DPFR. The start address of a download destination is specified 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 DPFR. Before the SCO bit is set to 1, incorrect determination must be prevented by setting the one byte of the start address (to be used as DPFR) specified by FTDAR to a value other than the return value (H′FF).

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 FTDAR setting are checked, the transfer processing to the on-chip RAM specified by FTDAR is executed.
— 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 of general registers of the CPU are held.
— In the download processing, any interrupts are not accepted. However, interrupt requests other than the NMI are held. Therefore, when the user procedure program is returned, the NMI interrupts occur.
— The sources of the interrupt requests from the on-chip module and at the falling edge of the IRQ are held during downloading.
— When the level-detection interrupt requests are to be held, interrupts must be input 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-byte is used, the area must be allocated before setting the SCO bit to 1.
— If a flash memory access by the DTC is requested during downloading, the operation cannot be guaranteed. Therefore, an access request by the DTC must not be generated.

(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.

— 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 and the FK bit in the DPFR parameter to ensure that the download program selection and FKEY 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 24.4.2, Clock Timing. When the frequency is set to 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 20.3.2 (2) (a), Flash programming/erasing frequency parameter (FPEFEQ).

— The start address of the user branch destination is set to the FUBRA parameter (general register ER1). This LSI requires FUBRA to be set to 0. When user branch is executed, the branch destination should be other than the user MAT to be programmed. The setting to the area of the downloaded on-chip program is impossible. Use the RTS instruction to return from the user branch processing. For details, see Flash User Branch Address Setting Parameter (FUBRA: General register ER1 of CPU) in section 20.3.2 (2) (b).

(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 the start address specified by FTDAR + 32 bytes of the on-chip RAM. 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 held 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 allocated 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 determined.

(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, the voltage for more than the specified time will be applied and flash memory may be damaged. Therefore, interrupts and movement of bus mastership other than the CPU are prohibited.

To prohibit the interrupt, bit 7 (I) in the condition code register (CCR) of the CPU should be set to B′1 in interrupt control mode 0. Then interrupts other than NMI are held and are not executed.

The NMI interrupts must be masked within the user system.

The interrupts that are held must be executed in the user branch destination or after all program processing. When the interrupt processing is executed in the user branch destination, prohibit the interrupt in the CCR register of the CPU after the process completes.

When the movement of bus mastership to other than the CPU, error protection state is entered. Therefore, prevent other than the CPU from getting bus, as is the case with interrupt prohibition.

(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 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 H′00 or H′80 as the boundary of 128 bytes.

— 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.

(l) Programming

There is an entry point of the programming program in the area from the start address specified by FTDAR + 16 bytes of the 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 held in the programming program.

— R0L 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 allocated in RAM

(m) The return value in the programming program, FPFR (general register R0L) is determined.

(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 (n). 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 µs.

(3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 20.12.

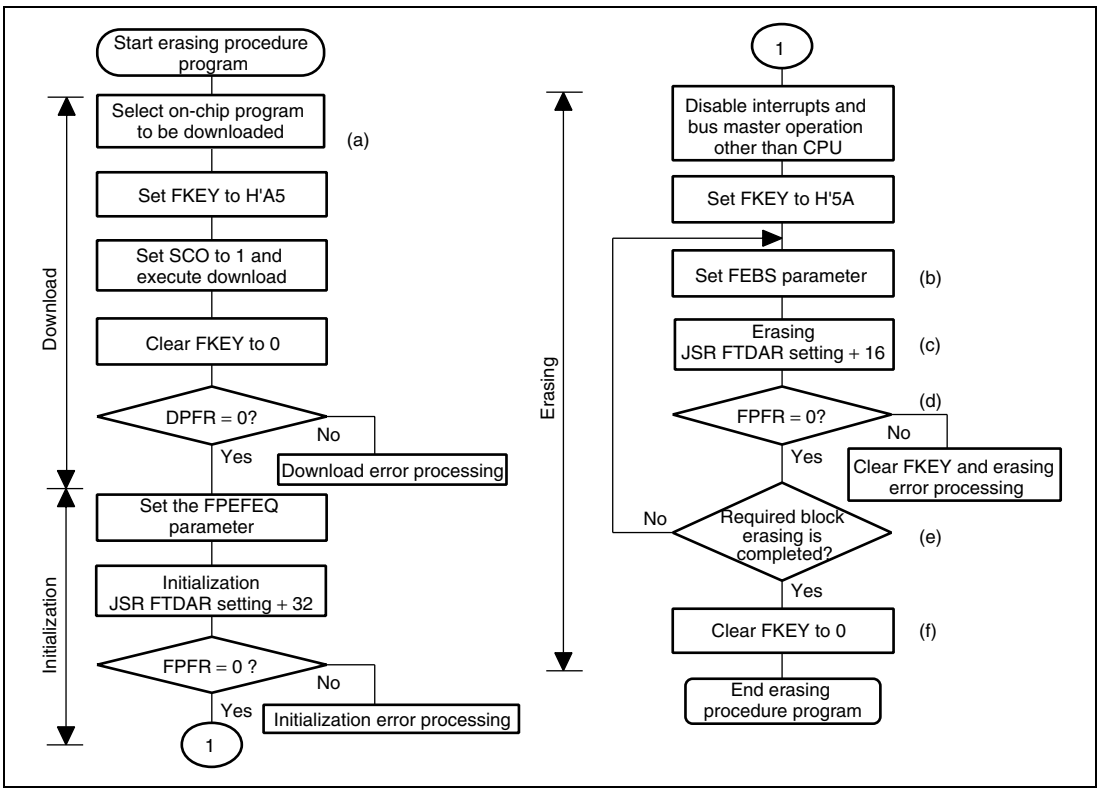


Figure 20.12 Erasing Procedure

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 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 20.4.4, 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 20.10.

A single divided block is erased by one erasing processing. For block divisions, refer to figure 20.4. 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 reported to the SS bit in the DPFR parameter.

Specify the start address of a download destination by FTDAR.

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 20.4.2, (2).

The procedures after setting parameters for erasing programs are as follows:

(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 the start address of a download destination specified 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 held in the erasing program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a stack area of maximum 128 bytes must be allocated in RAM

(d) The return value in the erasing program, FPFR (general register R0L) is determined.

(e) Determine whether erasure of the necessary blocks has completed.

If more than one block is to be erased, update the FEBS parameter and repeat steps (b) to (e). Blocks that have already been erased can be erased again.

(f) After erasure completes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT erasure has completed, secure a reset period (period of $\overline{RES}$ = 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 20.13 shows an example of repetitively executing RAM emulation, erasing, and programming.

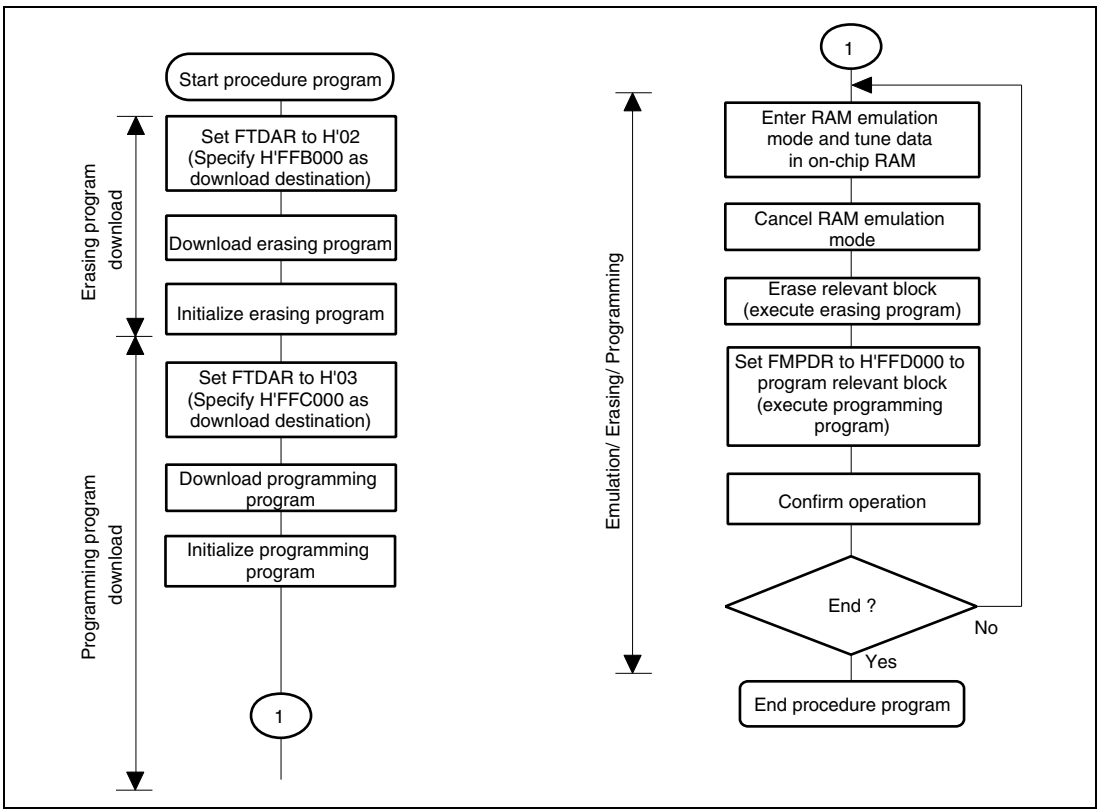


Figure 20.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Programming (Overview)

In the above example, the erasing program and programming program are downloaded to areas excluding the 4 kbytes (H'FFD000 to H'FFDFFF) in the on-chip RAM. 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 parameter 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′FFB040 in this example) and (download start address for programming program) + 32 bytes (H′FFC040 in this example).

20.4.3 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.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 20.1.

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 FMATS because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing 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.

Figure 20.14 shows the procedure for programming the user MAT in user boot mode.

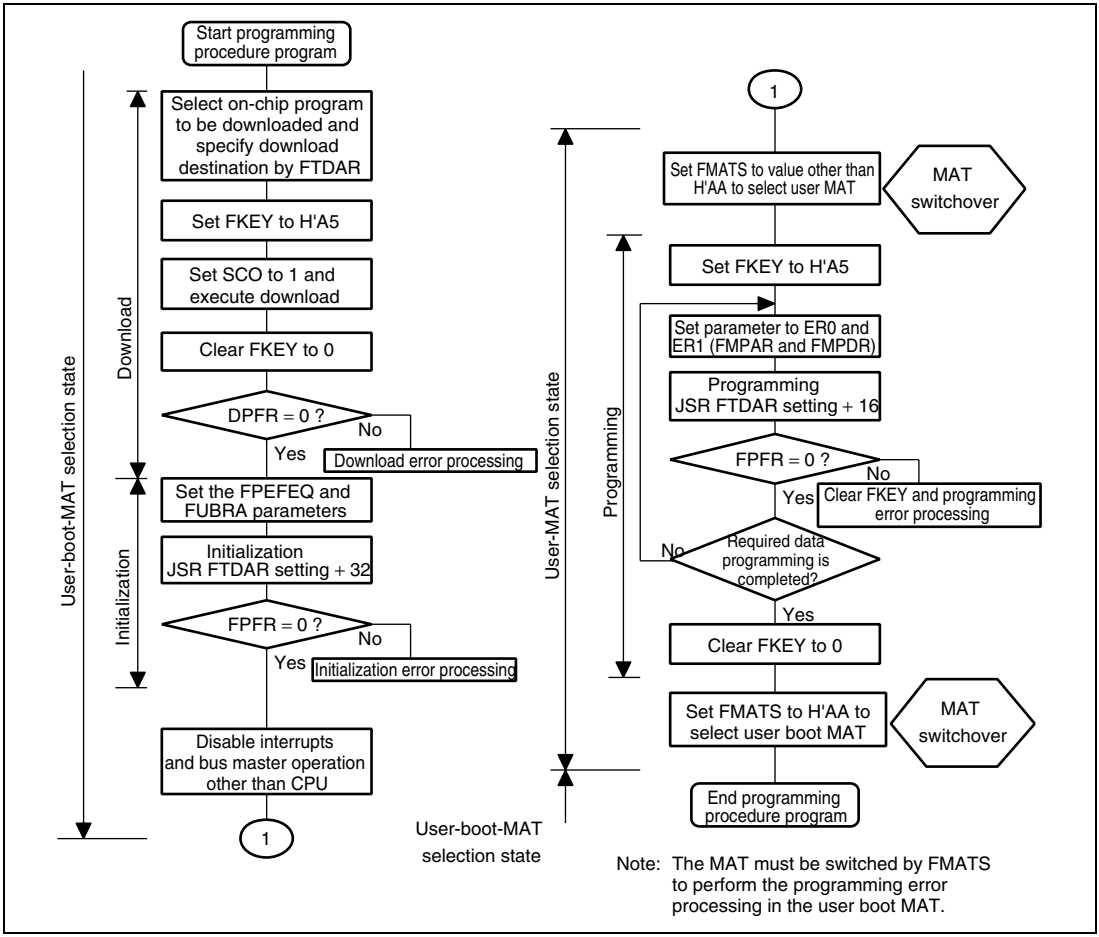


Figure 20.14 Procedure for Programming User MAT in User Boot Mode

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 20.14.

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 completes, switch the MATs again to return to the first state.

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, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 20.7, 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 20.4.4, Procedure Program and Storable Area for Programming Data.

(3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processing 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 20.15 shows the procedure for erasing the user MAT in user boot mode.

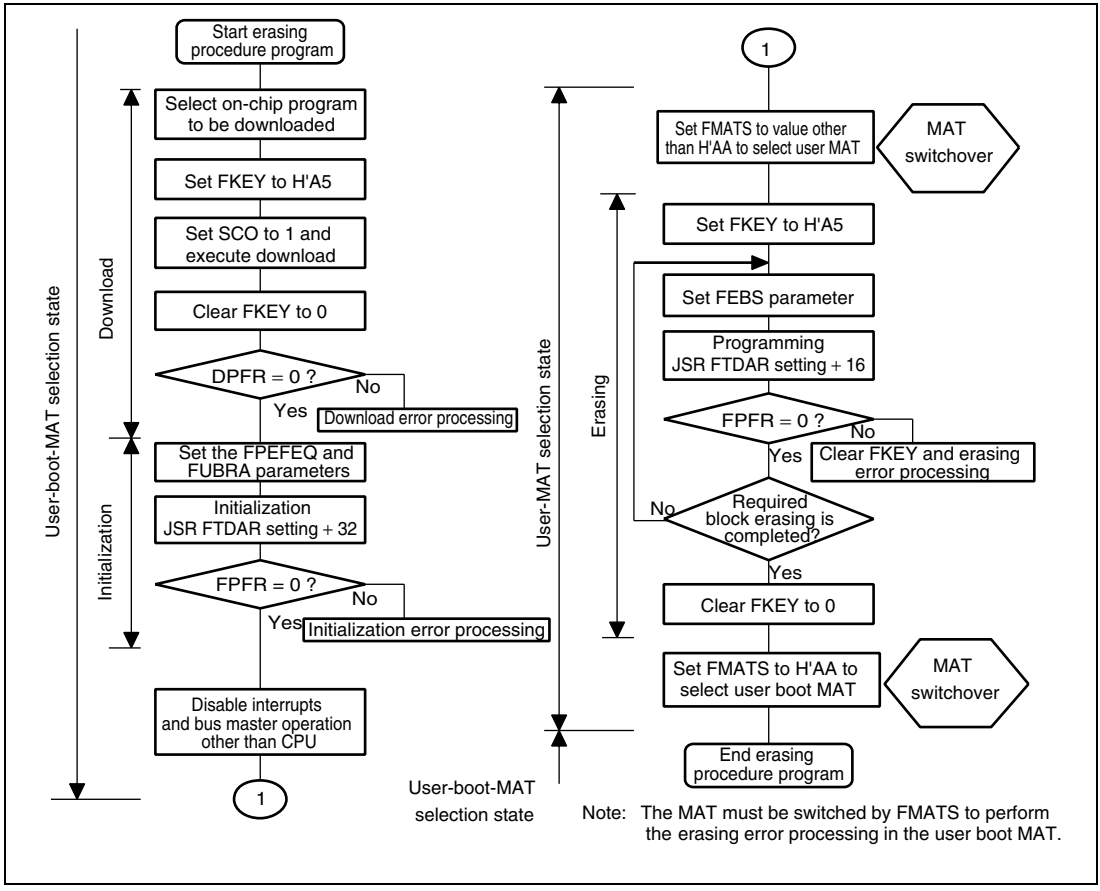


Figure 20.15 Procedure for Erasing User MAT in User Boot Mode

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 20.15.

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, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 20.7, 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 20.4.4, Procedure Program and Storable Area for Programming Data.

20.4.4 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.

(1) Conditions that Apply to Programming/Erasing

1. The on-chip programming/erasing program is downloaded from the address in the on-chip RAM specified by FTDAR, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes as a stack. So, make sure that this area is secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefore, this operation is used, it should be executed from the 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 determined. 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, 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 µs 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 20.7, 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 tables.

Table 20.8 Executable MAT

	Initiated Mode	
Operation	**User Program Mode**	**User Boot Mode***
Programming	Table 20.9 (1)	Table 20.9 (3)
Erasing	Table 20.9 (2)	Table 20.9 (4)

Note: * Programming/Erasing is possible to user MATs.

Table 20.9 (1) Useable Area for Programming in User Program Mode

	Storable /Executable Area			Selected MAT	
Item	**On-chip RAM**	**Target Flash Memory**	**External Space (Expanded Mode)**	**User MAT**	**Embedded Program Storage Area**
Storage Area for Program Data	○	×*	○	—	—
Operation for Selection of On-chip Program to be Downloaded	○	○	○	○	
Operation for Writing H′A5 to FKEY	○	○	○	○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	×	×		○
Operation for FKEY Clear	○	○	○	○	
Determination of Download Result	○	○	○	○	
Operation for Download Error	○	○	○	○	
Operation for Settings of Initial Parameter	○	○	○	○	
Execution of Initialization	○	×	×	○	
Determination of Initialization Result	○	○	○	○	
Operation for Initialization Error	○	○	○	○	
NMI Handling Routine	○	×	○	○	
Operation for Inhibit of Interrupt	○	○	○	○	
Operation for Writing H′5A to FKEY	○	○	○	○	
Operation for Settings of Program Parameter	○	×	○	○	
Execution of Programming	○	×	×	○	
Determination of Program Result	○	×	○	○	

Item	Storable /Executable Area			Selected MAT	
	On-chip RAM	Target Flash Memory	External Space (Expanded Mode)	User MAT	Embedded Program Storage Area
Operation for Program Error	○	×	○	○	
Operation for FKEY Clear	○	×	○	○	

Note: Transferring the data to the on-chip RAM enables this area to be used.

Table 20.9 (2) Useable Area for Erasure in User Program Mode

Item	Storable /Executable Area			Selected MAT	
	On-chip RAM	Target Flash Memory	External Space (Expanded Mode)	User MAT	Embedded Program Storage Area
Operation for Selection of On-chip Program to be Downloaded	○	○	○	○	
Operation for Writing H′A5 to FKEY	○	○	○	○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	×	×		○
Operation for FKEY Clear	○	○	○	○	
Determination of Download Result	○	○	○	○	
Operation for Download Error	○	○	○	○	
Operation for Settings of Initial Parameter	○	○	○	○	
Execution of Initialization	○	×	×	○	
Determination of Initialization Result	○	○	○	○	
Operation for Initialization Error	○	○	○	○	
NMI Handling Routine	○	×	○	○	
Operation for Inhibit of Interrupt	○	○	○	○	

Item	Storable /Executable Area: On-chip RAM	Target Flash Memory	External Space (Expanded Mode)	Selected MAT: User MAT	Embedded Program Storage Area
Operation for Writing H′5A to FKEY	○	○	○	○	
Operation for Settings of Erasure Parameter	○	×	○	○	
Execution of Erasure	○	×	×	○	
Determination of Erasure Result	○	×	○	○	
Operation for Erasure Error	○	×	○	○	
Operation for FKEY Clear	○	×	○	○	

Table 20.9 (3) Useable Area for Programming in User Boot Mode

Item	Storable/Executable Area: On-chip RAM	User Boot MAT	External Space (Expanded Mode)	Selected MAT: User MAT	User Boot MAT	Embedded Program Storage Area
Storage Area for Program Data	○	×*[1]	○	—	—	—
Operation for Selection of On-chip Program to be Downloaded	○	○	○		○	
Operation for Writing H′A5 to FKEY	○	○	○		○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	×	×			○
Operation for FKEY Clear	○	○	○		○	
Determination of Download Result	○	○	○		○	
Operation for Download Error	○	○	○		○	
Operation for Settings of Initial Parameter	○	○	○		○	
Execution of Initialization	○	×	×		○	

Item	Storable/Executable Area			Selected MAT		
	On-chip RAM	User Boot MAT	External Space (Expanded Mode)	User MAT	User Boot MAT	Embedded Program Storage Area
Determination of Initialization Result	○	○	○		○	
Operation for Initialization Error	○	○	○		○	
NMI Handling Routine	○	×	○		○	
Operation for Interrupt Inhibit	○	○	○		○	
Switching MATs by FMATS	○	×	×	○		
Operation for Writing H′5A to FKEY	○	×	○	○		
Operation for Settings of Program Parameter	○	×	○	○		
Execution of Programming	○	×	×	○		
Determination of Program Result	○	×	○	○		
Operation for Program Error	○	×*2	○	○		
Operation for FKEY Clear	○	×	○	○		
Switching MATs by FMATS	○	×	×		○	

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 20.9 (4) Useable Area for Erasure in User Boot Mode

	Storable/Executable Area			Selected MAT		
Item	**On-chip RAM**	**User Boot MAT**	**External Space (Expanded Mode)**	**User MAT**	**User Boot MAT**	**Embedded Program Storage Area**
Operation for Selection of On-chip Program to be Downloaded	○	○	○		○	
Operation for Writing H′A5 to FKEY	○	○	○		○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	×	×			○
Operation for FKEY Clear	○	○	○		○	
Determination of Download Result	○	○	○		○	
Operation for Download Error	○	○	○		○	
Operation for Settings of Initial Parameter	○	○	○		○	
Execution of Initialization	○	×	×		○	
Determination of Initialization Result	○	○	○		○	
Operation for Initialization Error	○	○	○		○	
NMI Handling Routine	○	×	○		○	
Operation for Interrupt Inhibit	○	○	○		○	
Switching MATs by FMATS	○	×	×	○		
Operation for Writing H′5A to FKEY	○	×	○	○		
Operation for Settings of Erasure Parameter	○	×	○	○		

Item	Storable/Executable Area			Selected MAT		
	On-chip RAM	User Boot MAT	External Space (Expanded Mode)	User MAT	User Boot MAT	Embedded Program Storage Area
Execution of Erasure	○	×	×	○		
Determination of Erasure Result	○	×	○	○		
Operation for Erasure Error	○	×*	○	○		
Operation for FKEY Clear	○	×	○	○		
Switching MATs by FMATS	○	×	×		○	

Note: Switching FMATS by a program in the on-chip RAM enables this area to be used.

20.5 Protection

There are two kinds of flash memory program/erase protection: hardware and software protection.

20.5.1 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 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 20.10 Hardware Protection

		Function to be Protected	
Item	**Description**	**Download**	**Program/Erase**
Reset/standby protection	• The program/erase interface registers are initialized in the power-on reset state (including a power-on reset by the WDT) and standby mode and the program/erase-protected state is entered. • The reset state will not be entered by a reset using the $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized after power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width that is specified in the section on AC characteristics section. If a reset is input during programming or erasure, data values in the flash memory are not guaranteed. In this case, after holding the $\overline{\text{RES}}$ pin low for 100 μs or longer, execute erasure and then execute program again.	○	○

20.5.2 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.

Table 20.11 Software Protection

		Function to be Protected	
Item	**Description**	**Download**	**Program/Erase**
Protection by the SCO bit	• The program/erase-protected state is entered by clearing the SCO bit in FCCS which disables the downloading of the programming/erasing programs.	○	○
Protection by the FKEY register	• Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing.	○	○
Emulation protection	• The program/erase-protected state is entered by setting the RAMS bit in the RAM emulation register (RAMER).	○	○

20.5.3 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 error-protection state is entered, and this aborts the programming or erasure.

The FLER bit is set in the following conditions:

1. When an interrupt such as NMI occurs during programming/erasing.
2. When the 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 the DTC gets bus mastership during programming/erasing.

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 µs 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 20.16 shows transitions to and from the error-protection state.

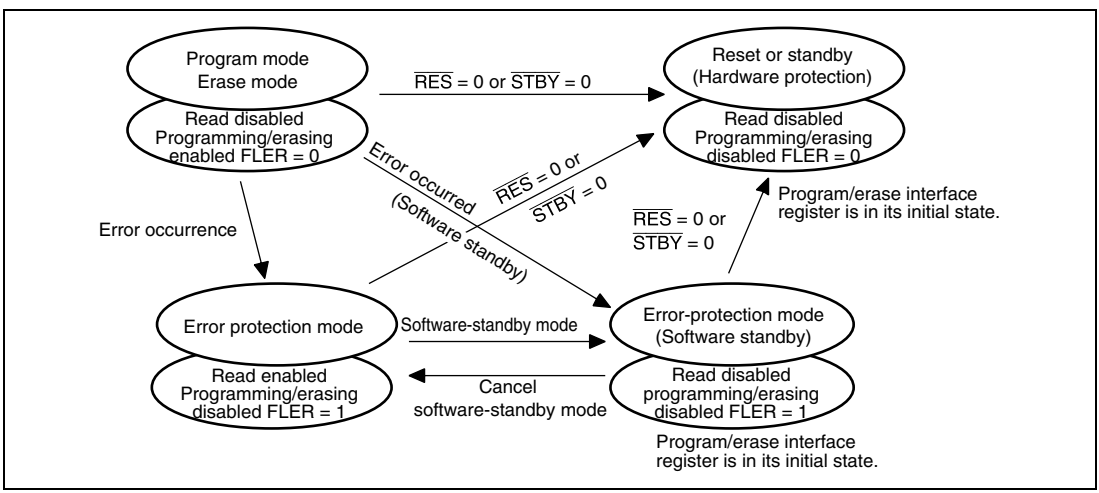


Figure 20.16 Transitions to Error-Protection State

20.6 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 overlapped on an area of flash memory (user MAT) that has been specified by the RAM emulation register (RAMER). The RAM is accessible in both the user MAT area specified by RAMER and as the RAM area that has been overlapped on the user MAT area. Such emulation is possible in both user mode and user-program mode.

Figures 20.17 and 20.18 show an example of the emulation of real-time programming of the user MAT area.

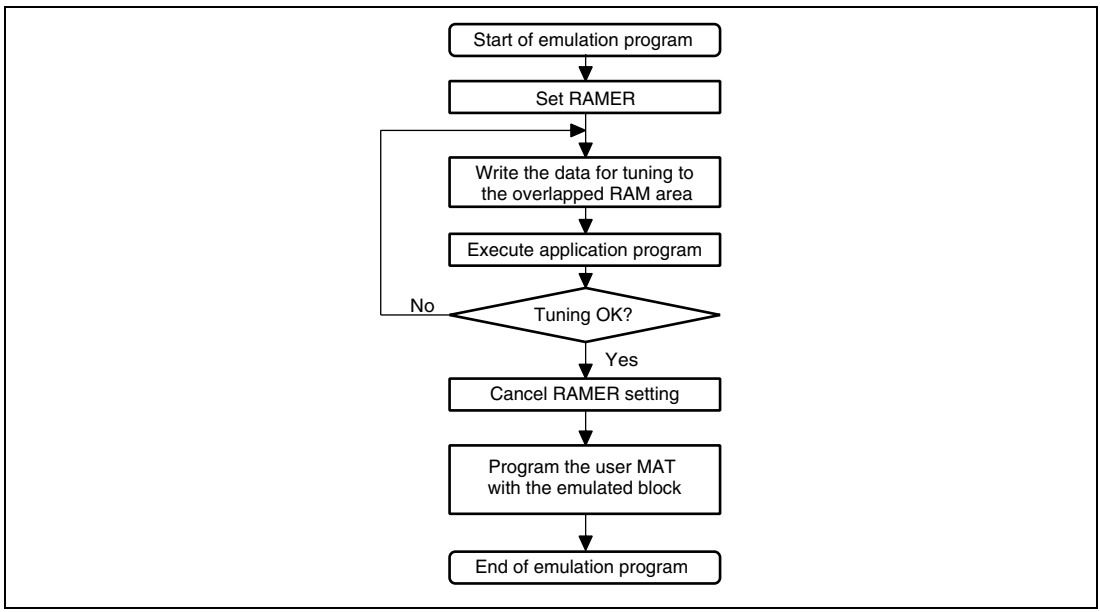


Figure 20.17 Emulation of Flash Memory in RAM

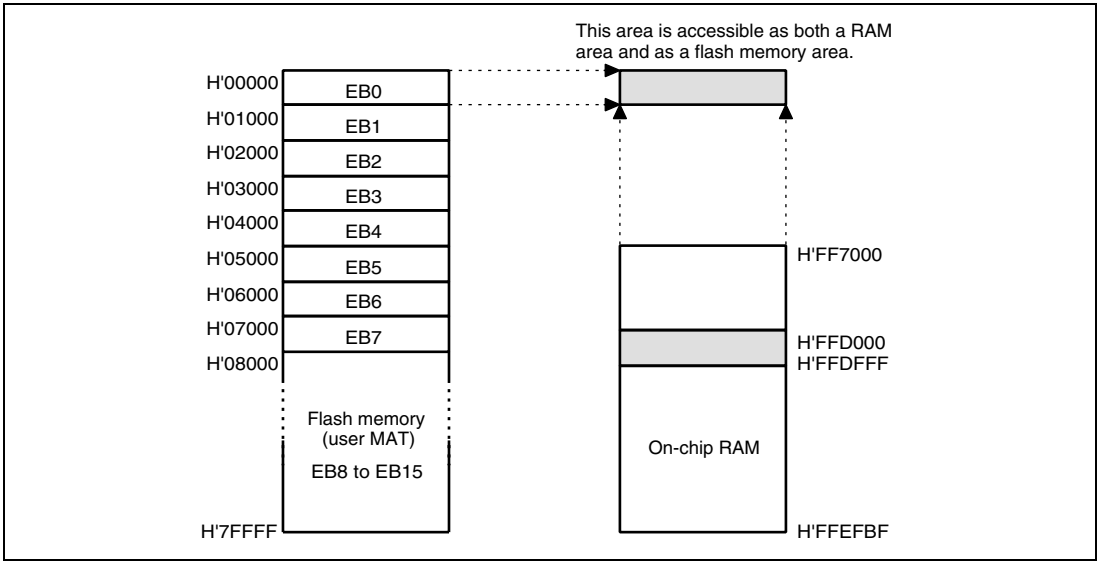


Figure 20.18 Example of a RAM-Overlap Operation

Figure 20.18 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 RAMER.

1. To overlap a part of the RAM on area EB0, to allow real-time programming of the data for this area, set the RAMS bit in RAMER to 1, and each of the RAM2 to RAM0 bits to 0.
2. Real-time programming is carried out using the overlapped 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 an on-chip program. In this process, the overlaid RAM area and the area where the on-chip program is to be downloaded overlap. Therefore, the data that is to be programmed must be saved beforehand in an area that is not used by the system.

Figure 20.19 shows an example of programming of the data, after emulation has been completed, to the EB0 area in the user MAT.

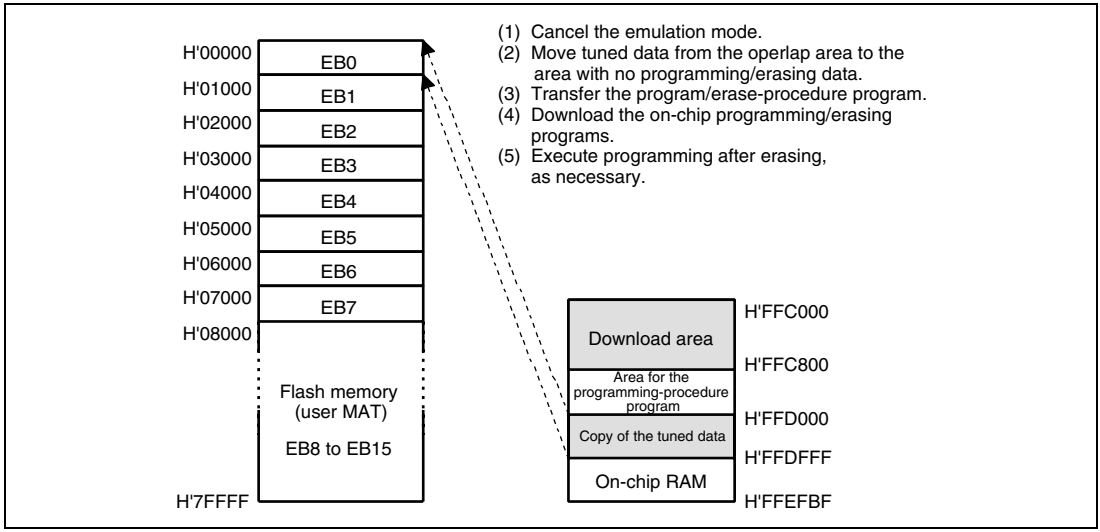


Figure 20.19 Programming of the Data After Tuning

1. After the data to be programmed has fixed values, clear the RAMS bit to cancel the overlap of RAM.
2. Move the fixed programmed data in the overlap area in the RAM to the area to which the programming/erasing program created by the user is transferred and outside the area in which on-chip programs are downloaded.
3. Transfer the user-created programming/erasing-procedure program to the RAM.
4. Run the programming/erasing-procedure program on the RAM and download the on-chip programming/erasing program.
5. When the EB0 area of the user MAT has not been erased, the programming program must be downloaded after erasing. Set the parameters FMPAR and FMPDR in the data to be programmed so that the tuned data that has been saved 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.

20.7 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 programmer mode.)

1. MAT switching by FMATS should always be executed from the on-chip RAM.
2. To ensure that the MAT that has been switched to is accessible, execute four NOP instructions in the on-chip RAM immediately after writing to FMATS 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 FVACR and FVADR 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.

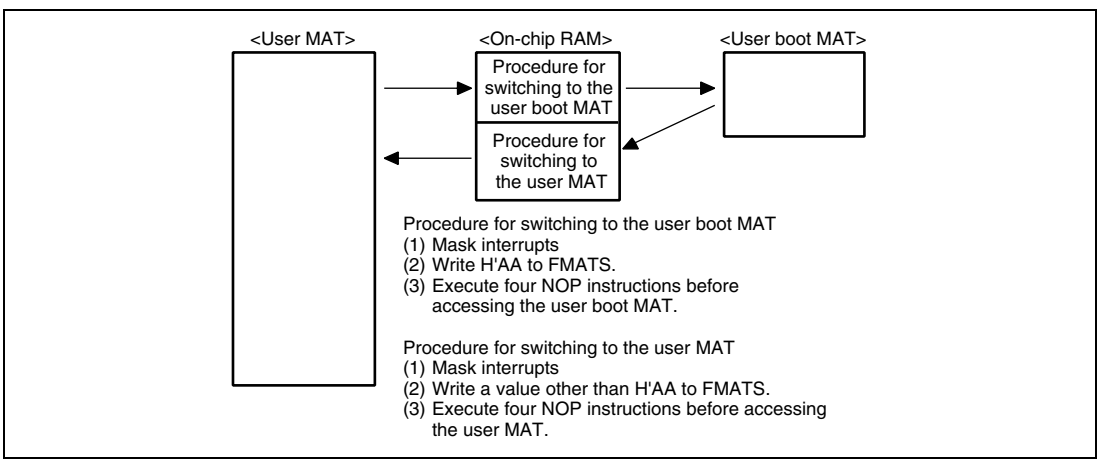


Figure 20.20 Switching between the User MAT and User Boot MAT

20.8 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 TBD µs at maximum.

2. Write to flash-memory related registers by DTC

   While an instruction in on-chip RAM is being executed, the DTC 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 may get out of control. Do not use DTC 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 an 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 LSI.

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 or the periodic timer interrupts) for WDT while taking the programming/erasing time into consideration as required.

20.9 Programmer Mode

Along with its on-board programming mode, this LSI also has a programmer mode as a further mode for the writing and erasing of programs and data. In the programmer 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 Hitachi microcomputers with 512-kbyte flash memory 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 programmer mode, provide a 12-MHz input-clock signal.

20.9.1 Pin Arrangement of Socket Adapter

Figure 20.21 and figure 20.22 show on-chip ROM memory map and socket adapter corresponding map respectively. Attach the socket adapter to the LSI in the way shown in figure 20.22. As a result, conversion to 40 pins is allowed.

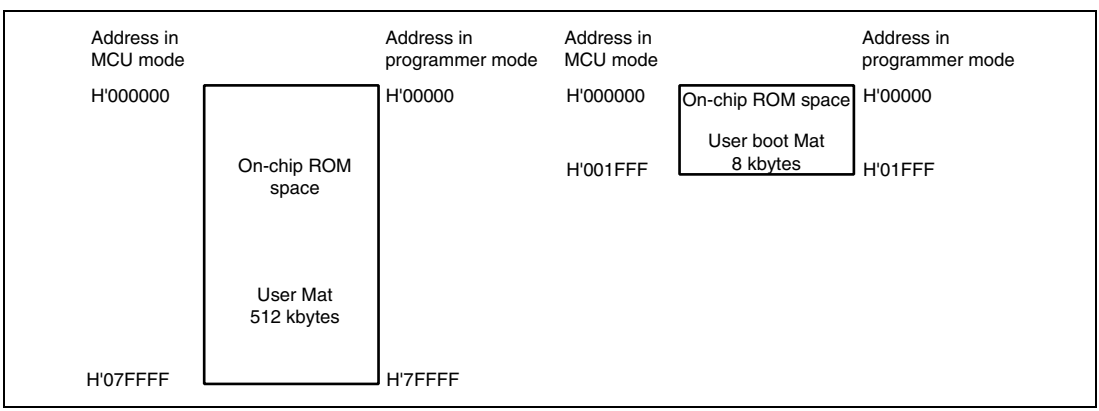


Figure 20.21 On-chip Flash Memory Map

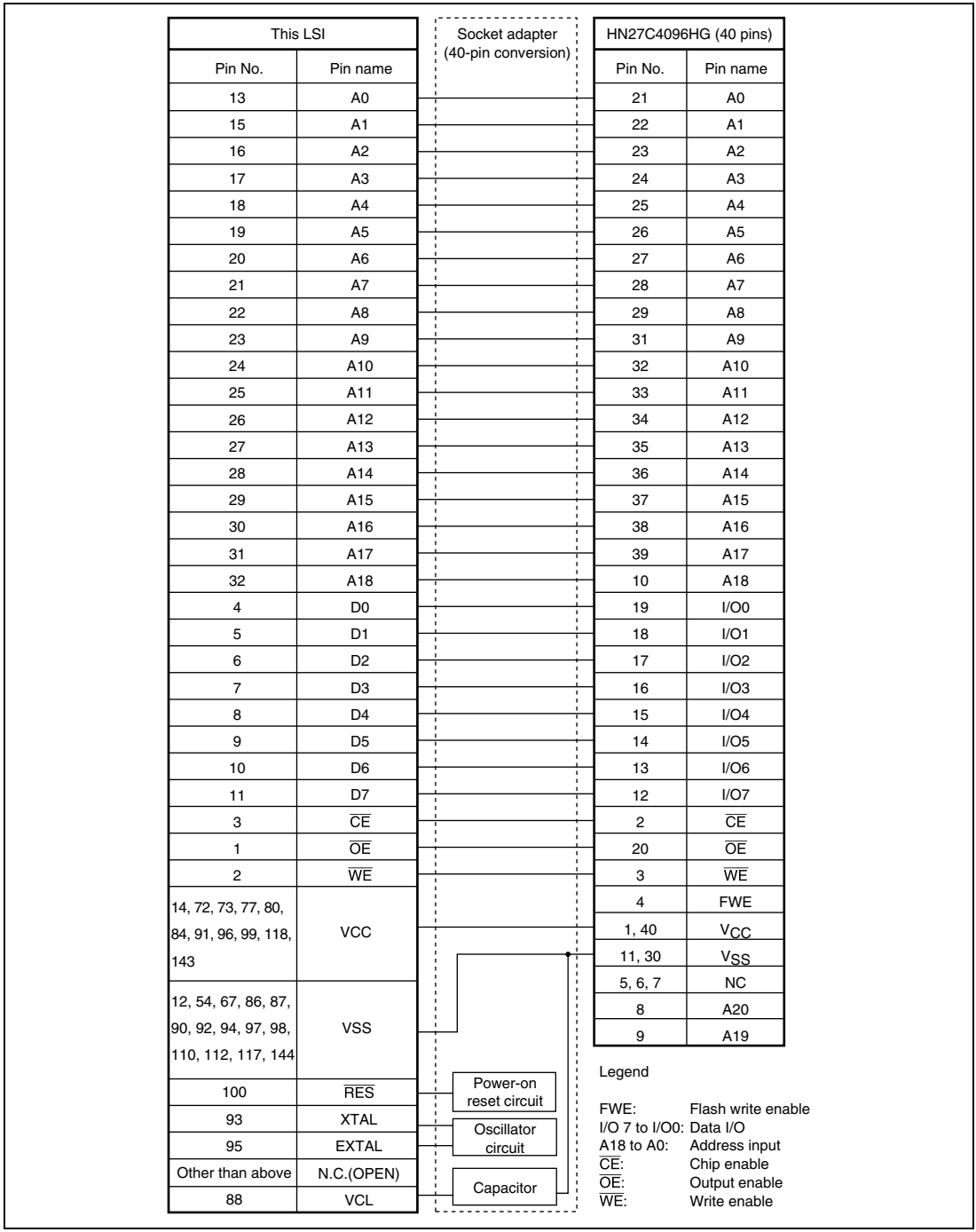


Figure 20.22 Pin arrangement of Socket Adapter

20.9.2 Programmer Mode Operation

Table 20.12 and table 20.13 show setting procedure of each operation mode in programmer mode and each command used in programmer mode, respectively.

- Memory-read mode: Memory read mode supports reading, in bytes, from the user MAT and user boot MAT
- Auto-program mode: Auto-program mode supports the simultaneous programming to the user MAT and user boot MAT in 128-byte units. Status polling is used to confirm the completion of automatic programming.
- Auto-erase mode: Auto-erase mode supports the only automatic erasing of whole user MAT and user boot MAT. Status polling is used to confirm the completion of automatic erasing.
- Status-read mode: Status polling is used with automatic programming/automatic erasure. Normal completion can be confirmed by reading the signal on the I/O 6. In status-read mode, error information is output when an error has occurred.

Table 20.12 Setting Procedure of each Operation Mode of Programmer Mode

	Pin Name				
Mode	**$\overline{CE}$**	**$\overline{OE}$**	**WE**	**I/O 7 to 0**	**A 18 to 0**
Read	L	L	H	Data output	Ain
Output disable	L	H	H	Hi-z	X
Command write	L	H	L	Data output	Ain*
Chip disable	H	X	X	Hi-z	X

Notes: Chip disable mode is not a standby state; internally, operating state.

* Ain indicates that address may be input in auto-program mode.

Table 20.13 Each Command in Programmer Mode

Command Name	Cycle Count	Target Memory MAT	First Cycle			Second Cycle		
			Mode	Address	Data	Mode	Address	Data
Memory-read mode	1 + n	User MAT	write	X	H′00	read	RA	Dout
		User boot MAT	write	X	H′05			
Auto- program mode	129	User MAT	write	X	H′40	write	WA	Din
		User boot MAT	write	X	H′45			
Auto-erase mode	2	User MAT	write	X	H′20	write	X	H′20
		User boot MAT	write	X	H′25			
Status-read mode	2	Common to both MAT	write	X	H′71	write	X	H′71

Notes: 1. In automatic programming mode, 129 cycles of command programming are required because of simultaneous 128-byte programming.

2. In memory-read mode, the number of cycles varies according to the number of address writing cycles (n).

20.9.3 Memory-Read Mode

1. On completion of an automatic program, automatic erase, or status read, the LSI enters a command waiting state. To read the contents of memory after these operations, issue the command to change the mode to memory-read mode before reading from the memory.

2. In memory-read mode, command programming can be performed as in the case of command waiting state.

3. Continuous read can be performed after the transition to memory-read mode is made.

4. Transition to the memory-read mode is made after power has been supplied.
   For the AC characteristics in memory-read mode, see section 20.11, AC Characteristics and Timing in Programmer Mode.

20.9.4 Auto-Program Mode

1. In auto-programming mode, 128-byte simultaneous programming is performed. In this process, 128 bytes of data are transferred in succession.
2. Data transfer of 128 bytes must be performed even in the programming of 128 bytes or less. H′FF should be written to those address that are unnecessary written to.

3. Lower seven bits of the address to be transferred should be set to low. When an address other than valid address is input, programming error is occurred, although memory programming operation is started.
4. The memory address transfer is made in the second cycle. A transfer should not be made in the third cycle or later.
5. Do not write commands while programming is in progress.
6. One time automatic programming should be performed for each 128-byte block of address. Additional programming of the block to the address where already programmed is not possible.
7. To confirm the end of automatic programming, check the signal on I/O6 pin. Confirmation in status-read mode is also possible. (Status polling of the I/O7 pin is used to check the end status of automatic programming.)
8. Information on the pins I/O6 and I/O7 is retained until the next command is written. As long as no command is written, the information can be read by enabling the $\overline{CE}$ and $\overline{OE}$.

For details on the AC characteristics in auto-program mode, see section 20.11, AC Characteristics and Timing in Auto-Program Mode.

20.9.5 Auto-Erase Mode

1. In auto-erase mode, only erasing the entire memory is supported.
2. Command writing should not be preformed during automatic erasing.
3. To confirm the end of automatic erasing, check the signal on the I/O6 pin. Confirmation in status read mode is also possible. (Status polling of the I/O7 pin is used to check the end status of automatic erasure.)
4. Information on the pins I/O6 and I/O7 is retained until the next command is written. As long as other command is written, the information can be read by enabling the $\overline{CE}$ and $\overline{OE}$.

Refer to 20.11, AC Characteristics and Timing in Programmer Mode, for details on AC characteristics in auto-erase mode.

20.9.6 Status-Read Mode

1. Status read mode is used to determine the type of an abnormal end. Use this mode when an abnormal end is occurred in auto-program/auto-erase mode.
2. Return code is maintained until the command programming other than for status-read mode is made.

Table 20.14 lists the return codes of status-read mode.

Refer to 20.11, AC Characteristics and Timing in Programmer Mode, for details on AC characteristics in status-read mode.

Table 20.14 Return Codes in 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 erasure count exceed	Valid address error
Initial value	0	0	0	0	0	0	0	0
Descrip-tion	Normal end: 0 Abnormal end: 1	Command error: 1 Other: 0	Program-ming error: 1 Other: 0	Erase error: 1 Other: 0	—	—	Count exceeded: 1 Other: 0	Valid address error: 1 Other: 0

Note: I/O2 and I/O3 are undefined.

20.9.7 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 in auto-program or auto-erase mode.

Table 20.15 True Value Table of Status Polling Output

Pin Name	In Progress	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0 to I/O5	0	0	0	0

20.9.8 Transition Time to Programmer Mode

When oscillation is not stabilized or programmer mode is being set up, no command can be accepted. Transition to memory-read mode is made after the programmer mode set up time has elapsed. Refer to 20.11, AC Characteristics and Timing in Programmer Mode.

20.9.9 Notes on Programmer Mode

1. To rewrite to an address already programmed, perform automatic programming after automatic erasure.
2. When rewriting is performed to the chip, which is programmed and erased in on-board programming mode with a programmer, it is recommended performing automatic programming after automatic erase.

3. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage flash memory permanently. If a reset is input accidentally, the reset must be released after a reset period of 100 µs which is longer than normal.
4. The initial state of a Hitachi product at shipment is the erased state. For a product whose history of erasing is undefined, automatic erasure for checking the initial state (erased state) and compensating is recommended.
5. In this LSI, production identification mode such as multipurpose EPROM is not supported; device name can not be automatically set to PROM writer.
6. For the PROM programmer suitable for programmer mode in this LSI and its program version, refer to the instruction manual of the socket adapter.

20.10 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.

(1) 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 20.23.

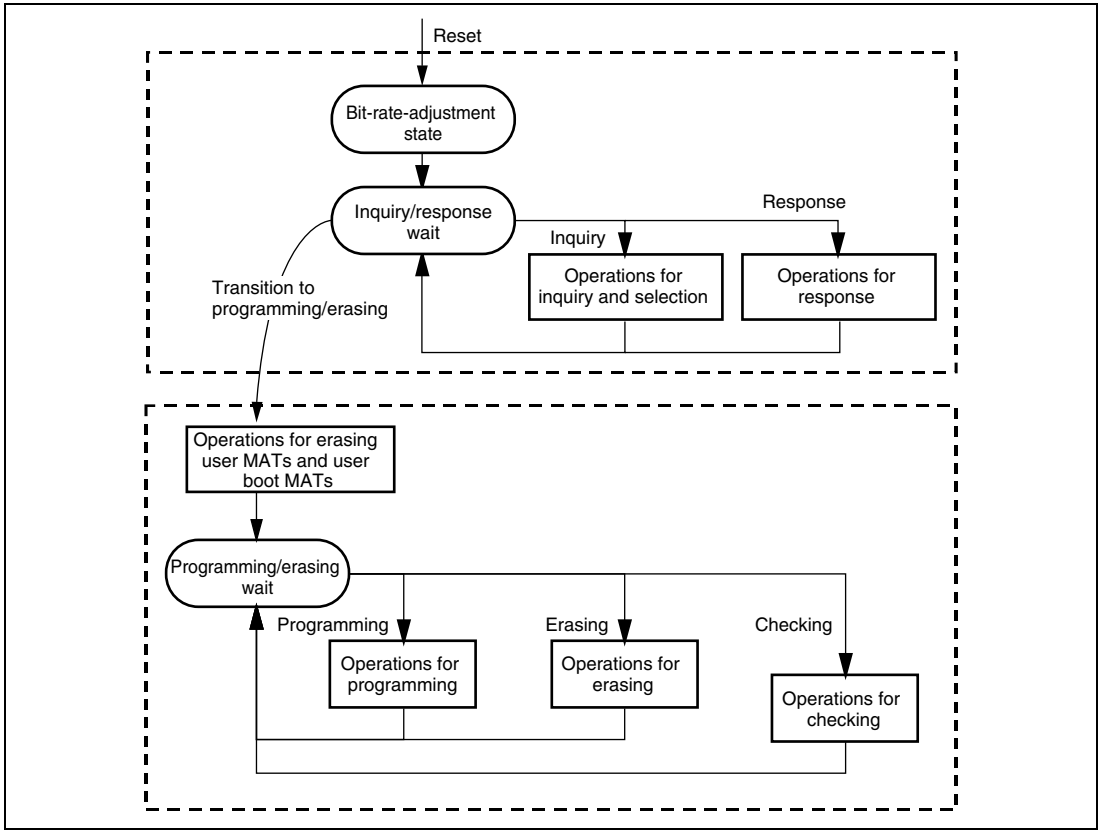


Figure 20.23 Boot Program States

(2) 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 20.24.

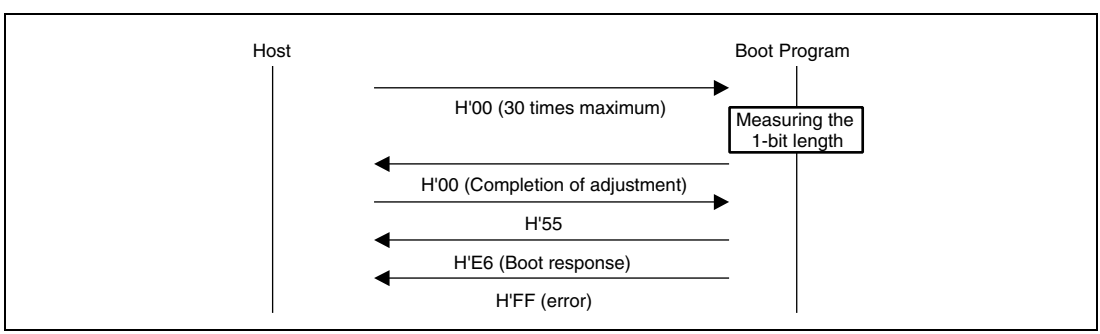


Figure 20.24 Bit-Rate-Adjustment Sequence

(3) 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.

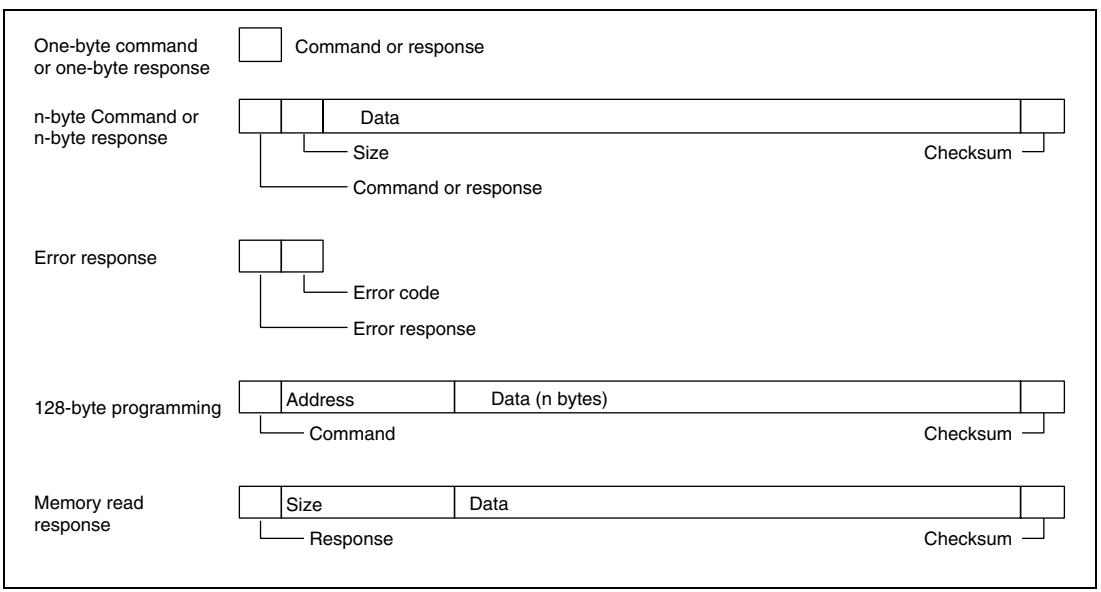


Figure 20.25 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasing, and checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amount of data, and checksum
- Checksum (one 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 (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

(4) 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 20.16 Inquiry and Selection Commands

Command	Command Name	Description
H′20	Supported Device Inquiry	Inquiry regarding device codes
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).

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product code in response to the supported device inquiry.

Command	H′20

- Command, H′20, (one byte): Inquiry regarding supported devices

Response	H′30	Size	Number of devices	
	Number of characters	Device code		Product name
	...			
	SUM			

- Response, H′30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, and checksum, that is, the amount of data contributes by the number of devices, characters, device codes and product names
- Number of devices (one byte): The number of device types supported by the boot program
- Number of characters (one byte): The number of characters in the device codes and boot program's name
- Device code (two bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one 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.

(b) 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, (one byte): Device selection
- Size (one byte): Amount of device-code data
  This is fixed at 2
- Device code (two bytes): Device code (ASCII code) returned in response to the supported device inquiry
- SUM (one byte): Checksum

Response	H′06

- Response, H′06, (one byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response	H′90	ERROR

- Error response, H′90, (one byte): Error response to the device selection command
  ERROR : (one byte): Error code
  H′11: Sum check error
  H′21: Device code error, that is, the device code does not match

(c) 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, (one byte): Inquiry regarding clock mode

Response	H′31	Size	Number of modes	Mode	···	SUM

- Response, H′31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the number of modes and modes
- Number of clock modes (one byte): The number of supported clock modes
  H′00 indicates no clock mode or the device allows to read the clock mode.
- Mode (one byte): Values of the supported clock modes (i.e. H′01 means clock mode 1.)
- SUM (one byte): Checksum

(d) 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, (one byte): Selection of clock mode
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): A clock mode returned in reply to the supported clock mode inquiry.
- SUM (one byte): Checksum

Response	H′06

- Response, H′06, (one 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, (one byte): Error response to the clock mode selection command
- ERROR, (one byte): Error code

  H′11: Checksum error
  H′22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H′00 and H′01 by a clock mode inquiry, the clock mode must be selected using these respective values.

(e) Multiplication Ratio Inquiry

The boot program will return the supported multiplication and division ratios.

Command

H′22

- Command, H′22, (one byte): Inquiry regarding multiplication ratio

Response

H′32	Size	Number of types					
Number of multiplication ratios	Multiplica-tion ratio	···					
...							
SUM							

- Response, H′32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of clock sources and multiplication ratios and the multiplication ratios
- Number of types (one 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.)
- Number of multiplication ratios (one 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 (one 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 (one byte): Checksum

(f) 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, (one byte): Inquiry regarding operating clock frequencies

Response	H′33	Size	Number of operating clock frequencies
	Minimum value of operating clock frequency		Maximum value of operating clock frequency
	...		
	SUM		

- Response, H′33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum values, and the number of frequencies.
- Number of operating clock frequencies (one 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 (two 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 (two bytes): Maximum value among the multiplied or divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (one byte): Checksum

(g) User Boot MAT Information Inquiry

The boot program will return the number of user boot MATs and their addresses.

Command	H′24

- Command, H′24, (one byte): Inquiry regarding user boot MAT information

Response	H′34	Size	Number of areas	
	Area-start address			Area-last address
	...			
	SUM			

- Response, H′34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start addresses, and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas
  When user boot MAT areas are consecutive, the number of areas returned is H′01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command	H′25

- Command, H′25, (one byte): Inquiry regarding user MAT information

Response	H′35	Size	Number of areas	
	Start address area			Last address area
	...			
	SUM			

- Response, H′35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start address and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H′01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command	H′26

- Command, H′26, (one byte): Inquiry regarding erased block information

Response	H′36	Size	Number of blocks	
	Block start address			Block last address
	...			
	SUM			

- Response, H′36, (one byte): Response to the number of erased blocks and addresses
- Size (two bytes): The number of bytes that represents the number of blocks, block-start addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command	H′27

- Command, H′27, (one byte): Inquiry regarding programming unit

Response	H′37	Size	Programming unit	SUM

- Response, H′37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fixed to 2
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

(k) 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 1	Multiplication ratio 2	
	SUM			

- Command, H′3F, (one byte): Selection of new bit rate
- Size (one byte): The number of bytes that represents the bit rate, input frequency, number of multiplication ratios, and multiplication ratio
- Bit rate (two 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 (two 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 (one byte): The number of multiplication ratios to which the device can be set.
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for the main operating frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H′04. In this LSI, set 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. In this LSI, set H′01.)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for the peripheral frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H′04. In this LSI, set H′01.)
  (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 In this LSI, set H′01.)
- SUM (one byte): Checksum

Response	H′06

- Response, H′06, (one 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, (one byte): Error response to selection of new bit rate
- ERROR: (one 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.

(5) 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

To facilitate error checking, the value (n) of clock select (CKS) in the serial mode register (SMR), and the value(N) in the bit rate register (BRR), which are found from the peripheral operating clock frequency (ϕ) and bit rate (B), are used to calculate the error rate to ensure that it is less than 4%. If the error is more than 4%, a bit rate error is generated. The error is calculated using the following expression:

$$\text{Error (\%)} = \{[\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2 \times n - 1)}}] - 1\} \times 100$$

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, (one byte): Confirmation of a new bit rate

Response	H′06

- Response, H′06, (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 20.26.

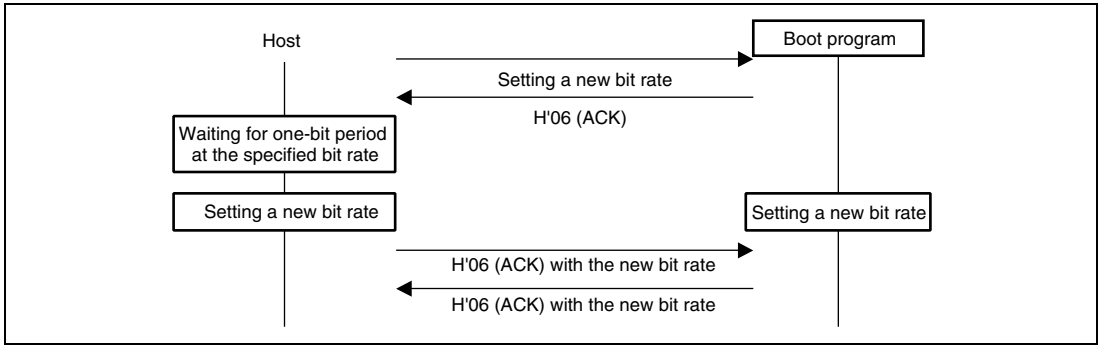


Figure 20.26 New Bit-Rate Selection Sequence

(6) 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, (one byte): Transition to programming/erasing state

Response	H′06

- Response, H′06, (one 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	H′51

- Error response, H′C0, (one byte): Error response for user boot MAT blank check
- Error code, H′51, (one byte): Erasing error
  An error occurred and erasure was not completed.

(7) 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, (one byte): Command error
- Command, H′xx, (one byte): Received command

(8) 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), which are needed for a new bit-rate selection.
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), and programming unit inquiry (H′27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H′40). The boot program will then enter the programming/erasing state.

(9) 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 20.17 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

(10) Programming

Programming is executed by a programming-selection command and a 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 a 128-byte programming command with H′FFFFFFFF 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 20.27.

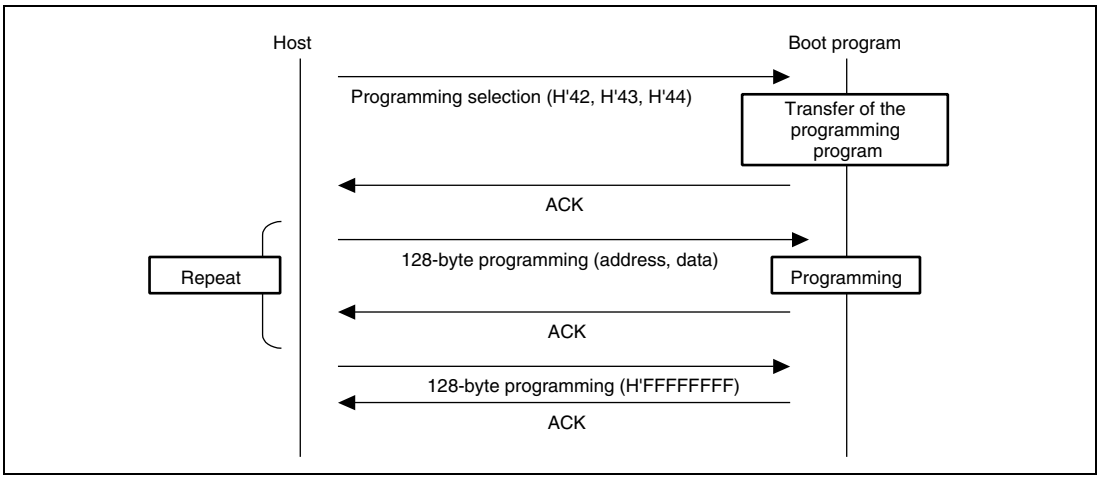


Figure 20.27 Programming Sequence

(a) 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, (one byte): User boot-program programming selection

Response	H′06

- Response, H′06, (one 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)

(b) User-program programming selection

The boot program will transfer a program for programming. The data is programmed to the user MATs by the transferred program for programming.

Command	H′43

- Command, H′43, (one byte): User-program programming selection

Response	H′06

- Response, H′06, (one 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 (one byte): Error response to user MAT programming selection
- ERROR : (one byte): Error code
  H’54: Selection processing error (transfer error occurs and processing is not completed)

(c) 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 in response to 128-byte programming.

Command	H′50	Address						
	Data	···						
	···							
	SUM							

- Command, H′50, (one byte): 128-byte programming
- Programming Address (four 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′01000000)
- Programming Data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response	H′06

- Response, H′06, (one 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, (one byte): Error response for 128-byte programming

- ERROR: (one byte): Error code
  H′11: Checksum Error
  H′28: Address error
  Address is not in the specified MAT.
  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′FFFFFFFF 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, (one byte): 128-byte programming
- Programming Address (four bytes): End code is H′FF, H′FF, H′FF, H′FF.
- SUM (one byte): Checksum

Response	H′06

- Response, H′06, (one 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, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code
  H′11: Sum check error
  H′53: Programming error
  An error has occurred in programming and programming cannot be continued.

(11) 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 20.28.

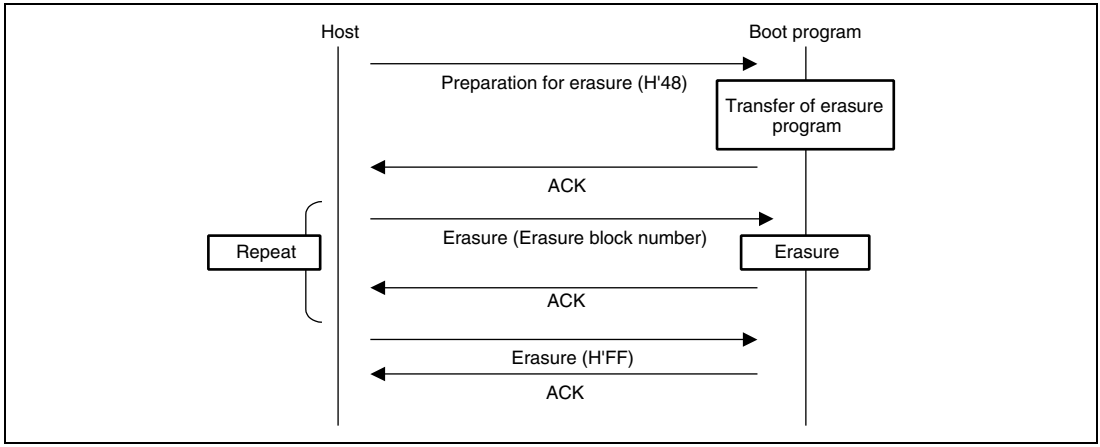


Figure 20.28 Erasure Sequence

(a) 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, (one byte): Erasure selection

Response	H′06

- Response, H′06, (one 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, (one byte): Error response to erasure selection
- ERROR: (one byte): Error code
  H′54: Selection processing error (transfer error occurs and processing is not completed)

(b) Block Erasure

The boot program will erase the contents of the specified block.

Command	H′58	Size	Block number	SUM

- Command, H′58, (one byte): Erasure
- Size (one byte): The number of bytes that represents the erasure block number
  This is fixed to 1.
- Block number (one byte): Number of the block to be erased

- SUM (one byte): Checksum

Response	H′06

- Response, H′06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H′D8	ERROR

- Error Response, H′D8, (one byte): Response to Erasure
- ERROR (one 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, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H′FF
  Stop code for erasure
- SUM (one byte): Checksum

Response	H′06

- Response, H′06, (one 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.

(12) 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.

(13) 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, as a four-byte value.

Command	H′4A

- Command, H′4A, (one byte): Sum check for user-boot program

Response	H′5A	Size	Checksum of user boot program	SUM

- Response, H′5A, (one byte): Response to the sum check of user-boot program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(14) 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, (one byte): Sum check for user program

Response	H′5B	Size	Checksum of user program	SUM

- Response, H′5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(15) 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, (one byte): Blank check for user boot MAT

Response	H′06

- Response, H′06, (one 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, (one byte): Response to blank check for user boot MAT
- Error Code, H′52, (one byte): Erasure has not been completed.

(16) 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, (one byte): Blank check for user MATs

Response	H′06

- Response, H′06, (one 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, (one byte): Error response to the blank check of user MATs.

- Error code, H′52, (one byte): Erasure has not been completed.

(17) 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, (one byte): Inquiry regarding boot program's state

Response	H′5F	Size	Status	ERROR	SUM

- Response, H′5F, (one byte): Response to boot program state inquiry
- Size (one byte): The number of bytes. This is fixed to 2.
- Status (one byte): State of the boot program
- ERROR (one byte): Error state

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (one byte): Checksum
  This command can be accepted during programming/erasing operation, however, response time will be longer.

Table 20.18 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
H′5F	Erasure Block Specification Wait (Erasure is completed)

Table 20.19 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 Incomplete Error
H′53	Programming Error
H′54	Selection Processing Error
H′80	Command Error
H′FF	Bit-Rate-Adjustment Confirmation Error

20.11 AC Characteristics and Timing in Programmer Mode

Table 20.20 AC Characteristics in Memory Read Mode

Conditions: $V_{CC} = 5.0\ V \pm 0.5\ V$, $V_{SS} = 0\ V$, $T_a = 25°C \pm 5°C$

Item	Symbol	Min.	Max.	Unit	Note
Command programming cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ 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	
$\overline{WE}$ rising time	t_r		30	ns	
$\overline{WE}$ falling time	t_f		30	ns	

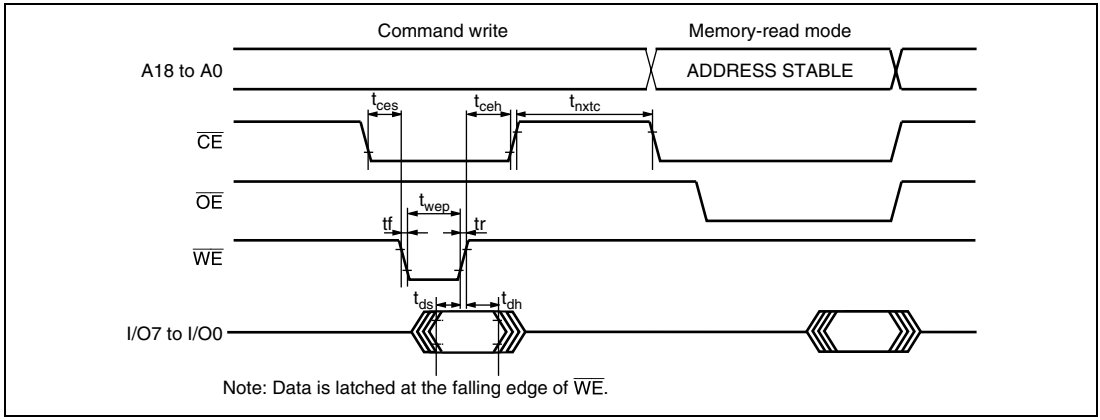


Figure 20.29 Memory Read Timing after Command Programming

Table 20.21 AC Characteristics in Transition from Memory-Read Mode to Other Mode

Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Item	Symbol	Min.	Max.	Unit	Note
Command programming cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ 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	
$\overline{WE}$ rising time	t_r	—	30	ns	
$\overline{WE}$ falling time	t_f	—	30	ns	

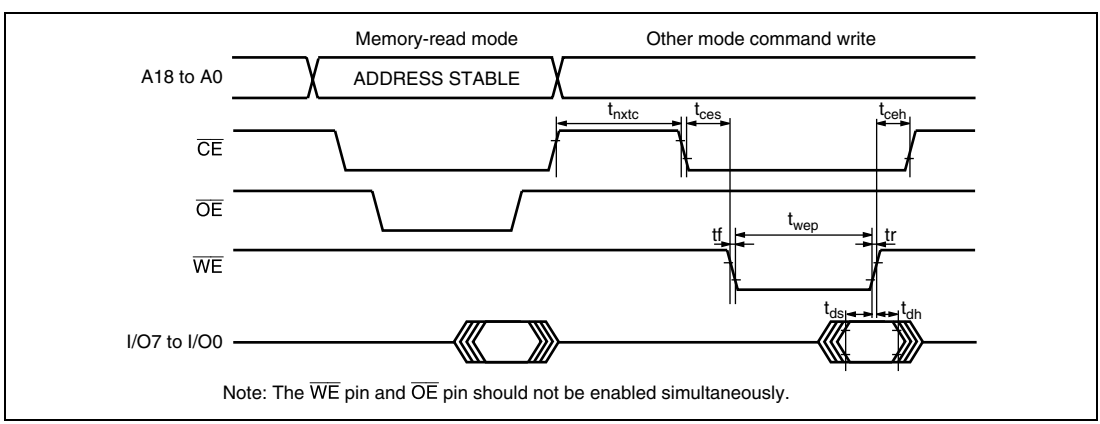


Figure 20.30 Waveform of Transition from Memory-Read Mode to Other Mode

Table 20.22 AC Characteristics in Memory-Read Mode

Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Item	Symbol	Min.	Max.	Unit	Note
Access time	t_{acc}	—	20	µs	
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Output disable delay time	t_{df}	—	100	ns	
Data output hold time	t_{oh}	5	—	ns	

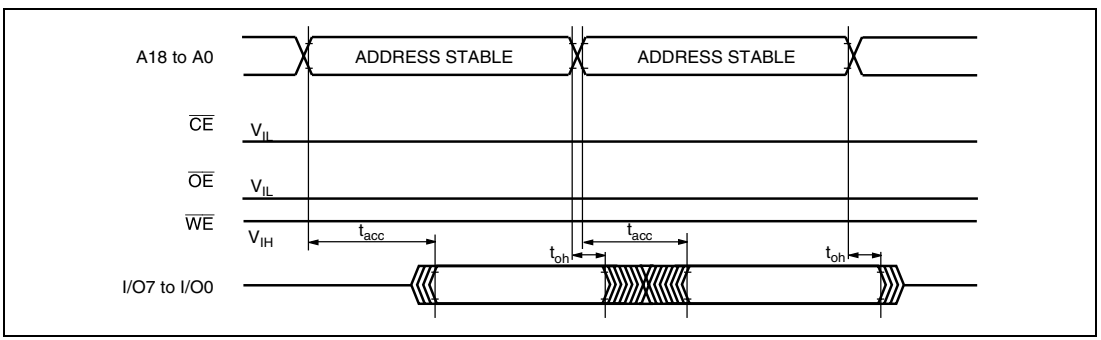


Figure 20.31 Waveform of $\overline{CE}$, $\overline{OE}$ Enable State Read

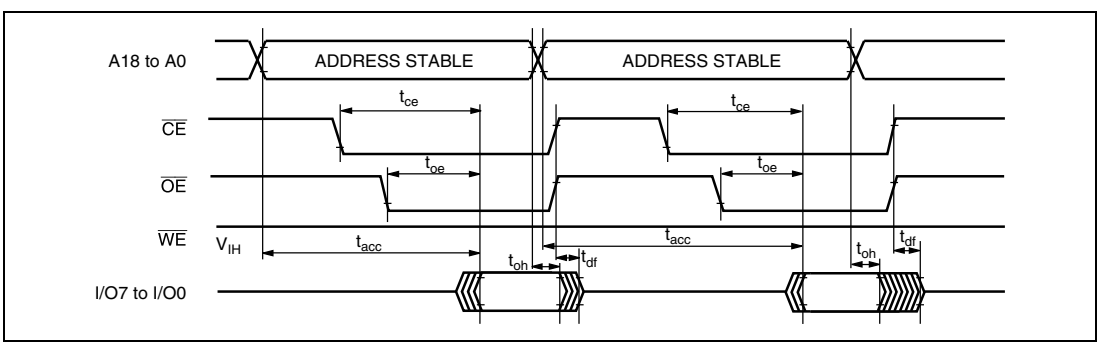


Figure 20.32 Waveform of $\overline{CE}$, $\overline{OE}$ Clock System Read

Table 20.23 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Item	Symbol	Min.	Max.	Unit	Note
Command programming cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ 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	t_{ah}	60	—	ns	
Memory programming time	t_{write}	1	3000	ms	
$\overline{WE}$ rising time	t_r	—	30	ns	
$\overline{WE}$ falling time	t_f	—	30	ns	

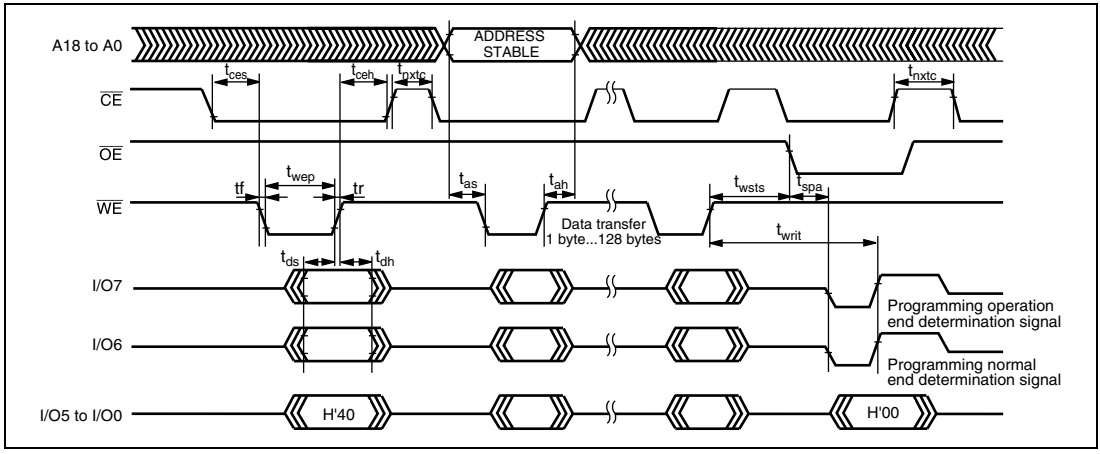


Figure 20.33 Waveform of Automatic Programming Mode

Table 20.24 AC Characteristic in Auto-Erase Mode

Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Item	Symbol	Min.	Max.	Unit	Note
Command programming cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ 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 erasing time	t_{erase}	100	40000	ms	
$\overline{WE}$ rising time	t_r	—	30	ns	
$\overline{WE}$ falling time	t_f	—	30	ns	

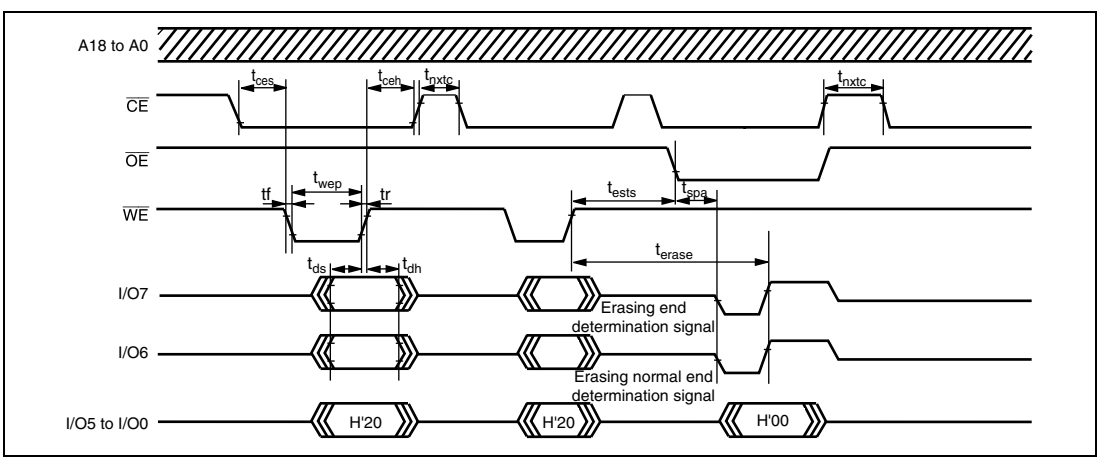


Figure 20.34 Waveform in Auto-Erase Mode

Table 20.25 AC Characteristics in Status-Read Mode

Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Item	Symbol	Min.	Max.	Unit	Note
Read time after command programming	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ 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	
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Disable delay time	t_{df}	—	100	ns	
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	
$\overline{WE}$ rising time	t_r	—	30	ns	
$\overline{WE}$ falling time	t_f	—	30	ns	

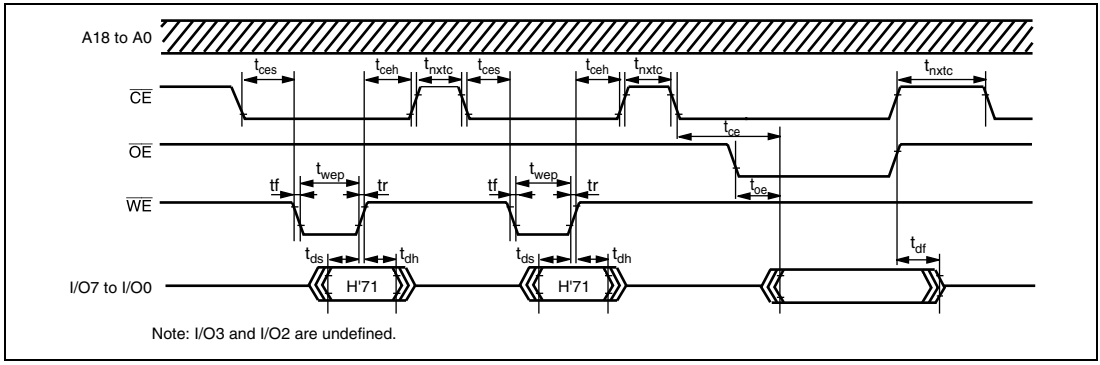


Figure 20.35 Waveform in Status-Read Mode

Table 20.26 Transition Time Rules before Command Wait State

Item	Symbol	MIN	MAX	Unit	Note
Standby clear (oscillation stabilized time)	t_{osc1}	30	—	ms	
Programming mode setup time	t_{bmv}	10	—	ms	
VCC hold time	t_{dwn}	0	—	ms	

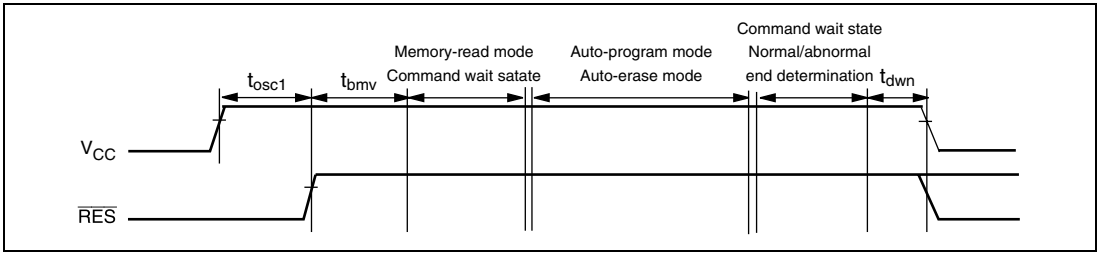


Figure 20.36 Oscillation Stabilized Time, Programming Mode Setup Time and Power Falling Sequence

Section 21 Clock Pulse Generator

This LSI has an on-chip clock pulse generator that generates the system clock (φ), the bus master clock, and internal clocks. The clock pulse generator consists of a system clock oscillator, PLL (Phase Locked Loop) circuit, clock selection circuit, medium-speed clock divider, bus master clock selection circuit, subclock oscillator, and waveform generation circuit.

Figure 21.1 shows a block diagram of the clock pulse generator.

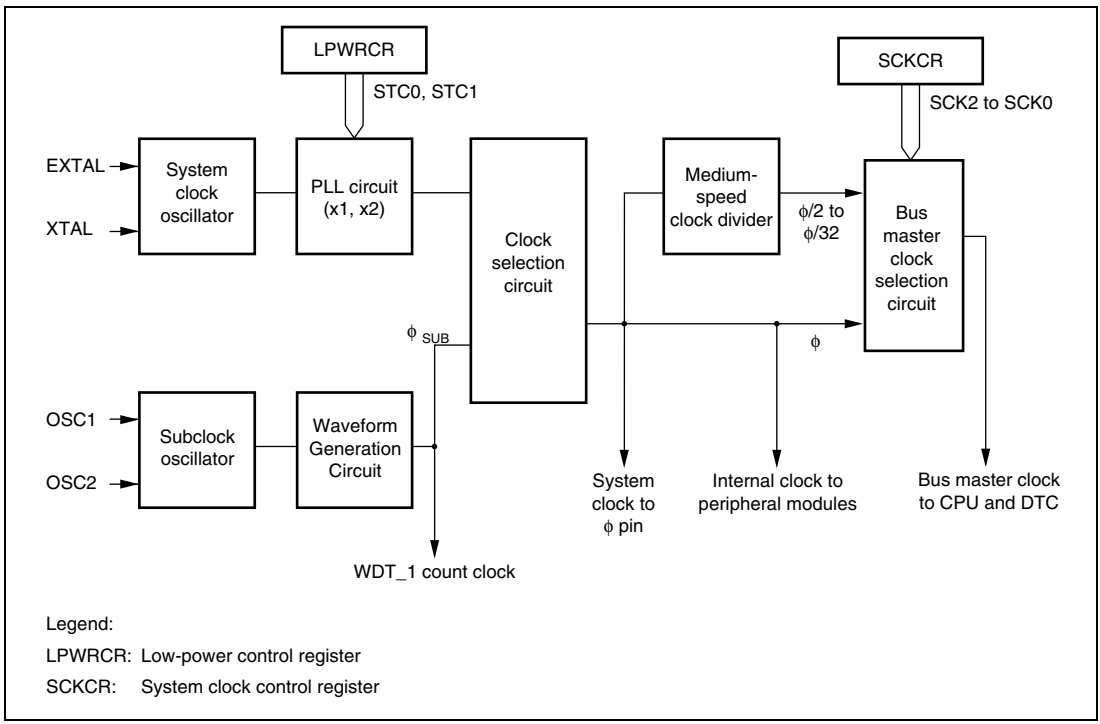


Figure 21.1 Block Diagram of Clock Pulse Generator

Frequency changes are performed by software by settings in the low-power control register (LPWRCR) and system clock control register (SCKCR).

21.1 Register Descriptions

The clock pulse generator has the following registers.

- System clock control register (SCKCR)
- Low-power control register (LPWRCR)

21.1.1 System Clock Control Register (SCKCR)

SCKCR performs the output of φ clock, operation selection when changing the multiplication ratio of the PLL circuit, and medium-speed mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	φ Clock Output Prohibited Controls φ output. • High-speed, medium-speed, and sleep modes 0: φ output 1: Fixed to high • Software standby and watch modes 0: Fixed to high 1: Fixed to high • Hardware standby mode 0: High impedance 1: High impedance
6	—	0	R/W	Reserved This is a readable/writable bit, but the write value should always be 0.
5, 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	STCS	0	R/W	Multiplication Ratio Switch Mode Selection Selects the operation when changing the multiplication ratio of PLL circuit. 0: The specified multiplication ratio is valid after software mode or watch mode is entered. 1: The specified multiplication ratio is valid immediately after rewriting the STC bit.

Bit	Bit Name	Initial Value	R/W	Description
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	Select the bus master clock.
0	SCK0	0	R/W	000: High-speed mode
				001: Medium-speed clock is ϕ/2
				010: Medium-speed clock is ϕ/4
				011: Medium-speed clock is ϕ/8
				100: Medium-speed clock is ϕ/16
				101: Medium-speed clock is ϕ/32
				11X: Setting prohibited

Legend
X: Don't care

21.1.2 Low-Power Control Register (LPWRCR)

LPWRCR controls power-down mode, selects sampling frequency for eliminating noise, controls a subclock oscillator, and specifies multiplication ratio.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	Direct Transfer On Flag
				0: When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*.
				1: Setting prohibited
6	LSON	0	R/W	Low-Speed On Flag
				0: When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*.
				1: Setting prohibited
5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select
				Selects the frequency of which the subclock (ϕ_{SUB}) generated by the subclock oscillator is sampled by the clock (ϕ) generated by the system clock oscillator.
				0: Sampling using ϕ/32
				1: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
4	SUBSTP	0	R/W	Subclock Oscillator Control Enables/disables a subclock oscillator. This bit should be set to 1 when a subclock is not used. 0: Enables subclock oscillator 1: Disables subclock oscillator
3	RFCUT	0	R/W	On-Chip Feedback Resistor Control Selects whether or not on-chip feedback resistor of the system clock generator is used when an external clock is input. Do not access when the crystal resonator is used. After setting this bit in the external clock input state, enter software standby mode or watch mode. When software standby mode or watch mode is entered, this bit switches whether or not on-chip feedback resistor is used. 0: On-chip feedback resistor of the system clock generator used 1: On-chip feedback resistor of the system clock generator not used
2	—	0	R/W	Reserved This is a readable/writable bit, but the write value should always be 0.
1	STC1	0	R/W	Multiplication Ratio Setting
0	STC0	0	R/W	Specify multiplication ratio of the PLL circuit. The specified multiplication ratio becomes valid after software standby mode or watch mode is entered. A setting of STC1 = 0 must be used in this LSI. 00: x 1 01: x 2 10: Setting prohibited 11: Setting prohibited

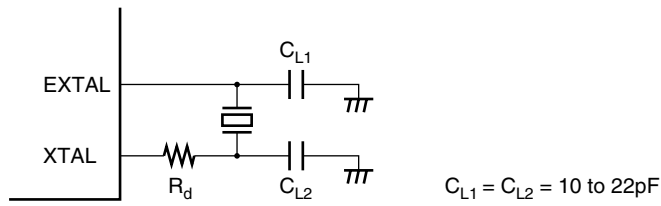
Note: * When watch mode is entered, high-speed mode must be set.

21.2 System Clock Oscillator

System clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

21.2.1 Connecting Crystal Resonator

A crystal resonator can be connected as shown in the example in figure 21.2. Select the damping resistance R_d according to table 21.1. An AT-cut parallel-resonance crystal should be used.



Note: C_{L1} and C_{L2} are reference values including the floating capacitance of the board.

Figure 21.2 Connection of Crystal Resonator (Example)

Table 21.1 Damping Resistance Value

Frequency (MHz)	8	10	12	16	20	25
R_d (Ω)	200	100	0	0	0	0

Figure 21.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 21.2.

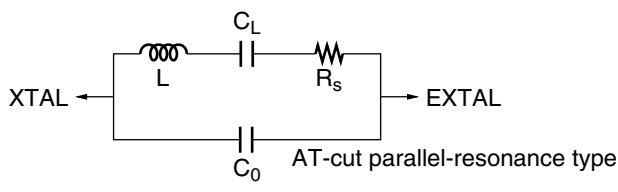


Figure 21.3 Crystal Resonator Equivalent Circuit

Table 21.2 Crystal Resonator Characteristics

Frequency (MHz)	8	10	12	16	20	25
R_S max (Ω)	80	60	60	50	40	40
C_0 max (pF)	7	7	7	7	7	7

21.2.2 External Clock Input

An external clock signal can be input as shown in the examples in figure 21.4. If the XTAL pin is left open, ensure that parasitic capacitance does not exceed 10 pF. When complementary clock is input to the XTAL pin, the external clock input should be fixed high in standby mode or watch mode.

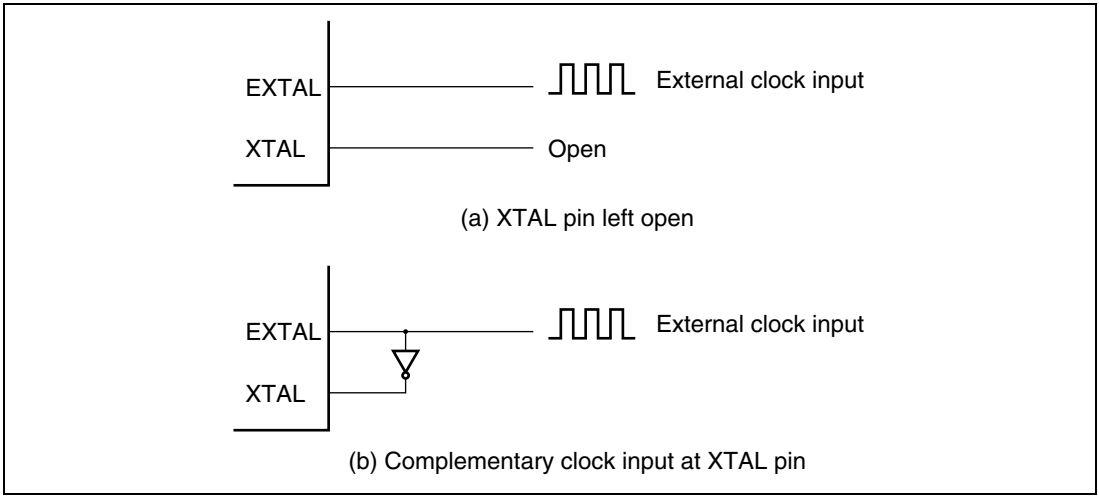


Figure 21.4 External Clock Input (Examples)

Table 21.3 shows the input conditions for the external clock.

Table 21.3 External Clock Input Conditions

		V_{cc}= 3.0 V to 5.5 V					
		H8S/2552 and H8S/2506 Series		H8S/2556 Series			
Item	Symbol	Min.	Max.	Min.	Max.	Unit	Test Conditions
External clock input low pulse width	t_{EXL}	TBD	—	TBD	—	ns	Figure 21.5
External clock input high pulse width	t_{EXH}	TBD	—	TBD	—	ns	
External clock rise time	t_{EXr}	—	TBD	—	TBD	ns	
External clock fall time	t_{EXf}	—	TBD	—	TBD	ns	
Clock pulse low width	t_{CL}	0.4		0.6		t_{cyc}	Figure 24.4
Clock pulse high width	t_{CH}	0.4		0.6		t_{cyc}	

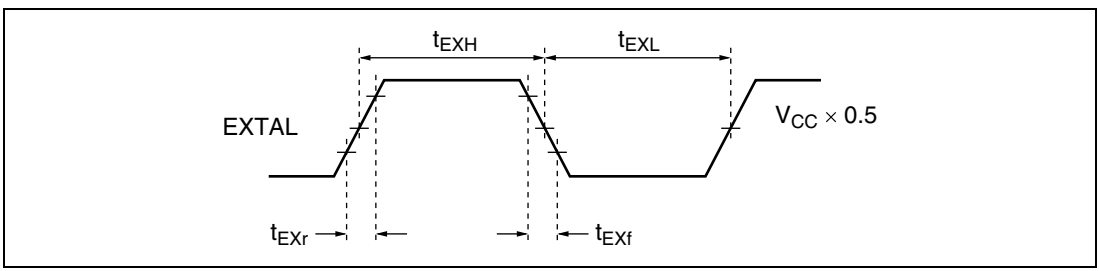


Figure 21.5 External Clock Input Timing

21.2.3 Notes on Switching External Clock

When two or more external clocks (e.g.: 10 MHz and 2 MHz) are used as the system clock, input clock should be switched in software standby mode.

An example of external clock switching circuit is shown in figure 21.6. An example of external clock switching timing is shown in figure 21.7.

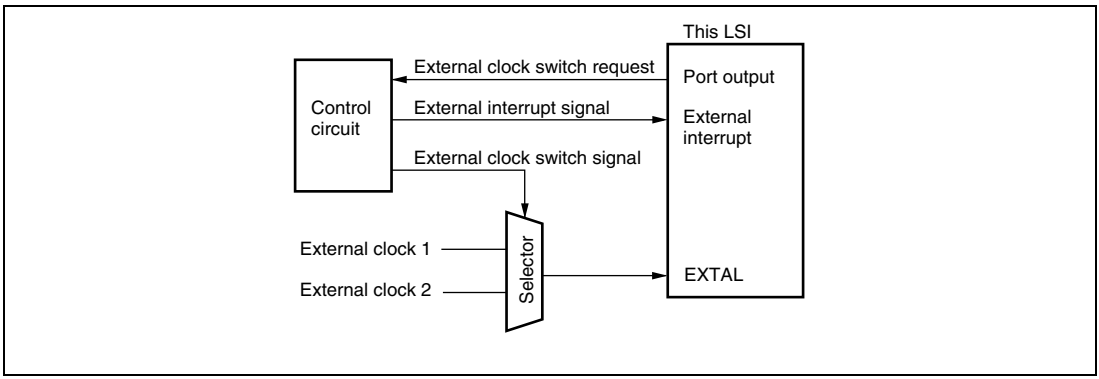


Figure 21.6 External Clock Switching Circuit (Examples)

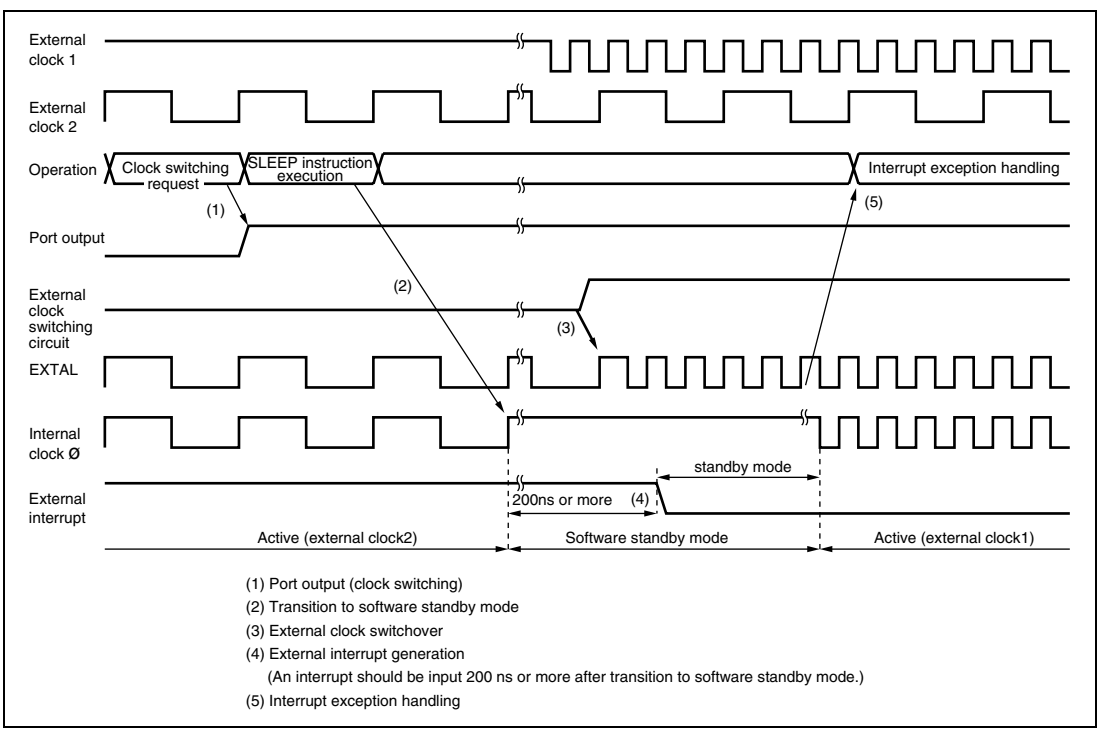


Figure 21.7 External Clock Switching Timing (Examples)

21.3 PLL Circuit

The PLL circuit multiplies the frequency from the clock pulse generator by one or two. The multiplication ratio is set by the STC1 and STC0 bits in LPWRCR. At the setting, the phase of the rising edge of an internal clock is controlled to match that of the rising edge of the EXTAL pin.

When changing the multiplication ratio of the PLL circuit, the operation differs according to the setting of the STCS bit in SCKCR.

When the STCS bit is 0, the changed multiplication ratio is valid after software standby mode or watch mode is entered. The transition time is set by the STS2 to STS0 bits in the standby control register (SBYCR). For details on SBYCR, refer to section 22.1.1, Standby Control Register (SBYCR).

1. In the initial state, the multiplication ratio of the PLL circuit is 1.
2. The transition time is set by the STS2 to STS0 bits.
3. The multiplication ratio is set by the STC1 and STC0 bits, and software standby mode or watch mode is entered.
4. The clock pulse generator stops, and the setting for the STC1 and STC0 bits becomes valid.
5. Software standby mode or watch mode is exited, and the transition time set by the STS2 to STS0 bits is ensured.
6. After the set transition time is elapsed, this LSI resumes operation with the changed multiplication ratio.

When the STCS bit is set to 1, after rewriting the STC1 and STC0 bits, this LSI operates with the changed multiplication ratio.

21.4 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, and $\phi/32$.

21.5 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the clock supplied to the bus master by setting the SCK2 to SCK0 bits in SCKCR. The bus master clock can be selected from system clock (ϕ), or medium-speed clocks ($\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, and $\phi/32$).

21.6 System Clock with IEBus

When using the IEBus*[1], the system clock should be set with one of 12 MHz, 12.58 MHz, 18 MHz, 18.87 MHz, 24 MHz, or 25.16 MHz.

When the IEBus*[1] is not used, any system clock frequency between 8 MHz and 26 MHz*[2] can be used.

Notes: 1. The IEBus is supported only by the H8S/2552 Series.
2. System clock frequency up to 20 MHz is supported by the H8S/2556 Series.

21.7 Subclock Oscillator

21.7.1 Connecting 32.768-kHz Crystal Resonator

To supply a clock to the subclock divider, connect a 32.768-kHz crystal resonator, as shown in figure 21.8. Figure 21.9 shows the equivalent circuit for a 32.768-kHz crystal resonator.

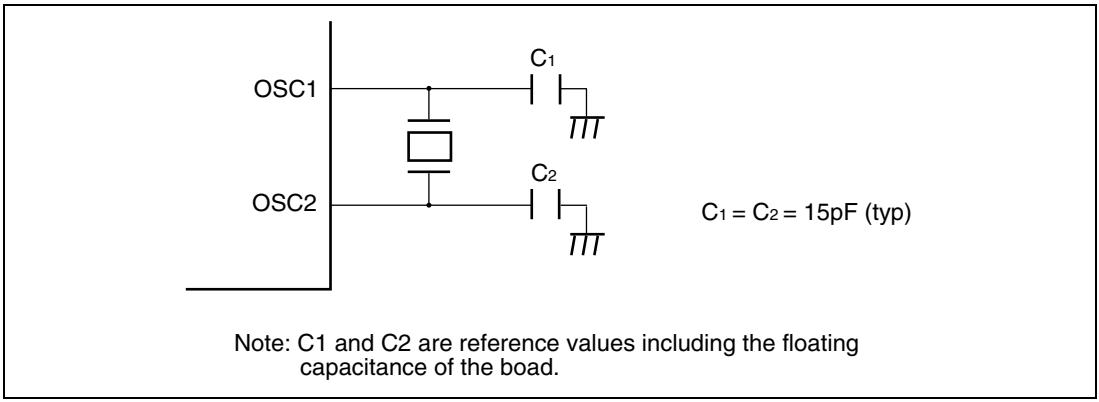


Figure 21.8 Connection Example of 32.768-kHz Crystal Resonator

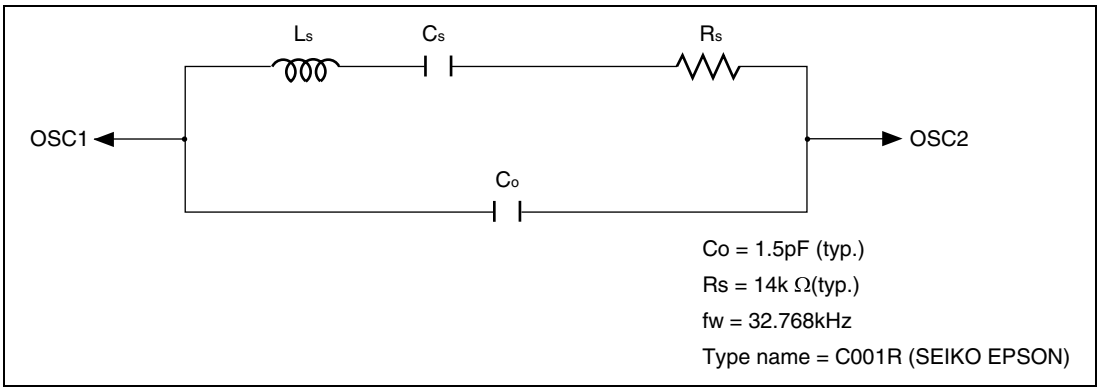


Figure 21.9 Equivalent Circuit for 32.768-kHz Crystal Resonator

21.7.2 Handling Pins when Subclock is not Used

If no subclock is required, connect the OSC1 pin to Vss and leave the OSC2 pin open, as shown in figure 21.10. The SUBSTP bit in LPWRCR must be set to 1.

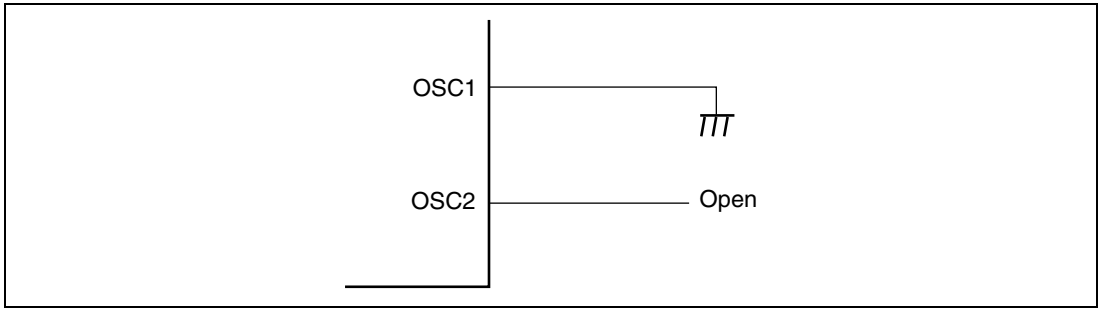


Figure 21.10 Pin Handling when Subclock is not Used

21.8 Subclock Waveform Generation Circuit

To eliminate noise from the subclock input from the OSC1 pin, the subclock is sampled using the dividing clock ϕ. The sampling frequency is set using the NESEL bit in LPWRCR. For details, see section 21.1.2, Low-Power Control Register (LPWRCR).

No sampling is performed in watch mode.

21.9 Usage Notes

21.9.1 Note on Crystal Resonator

As various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

21.9.2 Note on Board Design

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins. Make wires as short as possible. Other signal lines should be routed away from the oscillator circuit, as shown in figure 21.11. This is to prevent induction from interfering with correct oscillation.

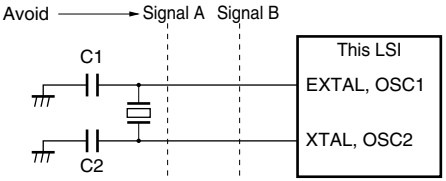
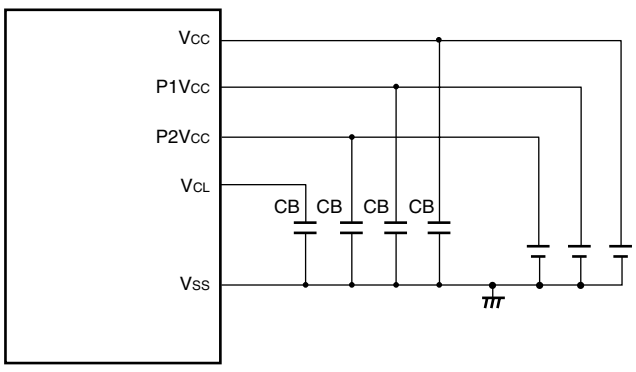


Figure 21.11 Note on Board Design of Oscillator Circuit

Figure 21.12 shows the recommended connection circuit between the power supply pins and Vss pin. The CB which is a capacitance for stabilization should be inserted near the pin between the power supply pins (V_{CC}, V_{CL}, $P1V_{CC}$, and $P2V_{CC}$) and Vss pin. Other signal lines should not be crossed.



CB = 0.1 µF (tentative recommended value)
Note: The CB is a laminated ceramic.

Figure 21.12 Recommended Connection Circuit between Power Supply Pins and Vss Pin

Section 22 Power-Down Modes

In addition to the normal program execution state, this LSI has power-down modes in which operation of the CPU and oscillator is halted and power consumption is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral modules, and so on.

This LSI's operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Sleep mode
(4) Watch mode
(5) Module stop mode
(6) Software standby mode
(7) Hardware standby mode

Of these, (2) to (7) are power-down modes. Sleep mode is a CPU state, medium-speed mode is CPU and bus master states, and module stop mode is an on-chip peripheral function (including bus masters other than the CPU) state.

After a reset, the LSI is in high-speed mode.

Table 22.1 shows the internal state of the LSI in the respective modes. Table 22.2 shows the conditions for shifting between power-down modes.

Figure 22.1 is a mode transition diagram.

Table 22.1 LSI Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted	Halted
Subclock pulse generator		Functioning /halted	Functioning /halted	Functioning /halted	Functioning /halted	Functioning	Functioning /halted	Halted
CPU	Instructions	Functioning	Medium-speed operation	Halted	Functioning	Halted	Halted	Halted
	Registers			Retained		Retained	Retained	Undefined

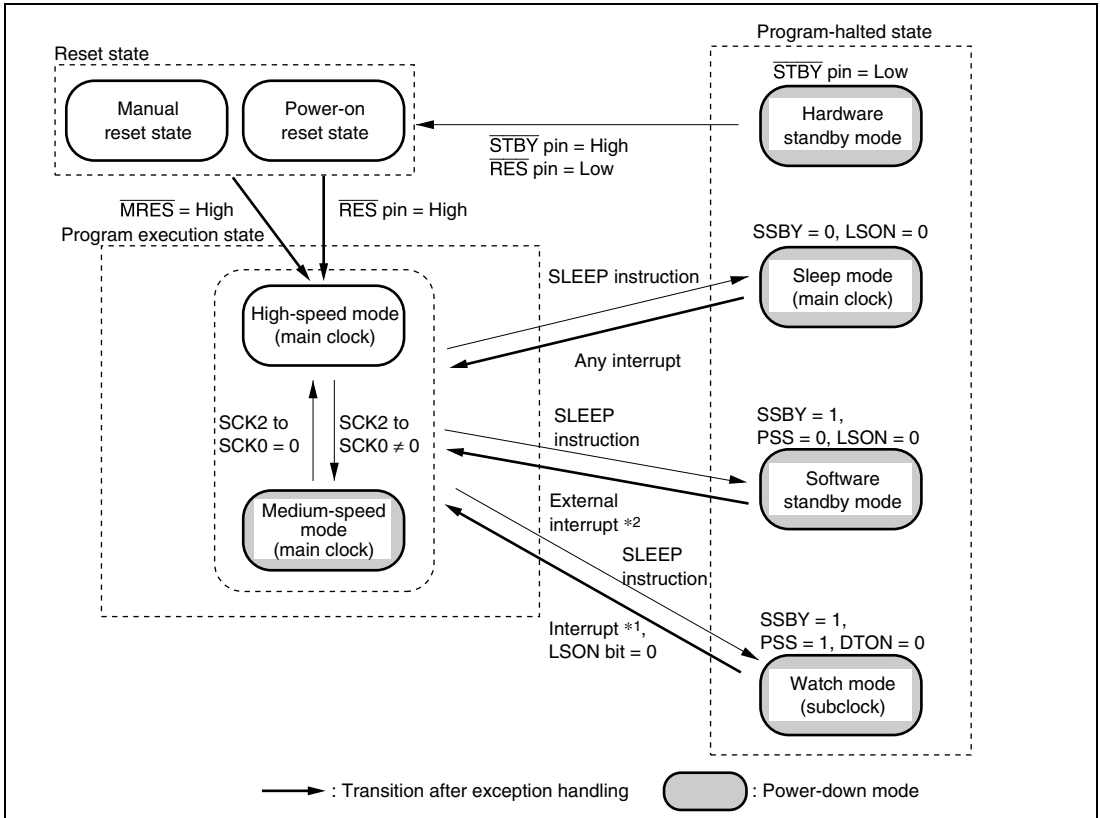
Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Software Standby	Hardware Standby
RAM		Functioning	Functioning	Functioning (DTC)	Functioning	Retained	Retained	Retained
I/O		Functioning	Functioning	Functioning	Functioning	Retained	Retained	High impedance
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQn							
Peripheral functions	PBC	Functioning	Medium-speed operation	Functioning	Functioning /halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	DTC	Functioning	Medium-speed operation	Functioning	Functioning /halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	WDT_1	Functioning	Functioning	Functioning	Functioning	Subclock operation*1	Halted (retained)	Halted (reset)
	WDT_0	Functioning	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	TMR	Functioning	Functioning	Functioning	Functioning /halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	TPU	Functioning	Functioning	Functioning	Functioning /halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	SCI							
	I^2C2							
	D/A							
	A/D	Functioning	Functioning	Functioning	Functioning /halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	IEB*2							
	HCAN*3							

Notes: "Halted (retained)" means that internal register values are retained. The internal state is "operation suspended."

"Halted (reset)" means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

1. When the WDT_1 is operated in watch mode, select the subclock as the clock to be used.
2. Supported only by the H8S/2552 Series.
3. Supported only by the H8S/2556 Series.



Notes: 1. NMI, IRQ0 to IRQ7, WDT1 interrupt
2. NMI, IRQ0 to IRQ7

- When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- From any state except hardware standby mode, a transition to the reset state occurs when $\overline{RES}$ is driven low. From any state except hardware standby mode and power-on reset state, a transition to the manual reset state occurs when $\overline{NRES}$ is driven low.
- From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.
- Always select high-speed mode before making a transition to watch mode.

Figure 22.1 Mode Transition Diagram

Table 22.2 Power-Down Mode Transition Conditions

Pre-Transition State	Status of Control Bit at Transition				State After Transition Invoked by SLEEP Instruction	State After Transition Back from Power-Down Mode Invoked by Interrupt
	SSBY	PSS	LSON	DTON		
High-speed/ Medium-speed	0	X	0	X	Sleep	High-speed/Medium-speed
	0	X	1	X	—	—
	1	0	0	X	Software standby	High-speed/Medium-speed
	1	0	1	X	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	X	—	—
	1	1	0	1	—	—

Legend

X: Don't care

—: Do not set.

22.1 Register Descriptions

The following registers relate to the power-down modes. For details on the system clock control register (SCKCR), refer to section 21.1.1, System Clock Control Register (SCKCR). For details on the low power control register (LPWRCR), refer to section 21.1.2, Low Power Control Register (LPWRCR). For details on the timer control/status register (TCSR), refer to section 11.3.5, Timer Control/Status Register (TCSR).

- Standby control register (SBYCR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)
- Low power control register (LPWRCR)
- System clock control register (SCKCR)
- Timer control/status register (TCSR)

22.1.1 Standby Control Register (SBYCR)

SBYCR performs power-down mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Specifies transition destination when the SLEEP instruction is executed. 0: Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. 1: Shifts to software standby mode or watch mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. Note that the value of the SSBY bit does not change even when software standby mode is canceled and making normal operation mode transition by executing an external interrupt. To clear this bit, 0 should be written to.
6	STS2	0	R/W	Standby Timer Select 2 to 0 These bits select the MCU wait time for clock settling to cancel software standby mode or watch mode by executing an external interrupt. With a crystal resonator (see table 22.3), select a wait time of 8 ms (oscillation settling time) or more, depending on the operating frequency. With an external clock, the standby time should be at least 8192 states. 000: Standby time = 8192 states 001: Standby time = 16384 states 010: Standby time = 32768 states 011: Standby time = 65536 states 100: Standby time = 131072 states 101: Standby time = 262144 states 110: Reserved 111: Reserved
5	STS1	0	R/W	
4	STS0	0	R/W	
3	OPE	1	R/W	Output Port Enable When in software standby mode or in watch mode, this bit selects whether to retain the output of the address bus and the bus control signals ($\overline{CS0}$ to $\overline{CS7}$*, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) or to set high impedance. 0: High impedance 1: Retains the output state

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

Note: * The $\overline{CS1}$ and $\overline{CS2}$ signals are not supported by the H8S/2556 Series.

22.1.2 Module Stop Control Registers A to C (MSTPCRA to MSTPCRC)

MSTPCR performs module stop mode control. When bits in MSTPCR are set to 1, module stop mode is set. When cleared to 0, module stop mode is cleared.

MSTPCRA

Bit	Bit Name	Initial Value	R/W	Target Module
7	MSTPA7*[1]	0	R/W	
6	MSTPA6	0	R/W	Data transfer controller (DTC)
5	MSTPA5	1	R/W	16-bit timer pulse unit (TPU)
4	MSTPA4	1	R/W	8-bit timer (TMR_0, TMR_1)
3	MSTPA3*[1]	1	R/W	
2	MSTPA2*[1]	1	R/W	
1	MSTPA1	1	R/W	A/D converter
0	MSTPA0	1	R/W	8-bit timer (TMR_2, TMR_3)

MSTPCRB

Bit	Bit Name	Initial Value	R/W	Target Module
7	MSTPB7	1	R/W	Serial communication interface 0 (SCI_0)
6	MSTPB6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTPB5	1	R/W	Serial communication interface 2 (SCI_2)
4	MSTPB4	1	R/W	I^2C bus interface 2_0 (I^2C2_0)
3	MSTPB3	1	R/W	I^2C bus interface 2_1 (I^2C2_1)
2	MSTPB2*[1]	1	R/W	
1	MSTPB1*[1]	1	R/W	
0	MSTPB0*[1]	1	R/W	

Bit	Bit Name	Initial Value	R/W	Target Module
7	MSTPC7	1	R/W	Serial communication interface 3 (SCI_3)
6	MSTPC6	1	R/W	Serial communication interface 4 (SCI_4)
5	MSTPC5	1	R/W	D/A converter
4	MSTPC4	1	R/W	PC break controller (PBC)
3	MSTPC3	1	R/W	IEBus™ controller (IEB)*2
2	MSTPC2	1	R/W	Hitachi controller area network (HCAN)*3
1	MSTPC1*1	1	R/W	
0	MSTPC0*1	1	R/W	

Notes: 1. The MSTPA7 bit can be read from or written to. This bit is initialized to 0. The write value should always be 0. The MSTPA3, MSTPA2, MSTPB2 to MSTPB0, MSTPC1, and MSTPC0 bits can be read from or written to. These bits are initialized to 1. The write value should always be 1.
2. Supported only by the H8S/2552 Series.
3 Supported only by the H8S/2556 Series.

22.2 Medium-Speed Mode

In high-speed mode, when the SCK2 to SCK0 bits in SCKCR are set to 1, the operating mode changes to medium-speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) specified by the SCK2 to SCK0 bits. The bus masters other than the CPU (DTC) also operate in medium-speed mode.

On-chip peripheral modules other than the bus masters always operate on the high-speed clock (ϕ).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, on-chip memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR and the LSON bit in LPWRCR are cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, LSON bit = 0, and PSS bit in TCSR_1 (WDT_1) = 0, operation shifts to the software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 22.2 shows the timing for transition to and clearance of medium-speed mode.

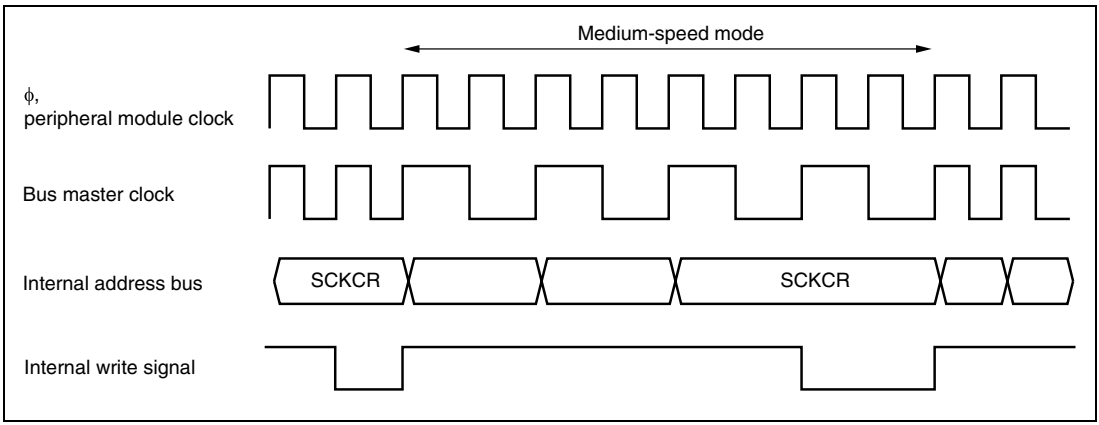


Figure 22.2 Medium-Speed Mode Transition and Clearance Timing

22.3 Sleep Mode

22.3.1 Transition to Sleep Mode

When the SLEEP instruction is executed while the SSBY bit in SBYCR = 0 and the LSON bit in LPWRCR = 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral modules do not stop.

22.3.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt, or signals at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an interrupt occurs, sleep mode is cleared and interrupt exception processing starts. Sleep mode is not cleared if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.
- Clearing with the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin

  Setting the $\overline{\text{RES}}$ pin or the $\overline{\text{MRES}}$ pin level low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin high starts the CPU performing reset exception processing.

- Clearing with the $\overline{\text{STBY}}$ Pin

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.4 Software Standby Mode

22.4.1 Transition to Software Standby Mode

A transition is made to software standby mode when the SLEEP instruction is executed while the SSBY bit in SBYCR = 1 and the LSON bit in LPWRCR = 0, and the PSS bit in TCSR_1 (WDT_1) = 0. In this mode, the CPU, on-chip peripheral modules, and oscillator all stop. However, the contents of the CPU's internal registers, on-chip RAM data, HCAN*[1], IEB*[2], and the states of on-chip peripheral modules other than the A/D converter, and the states of I/O ports are retained. In this mode the oscillator stops, and therefore power consumption is significantly reduced.

Notes: 1. Supported only by the H8S/2556 Series.
2. Supported only by the H8S/2552 Series.

22.4.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$), or by means of the $\overline{\text{RES}}$ pin, $\overline{\text{MRES}}$ pin, or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ7 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SBYCR, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.

  When clearing software standby mode with an IRQ0 to IRQ7 interrupt, set the corresponding enable bit/pin function switching bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ7 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.
- Clearing with the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin

  When the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin and $\overline{\text{MRES}}$ pin must be held low until clock oscillation settles. When the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin goes high, the CPU begins reset exception handling.
- Clearing with the $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.4.3 Oscillation Settling Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

- Using a crystal oscillator

  Set bits STS2 to STS0 so that the standby time is at least 8 ms (the oscillation settling time). Table 22.3 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.
- Using an external clock

  The standby time should be at least 8192 states.

Table 22.3 Oscillation Settling Time Settings

STS2	STS1	STS0	Standby Time	25 MHz*	20 MHz	16 MHz	13 MHz	10 MHz	8 MHz	Unit
0	0	0	8192 states	0.3	0.4	0.5	0.6	0.8	1.0	ms
		1	16384 states	0.7	0.8	1.0	1.3	1.6	2.0	
	1	0	32768 states	1.3	1.6	2.0	2.5	3.3	4.1	
		1	65536 states	2.6	3.3	4.1	5.0	6.6	8.2	
1	0	0	131072 states	5.2	6.6	8.2	10.1	13.1	16.4	
		1	262144 states	10.5	13.1	16.4	20.2	26.2	32.8	
	1	x	Reserved	—	—	—	—	—	—	—

Shading: Recommended time setting

Legend

X: Don't care

Note: * The H8S/2556 Series supports the operating frequencies up to 20 MHz.

22.4.4 Software Standby Mode Application Example

Figure 22.3 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

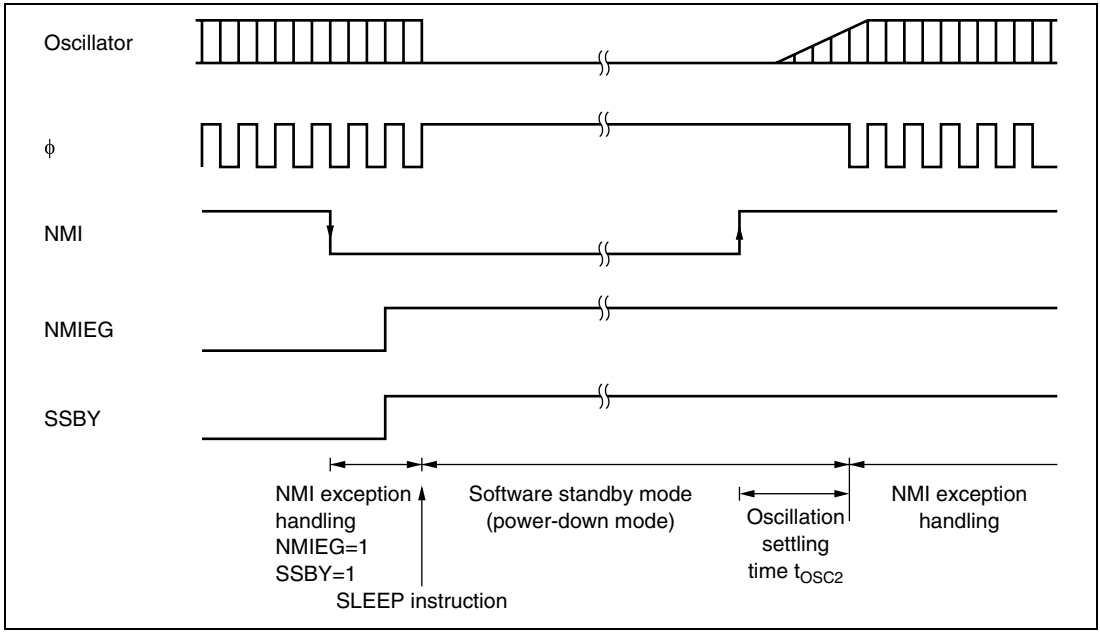


Figure 22.3 Software Standby Mode Application Example

22.5 Hardware Standby Mode

22.5.1 Transition to Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power consumption. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

Do not change the state of the mode pins (MD2 toMD0) during hardware standby mode.

22.5.2 Clearing Hardware Standby Mode

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator settles (at least t_{osc1} ms—the oscillation settling time). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

22.5.3 Hardware Standby Mode Timing

Figure 22.4 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

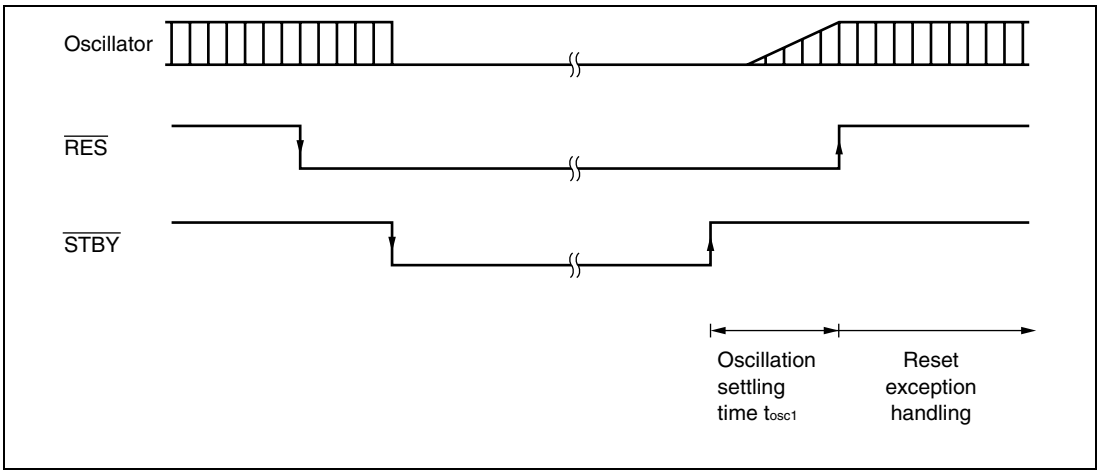


Figure 22.4 Hardware Standby Mode Timing

22.6 Module Stop Mode

Module stop mode can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the A/D converter are retained.

After reset clearance, all modules other than the DTC are in module stop mode.

When an on-chip peripheral module is in module stop mode, read/write access to its registers is disabled.

Since the operations of the bus controller and I/O port are stopped when sleep mode is entered at the all-module stop state (MSTPCR=H'FFFFFFFF), power consumption can further be reduced.

22.7 Watch Mode

22.7.1 Transition to Watch Mode

CPU operation makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode with SSBY bit in SBYCR = 1, DTON bit in LPWRCR = 0, and PSS bit in TCSR (WDT_1) = 1.

In watch mode, the CPU is stopped and peripheral modules other than the WDT_1 are also stopped. The contents of the CPU's internal registers, the data in on-chip RAM, and the statuses of the internal peripheral modules (excluding the HCAN*[1], IEB*[2], and A/D converter) and I/O ports are retained. To make a transition to watch mode, bits SCK2 to SCK0 in SCKCR must be set to 0.

Notes: 1. Supported only by the H8S/2556 Series
2. Supported only by the H8S/2552 Series

22.7.2 Clearing Watch Mode

Watch mode is cleared by any interrupt (WOVI1 interrupt, NMI pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$), or signals at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an interrupt occurs, watch mode is cleared and a transition is made to high-speed mode or medium-speed mode. When a transition is made to high-speed mode, a stable clock is supplied to all LSI circuits and interrupt exception processing starts after the time set in the STS2 to STS0 bits in SBYCR has elapsed. In the case of IRQ0 to IRQ7 interrupts, no transition is made from watch mode if the corresponding enable bit/pin function switching bit has been cleared to 0, and, in the case of interrupts from the internal peripheral modules, the interrupt enable register has been set to disable the reception of that interrupt, or is masked by the CPU.

  See section 22.4.3, Oscillation Settling Time after Clearing Software Standby Mode, for how to set the oscillation settling time when making a transition from watch mode to high-speed mode.
- Clearing with the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin

  For clearing watch mode by the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin, see section 22.4.2, Clearing Software Standby Mode.
- Clearing with the $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

22.8 φ Clock Output Disabled Function

The φ clock output can be controlled by the PSTOP bit in SCKCR and DDR of the corresponding port. When the PSTOP bit is set to 1, φ clock is stopped at the end of the bus cycle, and the φ output is driven high. When the PSTOP bit is cleared to 0, φ clock output is enabled. When DDR of the corresponding port is cleared to 0, φ clock output is disabled, and the port becomes an input port. Table 22.4 shows the φ pin state in each processing state.

Table 22.4 φ Pin State in Each Processing State

DDR	0		1
PSTOP	—	0	1
Hardware standby mode	High impedance		
Software standby mode, watch mode	High impedance	Fixed to high	
Sleep mode	High impedance	φ output	Fixed to high
High-speed mode, medium-speed mode	High impedance	φ output	Fixed to high

22.9 Usage Notes

22.9.1 I/O Port Status

In software standby mode and watch mode, I/O port states are retained. Therefore, there is no reduction in current consumption for the output current when a high-level signal is output.

22.9.2 Current Consumption during Oscillation Settling Wait Period

Current consumption increases during the oscillation settling wait period.

22.9.3 DTC Module Stop

Depending on the operating status of the DTC, the MSTPA6 bit may not be set to 1. Setting of the DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 8, Data Transfer Controller (DTC).

22.9.4 On-Chip Peripheral Module Interrupt

- Module stop mode

  Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source.

  Interrupts should therefore be disabled before entering module stop mode.

- Watch mode

  On-chip peripheral modules (DTC, TPU, and IIC2) that stop operation in watch mode cannot clear interrupt sources of the CPU after they make a transition to watch mode while an interrupt is being requested.

  Interrupts should therefore be disabled before executing the SLEEP instruction and entering watch mode.

22.9.5 Writing to MSTPCR

MSTPCR should only be written to by the CPU.

22.9.6 Entering Watch Mode and DTC Module Stop

To enter watch mode, set the DTC to module stop (write 1 to the MSTPA6 bit) and read the MSTPA6 bit as 1 before making a mode transition. After making a transition to active mode, clear module stop.

When the DTC activation source occurs in watch mode, the DTC is activated when module stop is cleared after active mode is entered.

Section 23 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register Addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The access size is indicated.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in ascending order of addresses).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functional module, in ascending order of addresses).
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip module, refer to the section on that on-chip module.

23.1 Register Addresses (by function module, in address order)

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
DTC mode register A	MRA	8	H'EBC0 to H'EFBF	DTC	16/32*[2]	1
DTC source address register	SAR	24		DTC	16/32*[2]	1
DTC mode register B	MRB	8		DTC	16/32*[2]	1
DTC destination address register	DAR	24		DTC	16/32*[2]	1
DTC transfer count register A	CRA	16		DTC	16/32*[2]	1
DTC transfer count register B	CRB	16		DTC	16/32*[2]	1
IEBus control register	IECTR	8	H'F800	IEB	8	5
IEBus command register	IECMR	8	H'F801	IEB	8	5
IEBus master control register	IEMCR	8	H'F802	IEB	8	5
IEBus master unit address register 1	IEAR1	8	H'F803	IEB	8	5
IEBus master unit address register 2	IEAR2	8	H'F804	IEB	8	5
IEBus slave address setting register 1	IESA1	8	H'F805	IEB	8	5
IEBus slave address setting register 2	IESA2	8	H'F806	IEB	8	5
IEBus transmit frame length register	IETBFL	8	H'F807	IEB	8	5
IEBus transmit buffer register	IETBR	8	H'F808	IEB	8	5
IEBus reception master address register 1	IEMA1	8	H'F809	IEB	8	5
IEBus reception master address register 2	IEMA2	8	H'F80A	IEB	8	5
IEBus receive control field register	IERCTL	8	H'F80B	IEB	8	5
IEBus receive frame length register	IERBFL	8	H'F80C	IEB	8	5
IEBus receive buffer register	IERBR	8	H'F80D	IEB	8	5
IEBus lock address register 1	IELA1	8	H'F80E	IEB	8	5
IEBus lock address register 2	IELA2	8	H'F80F	IEB	8	5
IEBus general flag register	IEFLG	8	H'F810	IEB	8	5
IEBus transmit status register	IETSR	8	H'F811	IEB	8	5
IEBus transmit interrupt enable register	IEIET	8	H'F812	IEB	8	5
IEBus transmit error flag register	IETEF	8	H'F813	IEB	8	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
IEBus receive status register	IERSR	8	H'F814	IEB	8	5
IEBus receive interrupt enable register	IEIER	8	H'F815	IEB	8	5
IEBus receive error flag register	IEREF	8	H'F816	IEB	8	5
Port H data direction register	PHDDR	8	H'FA10	PORT	8	4
Port J data direction register	PJDDR	8	H'FA11	PORT	8	4
Port H data register	PHDR	8	H'FA12	PORT	8	4
Port J data register	PJDR	8	H'FA13	PORT	8	4
Port H register	PORTH	8	H'FA14	PORT	8	4
Port J register	PORTJ	8	H'FA15	PORT	8	4
IIC bus control register 1_0	ICCR1_0	8	H'FA20	IIC2_0	8	4
IIC bus control register 2_0	ICCR2_0	8	H'FA21	IIC2_0	8	4
IIC bus mode register_0	ICMR_0	8	H'FA22	IIC2_0	8	4
IIC bus interrupt enable register_0	ICIER_0	8	H'FA23	IIC2_0	8	4
IIC bus status register_0	ICSR_0	8	H'FA24	IIC2_0	8	4
Slave address register_0	SAR_0	8	H'FA25	IIC2_0	8	4
IIC bus transmit data register_0	ICDRT_0	8	H'FA26	IIC2_0	8	4
IIC bus receive data register_0	ICDRR_0	8	H'FA27	IIC2_0	8	4
IIC bus control register 1_1	ICCR1_1	8	H'FA28	IIC2_1	8	4
IIC bus control register 2_1	ICCR2_1	8	H'FA29	IIC2_1	8	4
IIC bus mode register_1	ICMR_1	8	H'FA2A	IIC2_1	8	4
IIC bus interrupt enable register_1	ICIER_1	8	H'FA2B	IIC2_1	8	4
IIC bus status register_1	ICSR_1	8	H'FA2C	IIC2_1	8	4
Slave address register_1	SAR_1	8	H'FA2D	IIC2_1	8	4
IIC bus transmit data register_1	ICDRT_1	8	H'FA2E	IIC2_1	8	4
IIC bus receive data register_1	ICDRR_1	8	H'FA2F	IIC2_1	8	4
Master control register	MCR	8	H'FB00	HCAN	16	5
General status register	GSR	8	H'FB01	HCAN	16	5
Bit configuration register	BCR	16	H'FB02	HCAN	16	5
Mailbox configuration register	MBCR	16	H'FB04	HCAN	16	5
Transmit wait register	TXPR	16	H'FB06	HCAN	16	5
Transmit wait cancel register	TXCR	16	H'FB08	HCAN	16	5
Transmit acknowledge register	TXACK	16	H'FB0A	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Abort acknowledge register	ABACK	16	H'FB0C	HCAN	16	5
Receive complete register	RXPR	16	H'FB0E	HCAN	16	5
Remote request register	RFPR	16	H'FB10	HCAN	16	5
Interrupt register	IRR	16	H'FB12	HCAN	16	5
Mailbox interrupt mask register	MBIMR	16	H'FB14	HCAN	16	5
Interrupt mask register	IMR	16	H'FB16	HCAN	16	5
Receive error counter	REC	8	H'FB18	HCAN	16	5
Transmit error counter	TEC	8	H'FB19	HCAN	16	5
Unread message status register	UMSR	16	H'FB1A	HCAN	16	5
Local acceptance filter mask L	LAFML	16	H'FB1C	HCAN	16	5
Local acceptance filter mask H	LAFMH	16	H'FB1E	HCAN	16	5
Message control 0[1]	MC0[1]	8	H'FB20	HCAN	16	5
Message control 0[2]	MC0[2]	8	H'FB21	HCAN	16	5
Message control 0[3]	MC0[3]	8	H'FB22	HCAN	16	5
Message control 0[4]	MC0[4]	8	H'FB23	HCAN	16	5
Message control 0[5]	MC0[5]	8	H'FB24	HCAN	16	5
Message control 0[6]	MC0[6]	8	H'FB25	HCAN	16	5
Message control 0[7]	MC0[7]	8	H'FB26	HCAN	16	5
Message control 0[8]	MC0[8]	8	H'FB27	HCAN	16	5
Message control 1[1]	MC1[1]	8	H'FB28	HCAN	16	5
Message control 1[2]	MC1[2]	8	H'FB29	HCAN	16	5
Message control 1[3]	MC1[3]	8	H'FB2A	HCAN	16	5
Message control 1[4]	MC1[4]	8	H'FB2B	HCAN	16	5
Message control 1[5]	MC1[5]	8	H'FB2C	HCAN	16	5
Message control 1[6]	MC1[6]	8	H'FB2D	HCAN	16	5
Message control 1[7]	MC1[7]	8	H'FB2E	HCAN	16	5
Message control 1[8]	MC1[8]	8	H'FB2F	HCAN	16	5
Message control 2[1]	MC2[1]	8	H'FB30	HCAN	16	5
Message control 2[2]	MC2[2]	8	H'FB31	HCAN	16	5
Message control 2[3]	MC2[3]	8	H'FB32	HCAN	16	5
Message control 2[4]	MC2[4]	8	H'FB33	HCAN	16	5
Message control 2[5]	MC2[5]	8	H'FB34	HCAN	16	5
Message control 2[6]	MC2[6]	8	H'FB35	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message control 2[7]	MC2[7]	8	H'FB36	HCAN	16	5
Message control 2[8]	MC2[8]	8	H'FB37	HCAN	16	5
Message control 3[1]	MC3[1]	8	H'FB38	HCAN	16	5
Message control 3[2]	MC3[2]	8	H'FB39	HCAN	16	5
Message control 3[3]	MC3[3]	8	H'FB3A	HCAN	16	5
Message control 3[4]	MC3[4]	8	H'FB3B	HCAN	16	5
Message control 3[5]	MC3[5]	8	H'FB3C	HCAN	16	5
Message control 3[6]	MC3[6]	8	H'FB3D	HCAN	16	5
Message control 3[7]	MC3[7]	8	H'FB3E	HCAN	16	5
Message control 3[8]	MC3[8]	8	H'FB3F	HCAN	16	5
Message control 4[1]	MC4[1]	8	H'FB40	HCAN	16	5
Message control 4[2]	MC4[2]	8	H'FB41	HCAN	16	5
Message control 4[3]	MC4[3]	8	H'FB42	HCAN	16	5
Message control 4[4]	MC4[4]	8	H'FB43	HCAN	16	5
Message control 4[5]	MC4[5]	8	H'FB44	HCAN	16	5
Message control 4[6]	MC4[6]	8	H'FB45	HCAN	16	5
Message control 4[7]	MC4[7]	8	H'FB46	HCAN	16	5
Message control 4[8]	MC4[8]	8	H'FB47	HCAN	16	5
Message control 5[1]	MC5[1]	8	H'FB48	HCAN	16	5
Message control 5[2]	MC5[2]	8	H'FB49	HCAN	16	5
Message control 5[3]	MC5[3]	8	H'FB4A	HCAN	16	5
Message control 5[4]	MC5[4]	8	H'FB4B	HCAN	16	5
Message control 5[5]	MC5[5]	8	H'FB4C	HCAN	16	5
Message control 5[6]	MC5[6]	8	H'FB4D	HCAN	16	5
Message control 5[7]	MC5[7]	8	H'FB4E	HCAN	16	5
Message control 5[8]	MC5[8]	8	H'FB4F	HCAN	16	5
Message control 6[1]	MC6[1]	8	H'FB50	HCAN	16	5
Message control 6[2]	MC6[2]	8	H'FB51	HCAN	16	5
Message control 6[3]	MC6[3]	8	H'FB52	HCAN	16	5
Message control 6[4]	MC6[4]	8	H'FB53	HCAN	16	5
Message control 6[5]	MC6[5]	8	H'FB54	HCAN	16	5
Message control 6[6]	MC6[6]	8	H'FB55	HCAN	16	5
Message control 6[7]	MC6[7]	8	H'FB56	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message control 6[8]	MC6[8]	8	H'FB57	HCAN	16	5
Message control 7[1]	MC7[1]	8	H'FB58	HCAN	16	5
Message control 7[2]	MC7[2]	8	H'FB59	HCAN	16	5
Message control 7[3]	MC7[3]	8	H'FB5A	HCAN	16	5
Message control 7[4]	MC7[4]	8	H'FB5B	HCAN	16	5
Message control 7[5]	MC7[5]	8	H'FB5C	HCAN	16	5
Message control 7[6]	MC7[6]	8	H'FB5D	HCAN	16	5
Message control 7[7]	MC7[7]	8	H'FB5E	HCAN	16	5
Message control 7[8]	MC7[8]	8	H'FB5F	HCAN	16	5
Message control 8[1]	MC8[1]	8	H'FB60	HCAN	16	5
Message control 8[2]	MC8[2]	8	H'FB61	HCAN	16	5
Message control 8[3]	MC8[3]	8	H'FB62	HCAN	16	5
Message control 8[4]	MC8[4]	8	H'FB63	HCAN	16	5
Message control 8[5]	MC8[5]	8	H'FB64	HCAN	16	5
Message control 8[6]	MC8[6]	8	H'FB65	HCAN	16	5
Message control 8[7]	MC8[7]	8	H'FB66	HCAN	16	5
Message control 8[8]	MC8[8]	8	H'FB67	HCAN	16	5
Message control 9[1]	MC9[1]	8	H'FB68	HCAN	16	5
Message control 9[2]	MC9[2]	8	H'FB69	HCAN	16	5
Message control 9[3]	MC9[3]	8	H'FB6A	HCAN	16	5
Message control 9[4]	MC9[4]	8	H'FB6B	HCAN	16	5
Message control 9[5]	MC9[5]	8	H'FB6C	HCAN	16	5
Message control 9[6]	MC9[6]	8	H'FB6D	HCAN	16	5
Message control 9[7]	MC9[7]	8	H'FB6E	HCAN	16	5
Message control 9[8]	MC9[8]	8	H'FB6F	HCAN	16	5
Message control 10[1]	MC10[1]	8	H'FB70	HCAN	16	5
Message control 10[2]	MC10[2]	8	H'FB71	HCAN	16	5
Message control 10[3]	MC10[3]	8	H'FB72	HCAN	16	5
Message control 10[4]	MC10[4]	8	H'FB73	HCAN	16	5
Message control 10[5]	MC10[5]	8	H'FB74	HCAN	16	5
Message control 10[6]	MC10[6]	8	H'FB75	HCAN	16	5
Message control 10[7]	MC10[7]	8	H'FB76	HCAN	16	5
Message control 10[8]	MC10[8]	8	H'FB77	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message control 11[1]	MC11[1]	8	H'FB78	HCAN	16	5
Message control 11[2]	MC11[2]	8	H'FB79	HCAN	16	5
Message control 11[3]	MC11[3]	8	H'FB7A	HCAN	16	5
Message control 11[4]	MC11[4]	8	H'FB7B	HCAN	16	5
Message control 11[5]	MC11[5]	8	H'FB7C	HCAN	16	5
Message control 11[6]	MC11[6]	8	H'FB7D	HCAN	16	5
Message control 11[7]	MC11[7]	8	H'FB7E	HCAN	16	5
Message control 11[8]	MC11[8]	8	H'FB7F	HCAN	16	5
Message control 12[1]	MC12[1]	8	H'FB80	HCAN	16	5
Message control 12[2]	MC12[2]	8	H'FB81	HCAN	16	5
Message control 12[3]	MC12[3]	8	H'FB82	HCAN	16	5
Message control 12[4]	MC12[4]	8	H'FB83	HCAN	16	5
Message control 12[5]	MC12[5]	8	H'FB84	HCAN	16	5
Message control 12[6]	MC12[6]	8	H'FB85	HCAN	16	5
Message control 12[7]	MC12[7]	8	H'FB86	HCAN	16	5
Message control 12[8]	MC12[8]	8	H'FB87	HCAN	16	5
Message control 13[1]	MC13[1]	8	H'FB88	HCAN	16	5
Message control 13[2]	MC13[2]	8	H'FB89	HCAN	16	5
Message control 13[3]	MC13[3]	8	H'FB8A	HCAN	16	5
Message control 13[4]	MC13[4]	8	H'FB8B	HCAN	16	5
Message control 13[5]	MC13[5]	8	H'FB8C	HCAN	16	5
Message control 13[6]	MC13[6]	8	H'FB8D	HCAN	16	5
Message control 13[7]	MC13[7]	8	H'FB8E	HCAN	16	5
Message control 13[8]	MC13[8]	8	H'FB8F	HCAN	16	5
Message control 14[1]	MC14[1]	8	H'FB90	HCAN	16	5
Message control 14[2]	MC14[2]	8	H'FB91	HCAN	16	5
Message control 14[3]	MC14[3]	8	H'FB92	HCAN	16	5
Message control 14[4]	MC14[4]	8	H'FB93	HCAN	16	5
Message control 14[5]	MC14[5]	8	H'FB94	HCAN	16	5
Message control 14[6]	MC14[6]	8	H'FB95	HCAN	16	5
Message control 14[7]	MC14[7]	8	H'FB96	HCAN	16	5
Message control 14[8]	MC14[8]	8	H'FB97	HCAN	16	5
Message control 15[1]	MC15[1]	8	H'FB98	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message control 15[2]	MC15[2]	8	H'FB99	HCAN	16	5
Message control 15[3]	MC15[3]	8	H'FB9A	HCAN	16	5
Message control 15[4]	MC15[4]	8	H'FB9B	HCAN	16	5
Message control 15[5]	MC15[5]	8	H'FB9C	HCAN	16	5
Message control 15[6]	MC15[6]	8	H'FB9D	HCAN	16	5
Message control 15[7]	MC15[7]	8	H'FB9E	HCAN	16	5
Message control 15[8]	MC15[8]	8	H'FB9F	HCAN	16	5
Message data 0[1]	MD0[1]	8	H'FBB0	HCAN	16	5
Message data 0[2]	MD0[2]	8	H'FBB1	HCAN	16	5
Message data 0[3]	MD0[3]	8	H'FBB2	HCAN	16	5
Message data 0[4]	MD0[4]	8	H'FBB3	HCAN	16	5
Message data 0[5]	MD0[5]	8	H'FBB4	HCAN	16	5
Message data 0[6]	MD0[6]	8	H'FBB5	HCAN	16	5
Message data 0[7]	MD0[7]	8	H'FBB6	HCAN	16	5
Message data 0[8]	MD0[8]	8	H'FBB7	HCAN	16	5
Message data 1[1]	MD1[1]	8	H'FBB8	HCAN	16	5
Message data 1[2]	MD1[2]	8	H'FBB9	HCAN	16	5
Message data 1[3]	MD1[3]	8	H'FBBA	HCAN	16	5
Message data 1[4]	MD1[4]	8	H'FBBB	HCAN	16	5
Message data 1[5]	MD1[5]	8	H'FBBC	HCAN	16	5
Message data 1[6]	MD1[6]	8	H'FBBD	HCAN	16	5
Message data 1[7]	MD1[7]	8	H'FBBE	HCAN	16	5
Message data 1[8]	MD1[8]	8	H'FBBF	HCAN	16	5
Message data 2[1]	MD2[1]	8	H'FBC0	HCAN	16	5
Message data 2[2]	MD2[2]	8	H'FBC1	HCAN	16	5
Message data 2[3]	MD2[3]	8	H'FBC2	HCAN	16	5
Message data 2[4]	MD2[4]	8	H'FBC3	HCAN	16	5
Message data 2[5]	MD2[5]	8	H'FBC4	HCAN	16	5
Message data 2[6]	MD2[6]	8	H'FBC5	HCAN	16	5
Message data 2[7]	MD2[7]	8	H'FBC6	HCAN	16	5
Message data 2[8]	MD2[8]	8	H'FBC7	HCAN	16	5
Message data 3[1]	MD3[1]	8	H'FBC8	HCAN	16	5
Message data 3[2]	MD3[2]	8	H'FBC9	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message data 3[3]	MD3[3]	8	H'FBCA	HCAN	16	5
Message data 3[4]	MD3[4]	8	H'FBCB	HCAN	16	5
Message data 3[5]	MD3[5]	8	H'FBCC	HCAN	16	5
Message data 3[6]	MD3[6]	8	H'FBCD	HCAN	16	5
Message data 3[7]	MD3[7]	8	H'FBCE	HCAN	16	5
Message data 3[8]	MD3[8]	8	H'FBCF	HCAN	16	5
Message data 4[1]	MD4[1]	8	H'FBD0	HCAN	16	5
Message data 4[2]	MD4[2]	8	H'FBD1	HCAN	16	5
Message data 4[3]	MD4[3]	8	H'FBD2	HCAN	16	5
Message data 4[4]	MD4[4]	8	H'FBD3	HCAN	16	5
Message data 4[5]	MD4[5]	8	H'FBD4	HCAN	16	5
Message data 4[6]	MD4[6]	8	H'FBD5	HCAN	16	5
Message data 4[7]	MD4[7]	8	H'FBD6	HCAN	16	5
Message data 4[8]	MD4[8]	8	H'FBD7	HCAN	16	5
Message data 5[1]	MD5[1]	8	H'FBD8	HCAN	16	5
Message data 5[2]	MD5[2]	8	H'FBD9	HCAN	16	5
Message data 5[3]	MD5[3]	8	H'FBDA	HCAN	16	5
Message data 5[4]	MD5[4]	8	H'FBDB	HCAN	16	5
Message data 5[5]	MD5[5]	8	H'FBDC	HCAN	16	5
Message data 5[6]	MD5[6]	8	H'FBDD	HCAN	16	5
Message data 5[7]	MD5[7]	8	H'FBDE	HCAN	16	5
Message data 5[8]	MD5[8]	8	H'FBDF	HCAN	16	5
Message data 6[1]	MD6[1]	8	H'FBE0	HCAN	16	5
Message data 6[2]	MD6[2]	8	H'FBE1	HCAN	16	5
Message data 6[3]	MD6[3]	8	H'FBE2	HCAN	16	5
Message data 6[4]	MD6[4]	8	H'FBE3	HCAN	16	5
Message data 6[5]	MD6[5]	8	H'FBE4	HCAN	16	5
Message data 6[6]	MD6[6]	8	H'FBE5	HCAN	16	5
Message data 6[7]	MD6[7]	8	H'FBE6	HCAN	16	5
Message data 6[8]	MD6[8]	8	H'FBE7	HCAN	16	5
Message data 7[1]	MD7[1]	8	H'FBE8	HCAN	16	5
Message data 7[2]	MD7[2]	8	H'FBE9	HCAN	16	5
Message data 7[3]	MD7[3]	8	H'FBEA	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message data 7[4]	MD7[4]	8	H'FBEB	HCAN	16	5
Message data 7[5]	MD7[5]	8	H'FBEC	HCAN	16	5
Message data 7[6]	MD7[6]	8	H'FBED	HCAN	16	5
Message data 7[7]	MD7[7]	8	H'FBEE	HCAN	16	5
Message data 7[8]	MD7[8]	8	H'FBEF	HCAN	16	5
Message data 8[1]	MD8[1]	8	H'FBF0	HCAN	16	5
Message data 8[2]	MD8[2]	8	H'FBF1	HCAN	16	5
Message data 8[3]	MD8[3]	8	H'FBF2	HCAN	16	5
Message data 8[4]	MD8[4]	8	H'FBF3	HCAN	16	5
Message data 8[5]	MD8[5]	8	H'FBF4	HCAN	16	5
Message data 8[6]	MD8[6]	8	H'FBF5	HCAN	16	5
Message data 8[7]	MD8[7]	8	H'FBF6	HCAN	16	5
Message data 8[8]	MD8[8]	8	H'FBF7	HCAN	16	5
Message data 9[1]	MD9[1]	8	H'FBF8	HCAN	16	5
Message data 9[2]	MD9[2]	8	H'FBF9	HCAN	16	5
Message data 9[3]	MD9[3]	8	H'FBFA	HCAN	16	5
Message data 9[4]	MD9[4]	8	H'FBFB	HCAN	16	5
Message data 9[5]	MD9[5]	8	H'FBFC	HCAN	16	5
Message data 9[6]	MD9[6]	8	H'FBFD	HCAN	16	5
Message data 9[7]	MD9[7]	8	H'FBFE	HCAN	16	5
Message data 9[8]	MD9[8]	8	H'FBFF	HCAN	16	5
Message data 10[1]	MD10[1]	8	H'FC00	HCAN	16	5
Message data 10[2]	MD10[2]	8	H'FC01	HCAN	16	5
Message data 10[3]	MD10[3]	8	H'FC02	HCAN	16	5
Message data 10[4]	MD10[4]	8	H'FC03	HCAN	16	5
Message data 10[5]	MD10[5]	8	H'FC04	HCAN	16	5
Message data 10[6]	MD10[6]	8	H'FC05	HCAN	16	5
Message data 10[7]	MD10[7]	8	H'FC06	HCAN	16	5
Message data 10[8]	MD10[8]	8	H'FC07	HCAN	16	5
Message data 11[1]	MD11[1]	8	H'FC08	HCAN	16	5
Message data 11[2]	MD11[2]	8	H'FC09	HCAN	16	5
Message data 11[3]	MD11[3]	8	H'FC0A	HCAN	16	5
Message data 11[4]	MD11[4]	8	H'FC0B	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message data 11[5]	MD11[5]	8	H'FC0C	HCAN	16	5
Message data 11[6]	MD11[6]	8	H'FC0D	HCAN	16	5
Message data 11[7]	MD11[7]	8	H'FC0E	HCAN	16	5
Message data 11[8]	MD11[8]	8	H'FC0F	HCAN	16	5
Message data 12[1]	MD12[1]	8	H'FC10	HCAN	16	5
Message data 12[2]	MD12[2]	8	H'FC11	HCAN	16	5
Message data 12[3]	MD12[3]	8	H'FC12	HCAN	16	5
Message data 12[4]	MD12[4]	8	H'FC13	HCAN	16	5
Message data 12[5]	MD12[5]	8	H'FC14	HCAN	16	5
Message data 12[6]	MD12[6]	8	H'FC15	HCAN	16	5
Message data 12[7]	MD12[7]	8	H'FC16	HCAN	16	5
Message data 12[8]	MD12[8]	8	H'FC17	HCAN	16	5
Message data 13[1]	MD13[1]	8	H'FC18	HCAN	16	5
Message data 13[2]	MD13[2]	8	H'FC19	HCAN	16	5
Message data 13[3]	MD13[3]	8	H'FC1A	HCAN	16	5
Message data 13[4]	MD13[4]	8	H'FC1B	HCAN	16	5
Message data 13[5]	MD13[5]	8	H'FC1C	HCAN	16	5
Message data 13[6]	MD13[6]	8	H'FC1D	HCAN	16	5
Message data 13[7]	MD13[7]	8	H'FC1E	HCAN	16	5
Message data 13[8]	MD13[8]	8	H'FC1F	HCAN	16	5
Message data 14[1]	MD14[1]	8	H'FC20	HCAN	16	5
Message data 14[2]	MD14[2]	8	H'FC21	HCAN	16	5
Message data 14[3]	MD14[3]	8	H'FC22	HCAN	16	5
Message data 14[4]	MD14[4]	8	H'FC23	HCAN	16	5
Message data 14[5]	MD14[5]	8	H'FC24	HCAN	16	5
Message data 14[6]	MD14[6]	8	H'FC25	HCAN	16	5
Message data 14[7]	MD14[7]	8	H'FC26	HCAN	16	5
Message data 14[8]	MD14[8]	8	H'FC27	HCAN	16	5
Message data 15[1]	MD15[1]	8	H'FC28	HCAN	16	5
Message data 15[2]	MD15[2]	8	H'FC29	HCAN	16	5
Message data 15[3]	MD15[3]	8	H'FC2A	HCAN	16	5
Message data 15[4]	MD15[4]	8	H'FC2B	HCAN	16	5
Message data 15[5]	MD15[5]	8	H'FC2C	HCAN	16	5

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Message data 15[6]	MD15[6]	8	H'FC2D	HCAN	16	5
Message data 15[7]	MD15[7]	8	H'FC2E	HCAN	16	5
Message data 15[8]	MD15[8]	8	H'FC2F	HCAN	16	5
D/A data register 0	DADR0	8	H'FDAC	D/A converter	8	2
D/A data register 1	DADR1	8	H'FDAD	D/A converter	8	2
D/A control register	DACR	8	H'FDAE	D/A converter	8	2
Timer control register_2	TCR_2	8	H'FDC0	TMR_2	8/16	2
Timer control register_3	TCR_3	8	H'FDC1	TMR_3	8/16	2
Timer control/status register_2	TCSR_2	8	H'FDC2	TMR_2	8/16	2
Timer control/status register_3	TCSR_3	8	H'FDC3	TMR_3	8/16	2
Time constant register A_2	TCORA_2	8	H'FDC4	TMR_2	8/16	2
Time constant register A_3	TCORA_3	8	H'FDC5	TMR_3	8/16	2
Time constant register B_2	TCORB_2	8	H'FDC6	TMR_2	8/16	2
Time constant register B_3	TCORB_3	8	H'FDC7	TMR_3	8/16	2
Timer counter_2	TCNT_2	8	H'FDC8	TMR_2	8/16	2
Timer counter_3	TCNT_3	8	H'FDC9	TMR_3	8/16	2
Serial mode register_3	SMR_3	8	H'FDD0	SCI_3	8	2
Bit rate register_3	BRR_3	8	H'FDD1	SCI_3	8	2
Serial control register_3	SCR_3	8	H'FDD2	SCI_3	8	2
Transmit data register_3	TDR_3	8	H'FDD3	SCI_3	8	2
Serial status register_3	SSR_3	8	H'FDD4	SCI_3	8	2
Receive data register_3	RDR_3	8	H'FDD5	SCI_3	8	2
Smart card mode register_3	SCMR_3	8	H'FDD6	SCI_3	8	2
Serial mode register_4	SMR_4	8	H'FDD8	SCI_4	8	2
Bit rate register_4	BRR_4	8	H'FDD9	SCI_4	8	2
Serial control register_4	SCR_4	8	H'FDDA	SCI_4	8	2
Transmit data register_4	TDR_4	8	H'FDDB	SCI_4	8	2
Serial status register_4	SSR_4	8	H'FDDC	SCI_4	8	2
Receive data register_4	RDR_4	8	H'FDDD	SCI_4	8	2
Smart card mode register_4	SCMR_4	8	H'FDDE	SCI_4	8	2

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
System control register 2	SYSCR2	8	H'FDE2	FLASH	8	2
Standby control register	SBYCR	8	H'FDE4	SYSTEM	8	2
System control register	SYSCR	8	H'FDE5	SYSTEM	8	2
System clock control register	SCKCR	8	H'FDE6	SYSTEM	8	2
Mode control register	MDCR	8	H'FDE7	SYSTEM	8	2
Module stop control register A	MSTPCRA	8	H'FDE8	SYSTEM	8	2
Module stop control register B	MSTPCRB	8	H'FDE9	SYSTEM	8	2
Module stop control register C	MSTPCRC	8	H'FDEA	SYSTEM	8	2
Pin function control register	PFCR	8	H'FDEB	BSC	8	2
Low power control register	LPWRCR	8	H'FDEC	SYSTEM	8	2
Break address register A	BARA	32	H'FE00	PBC	8/16	2
Break address register B	BARB	32	H'FE04	PBC	8/16	2
Break control register A	BCRA	8	H'FE08	PBC	8/16	2
Break control register B	BCRB	8	H'FE09	PBC	8/16	2
IRQ sense control register H	ISCRH	8	H'FE12	INT	8	2
IRQ sense control register L	ISCRL	8	H'FE13	INT	8	2
IRQ enable register	IER	8	H'FE14	INT	8	2
IRQ status register	ISR	8	H'FE15	INT	8	2
DTC enable register A	DTCERA	8	H'FE16	DTC	8	2
DTC enable register B	DTCERB	8	H'FE17	DTC	8	2
DTC enable register C	DTCERC	8	H'FE18	DTC	8	2
DTC enable register D	DTCERD	8	H'FE19	DTC	8	2
DTC enable register E	DTCERE	8	H'FE1A	DTC	8	2
DTC enable register F	DTCERF	8	H'FE1B	DTC	8	2
DTC enable register G	DTCERG	8	H'FE1C	DTC	8	2
DTC enable register I	DTCERI	8	H'FE1E	DTC	8	2
DTC vector register	DTVECR	8	H'FE1F	DTC	8	2
Port 1 data direction register	P1DDR	8	H'FE30	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FE31	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FE32	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FE34	PORT	8	2
Port 7 data direction register	P7DDR	8	H'FE36	PORT	8	2
Port A data direction register	PADDR	8	H'FE39	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Port B data direction register	PBDDR	8	H'FE3A	PORT	8	2
Port C data direction register	PCDDR	8	H'FE3B	PORT	8	2
Port D data direction register	PDDDR	8	H'FE3C	PORT	8	2
Port E data direction register	PEDDR	8	H'FE3D	PORT	8	2
Port F data direction register	PFDDR	8	H'FE3E	PORT	8	2
Port G data direction register	PGDDR	8	H'FE3F	PORT	8	2
Port A pull-up MOS control register	PAPCR	8	H'FE40	PORT	8	2
Port B pull-up MOS control register	PBPCR	8	H'FE41	PORT	8	2
Port C pull-up MOS control register	PCPCR	8	H'FE42	PORT	8	2
Port D pull-up MOS control register	PDPCR	8	H'FE43	PORT	8	2
Port E pull-up MOS control register	PEPCR	8	H'FE44	PORT	8	2
Port 3 open drain control register	P3ODR	8	H'FE46	PORT	8	2
Port A open drain control register	PAODR	8	H'FE47	PORT	8	2
Timer control register_3	TCR_3	8	H'FE80	TPU_3	8/16	2
Timer mode register_3	TMDR_3	8	H'FE81	TPU_3	8/16	2
Timer I/O control register H_3	TIORH_3	8	H'FE82	TPU_3	8/16	2
Timer I/O control register L_3	TIORL_3	8	H'FE83	TPU_3	8/16	2
Timer interrupt enable register _3	TIER_3	8	H'FE84	TPU_3	8/16	2
Timer status register_3	TSR_3	8	H'FE85	TPU_3	8/16	2
Timer counter_3	TCNT_3	16	H'FE86	TPU_3	16	2
Timer general register A_3	TGRA_3	16	H'FE88	TPU_3	16	2
Timer general register B_3	TGRB_3	16	H'FE8A	TPU_3	16	2
Timer general register C_3	TGRC_3	16	H'FE8C	TPU_3	16	2
Timer general register D_3	TGRD_3	16	H'FE8E	TPU_3	16	2
Timer control register_4	TCR_4	8	H'FE90	TPU_4	8/16	2
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	8/16	2
Timer I/O control register_4	TIOR_4	8	H'FE92	TPU_4	8/16	2
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	8/16	2
Timer status register_4	TSR_4	8	H'FE95	TPU_4	8/16	2
Timer counter_4	TCNT_4	16	H'FE96	TPU_4	16	2
Timer general register A_4	TGRA_4	16	H'FE98	TPU_4	16	2
Timer general register B_4	TGRB_4	16	H'FE9A	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FEA0	TPU_5	8/16	2

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	8/16	2
Timer I/O control register_5	TIOR_5	8	H'FEA2	TPU_5	8/16	2
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	8/16	2
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	8/16	2
Timer counter_5	TCNT_5	16	H'FEA6	TPU_5	16	2
Timer general register A_5	TGRA_5	16	H'FEA8	TPU_5	16	2
Timer general register B_5	TGRB_5	16	H'FEAA	TPU_5	16	2
Timer start register	TSTR	8	H'FEB0	TPU	8/16	2
Timer synchro register	TSYR	8	H'FEB1	TPU	8/16	2
Interrupt priority register A	IPRA	8	H'FEC0	INT	8	2
Interrupt priority register B	IPRB	8	H'FEC1	INT	8	2
Interrupt priority register C	IPRC	8	H'FEC2	INT	8	2
Interrupt priority register D	IPRD	8	H'FEC3	INT	8	2
Interrupt priority register E	IPRE	8	H'FEC4	INT	8	2
Interrupt priority register F	IPRF	8	H'FEC5	INT	8	2
Interrupt priority register G	IPRG	8	H'FEC6	INT	8	2
Interrupt priority register H	IPRH	8	H'FEC7	INT	8	2
Interrupt priority register I	IPRI	8	H'FEC8	INT	8	2
Interrupt priority register J	IPRJ	8	H'FEC9	INT	8	2
Interrupt priority register K	IPRK	8	H'FECA	INT	8	2
Interrupt priority register L	IPRL	8	H'FECB	INT	8	2
Interrupt priority register M	IPRM	8	H'FECC	INT	8	2
Interrupt priority register O	IPRO	8	H'FECE	INT	8	2
Bus width control register	ABWCR	8	H'FED0	BSC	8	2
Access state control register	ASTCR	8	H'FED1	BSC	8	2
Wait control register H	WCRH	8	H'FED2	BSC	8	2
Wait control register L	WCRL	8	H'FED3	BSC	8	2
Bus control register H	BCRH	8	H'FED4	BSC	8	2
Bus control register L	BCRL	8	H'FED5	BSC	8	2
RAM emulation register	RAMER	8	H'FEDB	FLASH	8	2
Port 1 data register	P1DR	8	H'FF00	PORT	8	2
Port 2 data register	P2DR	8	H'FF01	PORT	8	2
Port 3 data register	P3DR	8	H'FF02	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Port 5 data register	P5DR	8	H'FF04	PORT	8	2
Port 7 data register	P7DR	8	H'FF06	PORT	8	2
Port A data register	PADR	8	H'FF09	PORT	8	2
Port B data register	PBDR	8	H'FF0A	PORT	8	2
Port C data register	PCDR	8	H'FF0B	PORT	8	2
Port D data register	PDDR	8	H'FF0C	PORT	8	2
Port E data register	PEDR	8	H'FF0D	PORT	8	2
Port F data register	PFDR	8	H'FF0E	PORT	8	2
Port G data register	PGDR	8	H'FF0F	PORT	8	2
Timer control register_0	TCR_0	8	H'FF10	TPU_0	8/16	2
Timer mode register_0	TMDR_0	8	H'FF11	TPU_0	8/16	2
Timer I/O control register H_0	TIORH_0	8	H'FF12	TPU_0	8/16	2
Timer I/O control register L_0	TIORL_0	8	H'FF13	TPU_0	8/16	2
Timer interrupt enable register_0	TIER_0	8	H'FF14	TPU_0	8/16	2
Timer status register_0	TSR_0	8	H'FF15	TPU_0	8/16	2
Timer counter_0	TCNT_0	16	H'FF16	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FF18	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FF1A	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FF1C	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FF1E	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FF20	TPU_1	8/16	2
Timer mode register_1	TMDR_1	8	H'FF21	TPU_1	8/16	2
Timer I/O control register_1	TIOR_1	8	H'FF22	TPU_1	8/16	2
Timer interrupt enable register_1	TIER_1	8	H'FF24	TPU_1	8/16	2
Timer status register_1	TSR_1	8	H'FF25	TPU_1	8/16	2
Timer counter_1	TCNT_1	16	H'FF26	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FF28	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FF2A	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FF30	TPU_2	8/16	2
Timer mode register_2	TMDR_2	8	H'FF31	TPU_2	8/16	2
Timer I/O control register_2	TIOR_2	8	H'FF32	TPU_2	8/16	2
Timer interrupt enable register_2	TIER_2	8	H'FF34	TPU_2	8/16	2
Timer status register_2	TSR_2	8	H'FF35	TPU_2	8/16	2

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Timer counter_2	TCNT_2	16	H'FF36	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FF38	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FF3A	TPU_2	16	2
Timer control register_0	TCR_0	8	H'FF68	TMR_0	8/16	2
Timer control register_1	TCR_1	8	H'FF69	TMR_1	8/16	2
Timer control/status register_0	TCSR_0	8	H'FF6A	TMR_0	8/16	2
Timer control/status register_1	TCSR_1	8	H'FF6B	TMR_1	8/16	2
Timer constant register A_0	TCORA_0	8	H'FF6C	TMR_0	8/16	2
Timer constant register A_1	TCORA_1	8	H'FF6D	TMR_1	8/16	2
Timer constant register B_0	TCORB_0	8	H'FF6E	TMR_0	8/16	2
Timer constant register B_1	TCORB_1	8	H'FF6F	TMR_1	8/16	2
Timer counter_0	TCNT_0	8	H'FF70	TMR_0	8/16	2
Timer counter_1	TCNT_1	8	H'FF71	TMR_1	8/16	2
Timer control/status register_0	TCSR_0	8	H'FF74	WDT_0	16	2
Timer counter_0	TCNT_0	8	H'FF74 (write)	WDT_0	16	2
Timer counter_0	TCNT_0	8	H'FF75 (read)	WDT_0	16	2
Reset control/status register	RSTCSR	8	H'FF76 (write)	WDT_0	16	2
Reset control/status register	RSTCSR	8	H'FF77 (read)	WDT_0	16	2
Serial mode register_0	SMR_0	8	H'FF78	SCI_0	8	2
Bit rate register_0	BRR_0	8	H'FF79	SCI_0	8	2
Serial control register_0	SCR_0	8	H'FF7A	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FF7B	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FF7C	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FF7D	SCI_0	8	2
Smart card mode register_0	SCMR_0	8	H'FF7E	SCI_0	8	2
Serial mode register_1	SMR_1	8	H'FF80	SCI_1	8	2
Bit rate register_1	BRR_1	8	H'FF81	SCI_1	8	2
Serial control register_1	SCR_1	8	H'FF82	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF83	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF84	SCI_1	8	2

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Receive data register_1	RDR_1	8	H'FF85	SCI_1	8	2
Smart card mode register_1	SCMR_1	8	H'FF86	SCI_1	8	2
Serial mode register_2	SMR_2	8	H'FF88	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FF89	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FF8A	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FF8B	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FF8C	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FF8D	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FF8E	SCI_2	8	2
A/D data register AH	ADDRAH	8	H'FF90	A/D	8	2
A/D data register AL	ADDRAL	8	H'FF91	A/D	8	2
A/D data register BH	ADDRBH	8	H'FF92	A/D	8	2
A/D data register BL	ADDRBL	8	H'FF93	A/D	8	2
A/D data register CH	ADDRCH	8	H'FF94	A/D	8	2
A/D data register CL	ADDRCL	8	H'FF95	A/D	8	2
A/D data register DH	ADDRDH	8	H'FF96	A/D	8	2
A/D data register DL	ADDRDL	8	H'FF97	A/D	8	2
A/D control/status register	ADCSR	8	H'FF98	A/D	8	2
A/D control register	ADCR	8	H'FF99	A/D	8	2
Timer control/status register_1	TCSR_1	8	H'FFA2	WDT_1	16	2
Timer counter_1	TCNT_1	8	H'FFA2 (write)	WDT_1	16	2
Timer counter_1	TCNT_1	8	H'FFA3 (read)	WDT_1	16	2
Flash code control status register	FCCS	8	H'FFA4	FLASH	8	2
Flash program code select register	FPCS	8	H'FFA5	FLASH	8	2
Flash erase code select register	FECS	8	H'FFA6	FLASH	8	2
Flash key code register	FKEY	8	H'FFA8	FLASH	8	2
Flash MAT select register	FMATS	8	H'FFA9	FLASH	8	2
Flash transfer destination address register	FTDAR	8	H'FFAA	FLASH	8	2
Flash vector address code control register	FVACR	8	H'FFAB	FLASH	8	2
Flash vector address data register R	FVADRR	8	H'FFAC	FLASH	8	2

Register Name	Abbreviation	Number of Bits	Address*[1]	Module	Data Bus Width	Number of Access States
Flash vector address data register E	FVADRE	8	H'FFAD	FLASH	8	2
Flash vector address data register H	FVADRH	8	H'FFAE	FLASH	8	2
Flash vector address data register L	FVADRL	8	H'FFAF	FLASH	8	2
Port 1 register	PORT1	8	H'FFB0	PORT	8	2
Port 2 register	PORT2	8	H'FFB1	PORT	8	2
Port 3 register	PORT3	8	H'FFB2	PORT	8	2
Port 4 register	PORT4	8	H'FFB3	PORT	8	2
Port 5 register	PORT5	8	H'FFB4	PORT	8	2
Port 7 register	PORT7	8	H'FFB6	PORT	8	2
Port 9 register	PORT9	8	H'FFB8	PORT	8	2
Port A register	PORTA	8	H'FFB9	PORT	8	2
Port B register	PORTB	8	H'FFBA	PORT	8	2
Port C register	PORTC	8	H'FFBB	PORT	8	2
Port D register	PORTD	8	H'FFBC	PORT	8	2
Port E register	PORTE	8	H'FFBD	PORT	8	2
Port F register	PORTF	8	H'FFBE	PORT	8	2
Port G register	PORTG	8	H'FFBF	PORT	8	2

Notes: 1. Lower 16 bits of the address.

2. Allocated on the on-chip RAM. 32-bit bus when DTC accesses as register information, and 16-bit in other cases.

23.2 Register Bits

On-chip peripheral module register addresses and bit names are shown in the following table.

16-bit registers are shown in two rows of 8 bits.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC
SAR	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MRB	CHNE	DISEL	—	—	—	—	—	—	
DAR	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
CRA	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
CRB	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IECTR	IEE	IOL	DEE	CKS1	RE	LUEE	CKS0	—	IEB
IECMR	—	—	—	—	—	CMD2	CMD1	CMD0	
IEMCR	SS	RN2	RN1	RN0	CTL3	CTL2	CTL1	CTL0	
IEAR1	IAR3	IAR2	IAR1	IAR0	IMD1	IMD0	—	STE	
IEAR2	IAR11	IAR10	IAR9	IAR8	IAR7	IAR6	IAR5	IAR4	
IESA1	ISA3	ISA2	ISA1	ISA0	—	—	—	—	
IESA2	ISA11	ISA10	ISA9	ISA8	ISA7	ISA6	ISA5	ISA4	
IETBFL	TBFL7	TBFL6	TBFL5	TBFL4	TBFL3	TBFL2	TBFL1	TBFL0	
IETBR	TBR7	TBR6	TBR5	TBR4	TBR3	TBR2	TBR1	TBR0	
IEMA1	IMA3	IMA2	IMA1	IMA0	—	—	—	—	
IEMA2	IMA11	IMA10	IMA9	IMA8	IMA7	IMA6	IMA5	IMA4	
IERCTL	—	—	—	—	RCTL3	RCTL2	RCTL1	RCTL0	
IERBFL	RBFL7	RBFL6	RBFL5	RBFL4	RBFL3	RBFL2	RBFL1	RBFL0	
IERBR	RBR7	RBR6	RBR5	RBR4	RBR3	RBR2	RBR1	RBR0	
IELA1	ILA7	ILA6	ILA5	ILA4	ILA3	ILA2	ILA1	ILA0	
IELA2	—	—	—	—	ILA11	ILA10	ILA9	ILA8	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IEFLG	CMX	MRQ	SRQ	SRE	LCK	—	RSS	GG	IEB
IETSR	TxRDY	—	—	—	IRA	TxS	TxF	TxE	
IEIET	TxRDYE	—	—	—	IRAE	TxSE	TxFE	TxEE	
IETEF	—	—	—	AL	UE	TIME	RO	ACK	
IERSR	RxRDY	—	—	—	—	RxS	RxF	RxE	
IEIER	RxRDYE	—	—	—	—	RxSE	RxFE	RxEE	
IEREF	—	—	—	—	OVE	RTME	DLE	PE	
PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR	PORT
PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR	
PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR	
PJDR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR	
PORTH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	
PORTJ	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	
ICCR1_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_0
ICCR2_0	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
ICMR_0	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT_0	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_0	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICCR1_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_1
ICCR2_1	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
ICMR_1	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT_1	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_1	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
MCR	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	HCAN
GSR	—	—	—	—	GSR3	GSR2	GSR1	GSR0	
BCR	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0	
	BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MBCR	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—	HCAN
	MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8	
TXPR	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—	
	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8	
TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—	
	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8	
TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—	
	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8	
ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—	
	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8	
RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0	
	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8	
RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8	
IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
	—	—	—	IRR12	—	—	IRR9	IRR8	
MBIMR	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0	
	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8	
IMR	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—	
	—	—	—	IMR12	—	—	IMR9	IMR8	
REC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
UMSR	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0	
	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8	
LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0	
	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8	
LAFMH	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0	
	LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8	
MC0[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC0[2]	—	—	—	—	—	—	—	—	
MC0[3]	—	—	—	—	—	—	—	—	
MC0[4]	—	—	—	—	—	—	—	—	
MC0[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MC0[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	HCAN
MC0[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC0[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC1[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC1[2]	—	—	—	—	—	—	—	—	
MC1[3]	—	—	—	—	—	—	—	—	
MC1[4]	—	—	—	—	—	—	—	—	
MC1[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC1[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC1[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC1[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC2[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC2[2]	—	—	—	—	—	—	—	—	
MC2[3]	—	—	—	—	—	—	—	—	
MC2[4]	—	—	—	—	—	—	—	—	
MC2[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC2[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC2[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC2[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC3[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC3[2]	—	—	—	—	—	—	—	—	
MC3[3]	—	—	—	—	—	—	—	—	
MC3[4]	—	—	—	—	—	—	—	—	
MC3[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC3[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC3[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC3[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC4[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC4[2]	—	—	—	—	—	—	—	—	
MC4[3]	—	—	—	—	—	—	—	—	
MC4[4]	—	—	—	—	—	—	—	—	
MC4[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC4[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MC4[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	HCAN
MC4[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC5[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC5[2]	—	—	—	—	—	—	—	—	
MC5[3]	—	—	—	—	—	—	—	—	
MC5[4]	—	—	—	—	—	—	—	—	
MC5[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC5[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC5[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC5[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC6[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC6[2]	—	—	—	—	—	—	—	—	
MC6[3]	—	—	—	—	—	—	—	—	
MC6[4]	—	—	—	—	—	—	—	—	
MC6[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC6[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC6[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC6[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC7[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC7[2]	—	—	—	—	—	—	—	—	
MC7[3]	—	—	—	—	—	—	—	—	
MC7[4]	—	—	—	—	—	—	—	—	
MC7[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC7[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC7[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC7[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC8[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC8[2]	—	—	—	—	—	—	—	—	
MC8[3]	—	—	—	—	—	—	—	—	
MC8[4]	—	—	—	—	—	—	—	—	
MC8[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC8[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC8[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MC8[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	HCAN
MC9[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC9[2]	—	—	—	—	—	—	—	—	
MC9[3]	—	—	—	—	—	—	—	—	
MC9[4]	—	—	—	—	—	—	—	—	
MC9[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC9[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC9[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC9[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC10[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC10[2]	—	—	—	—	—	—	—	—	
MC10[3]	—	—	—	—	—	—	—	—	
MC10[4]	—	—	—	—	—	—	—	—	
MC10[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC10[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC10[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC10[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC11[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC11[2]	—	—	—	—	—	—	—	—	
MC11[3]	—	—	—	—	—	—	—	—	
MC11[4]	—	—	—	—	—	—	—	—	
MC11[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC11[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC11[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC11[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC12[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC12[2]	—	—	—	—	—	—	—	—	
MC12[3]	—	—	—	—	—	—	—	—	
MC12[4]	—	—	—	—	—	—	—	—	
MC12[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC12[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC12[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC12[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MC13[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN
MC13[2]	—	—	—	—	—	—	—	—	
MC13[3]	—	—	—	—	—	—	—	—	
MC13[4]	—	—	—	—	—	—	—	—	
MC13[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC13[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC13[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC13[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC14[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC14[2]	—	—	—	—	—	—	—	—	
MC14[3]	—	—	—	—	—	—	—	—	
MC14[4]	—	—	—	—	—	—	—	—	
MC14[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC14[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC14[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC14[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MC15[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MC15[2]	—	—	—	—	—	—	—	—	
MC15[3]	—	—	—	—	—	—	—	—	
MC15[4]	—	—	—	—	—	—	—	—	
MC15[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16	
MC15[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	
MC15[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0	
MC15[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	
MD0[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD0[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD0[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD0[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD0[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD0[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD0[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD0[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD1[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MD1[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	HCAN
MD1[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD1[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD1[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD1[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD1[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD1[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD2[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD3[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD4[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD5[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD5[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MD5[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	HCAN
MD5[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD5[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD5[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD5[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD5[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD6[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD7[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD8[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD9[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD9[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD9[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MD9[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	HCAN
MD9[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD9[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD9[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD9[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD10[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD11[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD12[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD13[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD13[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD13[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD13[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MD13[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	HCAN
MD13[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD13[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD13[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD14[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DADR0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	D/A converter
DADR1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
TCR_2	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_2
TCR_3	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_3
TCSR_2	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	TMR_2
TCSR_3	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	TMR_3
TCORA_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_2
TCORA_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_3
TCORB_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_2
TCORB_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_3
TCNT_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_2
TCNT_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_3

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SMR_3*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_3
BRR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_3*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_3	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_4*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_4
BRR_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_4*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_4	—	—	—	—	SDIR	SINV	—	SMIF	
SYSCR2	—	—	—	—	FLSHE	—	—	—	FLASH
SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	SYSTEM
SYSCR	—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME	
SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
PFCR	—	—	BUZZE	—	AE3	AE2	AE1	AE0	BSC
LPWRCR	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0	SYSTEM
BARA	—	—	—	—	—	—	—	—	PBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
BARB	—	—	—	—	—	—	—	—	PBC
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BCRA	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA	
BCRB	CMFB	CDB	BAMRB2	BAMRB1	BAMRB0	CSELB1	CSELB0	BIEB	
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	INT
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	—	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	—	—	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	—	—	—	—	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	—	DTCEG6	DTCEG5	—	DTCEG3	DTCEG2	—	—	
DTCERI	DTCEI7	DTCEI6	DTCEI5	DTCEI4	—	—	—	—	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P5DDR	—	—	—	—	—	P52DDR	P51DDR	P50DDR	
P7DDR	P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR	
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR*2	PF2DDR*2	PF1DDR	PF0DDR	
PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PAPCR	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	PORT
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	P37ODR	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR	
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_3
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_4
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	INT
IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRC	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRD	—	IPR6	IPR5	IPR4	—	—	—	—	
IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRI	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRJ	—	—	—	—	—	IPR2	IPR1	IPR0	
IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRL	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRO	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WCRH	W71	W70	W61	W60	W51	W50	W41	W40	
WCRL	W31	W30	W21	W20	W11	W10	W01	W00	
BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—	
BCRL	BRLE	—	—	—	—	—	—	WAITE	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	FLASH
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	PORT
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P5DR	—	—	—	—	—	P52DR	P51DR	P50DR	
P7DR	P77DR	P76DR	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR	
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PGDR	—	—	—	PG4DR	PG3DR*[2]	PG2DR*[2]	PG1DR	PG0DR	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	TPU_1
TCNT_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_1
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	TMR_0
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	TMR_1
TCORA_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_0
TCORA_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_1
TCORB_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_0
TCORB_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_1
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_0
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	TMR_1
TCSR_0	OVF	WT / $\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT_0
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
SMR_0*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_0

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
BRR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SCI_0
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_0*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_1*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_1
BRR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_1*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_2
BRR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_2*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D converter
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCSR_1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT_1
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
FCCS	—	—	—	FLER	—	—	—	SCO	FLASH
FPCS	—	—	—	—	—	—	—	PPVS	
FECS	—	—	—	—	—	—	—	EPVB	
FKEY	K7	K6	K5	K4	K3	K2	K1	K0	
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
FVACR	FVCHGE	—	—	—	—	—	—	—	
FVADRR	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	
FVADRE	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	
FVADRH	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
FVADRL	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT2	P27	P26	P25	P24	P23	P22	P21	P20	
PORT3	P37	P36	P35	P34	P33	P32	P31	P30	
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT5	—	—	—	—	—	P52	P51	P50	
PORT7	P77	P76	P75	P74	P73	P72	P71	P70	
PORT9	P97	P96	P95	P94	P93	P92	P91	P90	
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	
PORTG	—	—	—	PG4	PG3*[2]	PG2*[2]	PG1	PG0	

Notes: 1. Some bits have different names in normal mode and in smart card interface mode. The bit name in smart card interface mode is enclosed in parentheses.

2. Reserved in the H8S/2556.

23.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
MRA	Initialized	—	—	—	—	—	—	—	—	Initialized	DTC
SAR	Initialized	—	—	—	—	—	—	—	—	Initialized	
MRB	Initialized	—	—	—	—	—	—	—	—	Initialized	
DAR	Initialized	—	—	—	—	—	—	—	—	Initialized	
CRA	Initialized	—	—	—	—	—	—	—	—	Initialized	
CRB	Initialized	—	—	—	—	—	—	—	—	Initialized	
IECTR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	IEB
IECMR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEMCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEAR1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEAR2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IESA1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IESA2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IETBFL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IETBR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEMA1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEMA2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IERCTL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IERBFL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IERBR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IELA1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IELA2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEFLG	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IETSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEIET	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IETEF	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IERSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEIER	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IEREF	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
PHDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	PORT
PJDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PHDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
PJDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	PORT
PORTH	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PORTJ	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
ICCR1_0	Initialized	—	—	—	—	—	—	—	—	Initialized	IIC2_0
ICCR2_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICMR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICIER_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICSR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
SAR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICDRT_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICDRR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICCR1_1	Initialized	—	—	—	—	—	—	—	—	Initialized	IIC2_1
ICCR2_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICMR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICIER_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICSR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
SAR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICDRT_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICDRR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
MCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
GSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
BCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MBCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TXPR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TXCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TXACK	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ABACK	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RXPR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RFPR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IRR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MBIMR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
IMR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
REC	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TEC	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
UMSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
LAFML	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
LAFMH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC0[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC1[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC2[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC3[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*[1]	Subsleep*[1]	Software Standby	Hardware Standby	Module
MC3[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MC4[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC4[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC5[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC6[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC7[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
MC8[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MC8[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC8[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC9[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC10[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC11[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*[1]	Subsleep*[1]	Software Standby	Hardware Standby	Module
MC12[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MC12[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC12[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC13[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC14[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MC15[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
MD0[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MD0[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD0[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD1[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD2[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD3[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD4[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*[1]	Subsleep*[1]	Software Standby	Hardware Standby	Module
MD4[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MD5[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD5[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD6[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD7[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD8[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
MD9[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MD9[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD9[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD10[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD11[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD12[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*[1]	Subsleep*[1]	Software Standby	Hardware Standby	Module
MD13[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	HCAN
MD13[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD13[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD14[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
MD15[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
DADR0	Initialized	—	—	—	—	—	—	—	—	Initialized	D/A
DADR1	Initialized	—	—	—	—	—	—	—	—	Initialized	
DACR	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_2
TCR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_3
TCSR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_2
TCSR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_3
TCORA_2	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_2
TCORA_3	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_3
TCORB_2	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_2
TCORB_3	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_3
TCNT_2	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_2
TCNT_3	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_3

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
SMR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_3
BRR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
SCR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
SMR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_4
BRR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
SCR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TDR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
SYSCR2	Initialized	—	—	—	—	—	—	—	—	Initialized	FLASH
SBYCR	Initialized	—	—	—	—	—	—	—	—	Initialized	SYSTEM
SYSCR	Initialized	—	—	—	—	—	—	—	—	Initialized	
SCKCR	Initialized	—	—	—	—	—	—	—	—	Initialized	
MDCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
MSTPCRA	Initialized	—	—	—	—	—	—	—	—	Initialized	
MSTPCRB	Initialized	—	—	—	—	—	—	—	—	Initialized	
MSTPCRC	Initialized	—	—	—	—	—	—	—	—	Initialized	
PFCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	BSC
LPWRCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	SYSTEM
BARA	Initialized*2	—	—	—	—	—	—	—	—	Initialized	PBC
BARB	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
BCRA	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
BCRB	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
ISCRH	Initialized	—	—	—	—	—	—	—	—	Initialized	INT
ISCRL	Initialized	—	—	—	—	—	—	—	—	Initialized	
IER	Initialized	—	—	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	—	—	Initialized	
DTCERA	Initialized	—	—	—	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	—	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
DTCERC	Initialized	—	—	—	—	—	—	—	—	Initialized	DTC
DTCERD	Initialized	—	—	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	—	—	Initialized	
DTCERF	Initialized	—	—	—	—	—	—	—	—	Initialized	
DTCERG	Initialized	—	—	—	—	—	—	—	—	Initialized	
DTCERI	Initialized	—	—	—	—	—	—	—	—	Initialized	
DTVECR	Initialized	—	—	—	—	—	—	—	—	Initialized	
P1DDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	PORT
P2DDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
P3DDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
P5DDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
P7DDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PADDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PBDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PCDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PDDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PEDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PFDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PGDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PAPCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PBPCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PCPCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PDPCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PEPCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
P3ODR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PAODR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
TCR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_3
TMDR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIORH_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIORL_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIER_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TSR_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCNT_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRA_3	Initialized	—	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
TGRB_3	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_3
TGRC_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRD_3	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_4
TMDR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIOR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIER_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TSR_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCNT_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRA_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRB_4	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCR_5	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_5
TMDR_5	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIOR_5	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIER_5	Initialized	—	—	—	—	—	—	—	—	Initialized	
TSR_5	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCNT_5	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRA_5	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRB_5	Initialized	—	—	—	—	—	—	—	—	Initialized	
TSTR	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU
TSYR	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRA	Initialized	—	—	—	—	—	—	—	—	Initialized	INT
IPRB	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRC	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRD	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRE	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRF	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRG	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRH	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRI	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRJ	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRL	Initialized	—	—	—	—	—	—	—	—	Initialized	
IPRM	Initialized	—	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
IPRO	Initialized	—	—	—	—	—	—	—	—	Initialized	INT
ABWCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	BSC
ASTCR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
WCRH	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
WCRL	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
BCRH	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
BCRL	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
RAMER	Initialized	—	—	—	—	—	—	—	—	Initialized	FLASH
P1DR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	PORT
P2DR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
P3DR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
P5DR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
P7DR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PADR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PBDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PCDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PDDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PEDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PFDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
PGDR	Initialized*2	—	—	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRA_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRB_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRC_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRD_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_1
TMDR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIOR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
TIER_1	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_1
TSR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRA_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRB_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_0
TCR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_1
TCSR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_0
TCSR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_1
TCORA_0	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_0
TCORA_1	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_1
TCORB_0	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_0
TCORB_1	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_1
TCNT_0	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_0
TCNT_1	Initialized	—	—	—	—	—	—	—	—	Initialized	TMR_1
TCSR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	WDT_0
TCNT_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
RSTCSR	Initialized	—	—	—	—	—	—	—	—	Initialized	
SMR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_0
BRR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
SCR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	
TDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_0	Initialized	—	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*[1]	Subsleep*[1]	Software Standby	Hardware Standby	Module
SMR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_1
BRR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
SCR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
TDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
SMR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_2
BRR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
SCR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
TDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	—	—	—	—	Initialized	
ADDRAH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	A/D
ADDRAL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRDH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRDL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
TCSR_1	Initialized	—	—	—	—	—	—	—	—	Initialized	WDT_1
TCNT_1	Initialized	—	—	—	—	—	—	—	—	Initialized	
FCCS	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	FLASH
FPCS	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
FECS	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
FKEY	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
FMATS	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
FTDAR	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
FVACR	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
FVADRR	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	

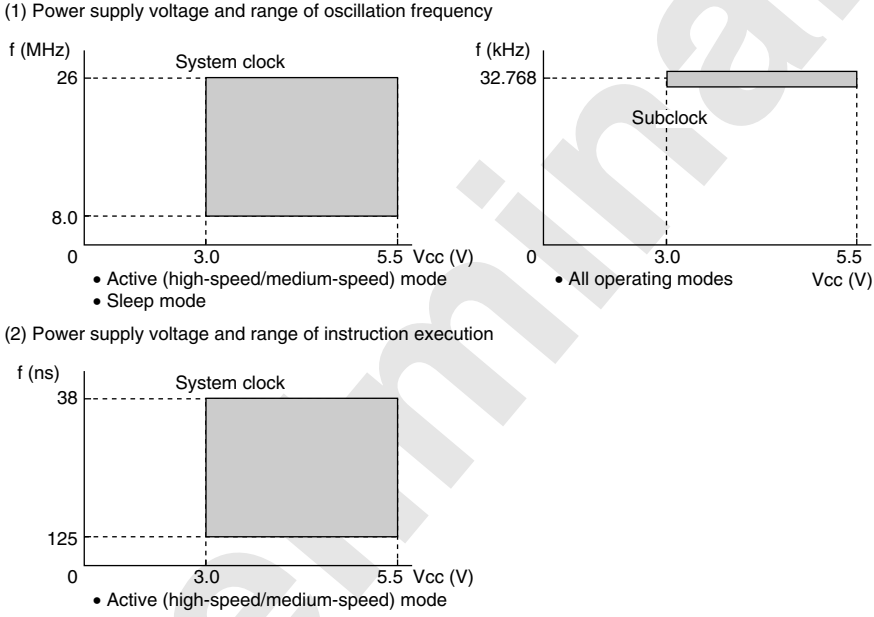
Register Abbreviation	Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive*1	Subsleep*1	Software Standby	Hardware Standby	Module
FVADRE	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	FLASH
FVADRH	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
FVADRL	Initialized	—	—	—	—	—	—	—	Initialized	Initialized	
PORT1	—	—	—	—	—	—	—	—	—	—	PORT
PORT2	—	—	—	—	—	—	—	—	—	—	
PORT3	—	—	—	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	—	—	—	—	
PORT7	—	—	—	—	—	—	—	—	—	—	
PORT9	—	—	—	—	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	—	—	—	—	
PORTG	—	—	—	—	—	—	—	—	—	—	

Notes: – is not initialized.

1. Cannot be used in this LSI.
2. Not initialized by a manual reset.

24.1 Power Supply Voltage and Operating Frequency Range

Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 24.1.



Note: * When the IEBus is used in the H8S/2552 Series, the system clock should be selected among 12 MHz, 12.58 MHz, 18 MHz, 18.87 MHz, 24 MHz, and 25.16 MHz.
When not using the IEBus or in the H8S/2506 Series, the system clock can be selected arbitrarily from 8 MHz to 26 MHz.

Figure 24.1 (1) Power Supply Voltage and Operating Ranges (H8S/2552 Series, H8S/2506 Series)

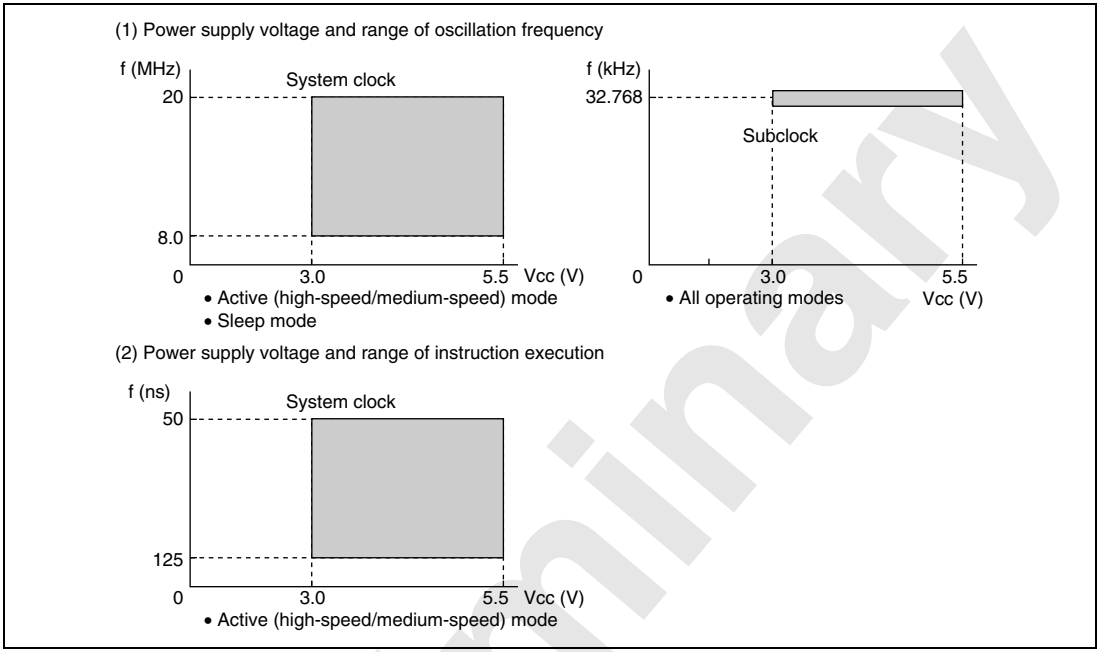


Figure 24.1 (2) Power Supply Voltage and Operating Ranges (H8S/2556 Series)

24.2 Absolute Maximum Ratings

Table 24.1 lists the absolute maximum ratings.

Table 24.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage*[1]	V_{CC}	–0.3 to +7.0	V
	$P1V_{CC}$, $P2V_{CC}$	–0.3 to +7.0	V
Input voltage (port 4, 9)*[2]	V_{in}	–0.3 to AV_{CC} +0.3	V
Input voltage (port 1, 2, 3, 7)*[2]	V_{in}	–0.3 to $P2V_{CC}$ +0.3	V
Input voltage (port 5, A to H, J)*[2]	V_{in}	–0.3 to $P1V_{CC}$ +0.3	V
Input voltage (port HRxD)*[2]*[3]	V_{in}	–0.3 to $P1V_{CC}$ +0.3	V
Input voltage (others)*[2]*[5]	V_{in}	–0.3 to V_{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*[4]	°C
		Wide-range specifications: –40 to +85*[4]	
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum rating are exceeded.

Notes: 1. Do not apply a power supply voltage to the VCL pin. The capacitor should be connected to GND externally.

2. Do not apply a voltage 12 V to any pins. Applying a voltage 12 V can permanently damage to the chip.

3. HRxD is supported in the H8S/2556 Series only. Do not apply a power supply voltage to TRxD because it is an output pin.

4. Operating temperature range for flash memory programming/erasing is Ta = 0 to +75°C.

5. The OSC1 and OSC2 pins should only be connected to the 32.768-kHz crystal resonator. When not connecting to the resonator, the OSC1 pin should be connected to Vss and the OSC2 pin should be open.

24.3 DC Characteristics

Table 24.2 lists the DC characteristics. Table 24.3 lists the permissible output currents. Table 24.4 lists the bus drive characteristics.

Table 24.2 DC Characteristics (1)

Conditions: V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications)*[1], T_a = –40°C to +85°C (wide-range specifications)*[2]

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	IRQ0, IRQ1, IRQ4 to IRQ7	VT^-	$P2V_{CC}$ × 0.2	—	—	V	
		VT^+	—	—	$P2V_{CC}$ × 0.8	V	
		$VT^+ - VT^-$	$P2V_{CC}$ × 0.05	—	—	V	
	IRQ2, IRQ3	VT^-	$P1V_{CC}$ × 0.2	—	—	V	
		VT^+	—	—	$P1V_{CC}$ × 0.8	V	
		$VT^+ - VT^-$	$P1V_{CC}$ × 0.05	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0	V_{IH}	V_{CC} × 0.9	—	V_{CC} + 0.3	V	
	EXTAL		V_{CC} × 0.8	—	V_{CC} + 0.3	V	
	Port 1 to 3, 7		$P2V_{CC}$ × 0.8	—	$P2V_{CC}$ + 0.3	V	
	Port 5, A to H, J, HRxD*[5]		$P1V_{CC}$ × 0.8	—	$P1V_{CC}$ + 0.3	V	
	Port 4, 9		AV_{CC} × 0.8	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0	V_{IL}	- 0.3	—	V_{CC} × 0.1	V	
	Port 4, 9		- 0.3	—	AV_{CC} × 0.2	V	
	NMI, EXTAL, TEST		- 0.3	—	V_{CC} × 0.2	V	
	Port 1 to 3, 7		- 0.3	—	$P2V_{CC}$ × 0.2	V	
	Port 5, A to H, J, HRxD*[5]		- 0.3	—	$P1V_{CC}$ × 0.2	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	Port 1 to 3, 7	V_{OH}	$P2V_{CC}$ - 0.5	—	—	V	I_{OH} = - 200 µA
			$P2V_{CC}$ - 1.0	—	—	V	I_{OH} = - 1 mA
	Port 5, A to H, J, HTxD*[5]		$P1V_{CC}$ - 0.5	—	—	V	I_{OH} = - 200 µA
			$P1V_{CC}$ - 1.0	—	—	V	I_{OH} = - 1 mA
	P34, P35*[3]		$P2V_{CC}$ - 2.7	—	—	V	I_{OH} = - 100 µA
	Output pins except port 1 to 3, 5, 7, A to J, HTxD*[5], P34, and P35		V_{CC} - 0.5	—	—	V	I_{OH} = - 200 µA
			V_{CC} - 1.0	—	—	V	I_{OH} = - 1 mA
Output low voltage	All output pins*[4]	V_{OL}	—	—	0.4	V	I_{OL} = 0.4 mA
			—	—	0.4	V	I_{OL} = 0.8 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC}- 0.5 V
	$\overline{STBY}$, NMI, MD2 to MD0, TEST		—	—	1.0	µA	V_{in} = 0.5 to V_{CC}- 0.5 V
	HRxD*[5]		—	—	1.0	µA	V_{in} = 0.5 to $P1V_{CC}$ - 0.5 V
	Port 4, 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} - 0.5 V
Three-state leakage current (off state)	Port 1 to 3, 7	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to $P2V_{CC}$ - 0.5 V
	Port 5, A to H, J		—	—	1.0	µA	V_{in} = 0.5 to $P1V_{CC}$ - 0.5 V
Input pull-up MOS current	Port A to E	$-I_P$	10	—	300	µA	V_{in} = 0 V

Notes: 1. The regular specifications are supported in the H8S/2506 Series only.

2. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Apply a voltage 3.0 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.
3. P35/SCK1/SCL0 and P34/SDA0 are NMOS push/pull outputs.

   To output high level signal from SCL0 and SDA0 (ICE = 1), pull-up resistance must be connected externally. P35/SCK1 and P34 (ICE = 0) are driven high by NMOS.
4. When IICS = 0 and ICE = 0. To output low when bus drive function is selected is determined in table 24.4, Bus Drive Characteristics.
5. HRxD and HTxD are supported in the H8S/2556 Series only.

Table 24.2 DC Characteristics (2)

Conditions (for H8S/2552 Series and H8S/2506 Series):
V_{CC} = 3.0V to 5.5 V, $P1V_{CC}$ = 3.0V to 5.5 V, $P2V_{CC}$ = 3.0V to 5.5 V,
AV_{CC} = 3.0V to 5.5V, V_{ref} = 3.0V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C
(regular specifications)*[1], T_a = –40°C to +85°C (wide-range specifications)*[2]

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30	pF	f = 8 MHz
	P32 to P35		—	—	20	pF	Ta = 25°C
	All input pins except $\overline{RES}$, NMI, and P32 to P35		—	—	15	pF	
Current consumption*[3]	Normal operation	I_{CC}*[5]	—	TBD V_{CC} = 5.0 V	TBD V_{CC} = 5.5 V	mA	f = 26 MHz
	Sleep mode		—	TBD V_{CC} = 5.0 V	TBD V_{CC} = 5.5 V	mA	f = 26 MHz
	All modules stopped		—	TBD	—	mA	f = 26 MHz, V_{CC} = 5.0 V (reference values)
	Medium-speed mode (ϕ/32)		—	TBD	—	mA	f = 26 MHz, V_{CC} = 5.0 V (reference values)
	Watch mode		—	TBD	TBD	µA	Using 32.768 kHz crystal resonator Vcc = 3.0 V
	Standby mode*[4]		—	TBD	TBD	µA	$T_a \leq 50°C$ 32.768 kHz not used
			—	—	TBD		50°C < T_a 32.768 kHz not used
Analog power supply current	During A/D conversion, D/A conversion	AI_{CC}	—	TBD	TBD	mA	
	Waiting for A/D conversion, D/A conversion		—	TBD	TBD	µA	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference current	During A/D conversion, D/A conversion	AI_{CC}	—	TBD	TBD	mA	
	Waiting for A/D conversion, D/A conversion		—	TBD	TBD	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. The regular specifications are supported in the H8S/2506 Series only.

2. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Apply a voltage 3.0 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.
3. Current consumption values are for $P1V_{CC} = P2V_{CC} = AV_{CC} = V_{CC}$, V_{IH} min. = V_{CC} – TBD V, V_{IL} max. = TBD V with all output pins unloaded and the on-chip pull-up MOS in the off state.
4. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min. = $V_{CC} \times 0.9$, and V_{IL} max. = 0.3 V.
5. I_{CC} depends on V_{CC} and f as follows:

   I_{CC} max. = TBD (mA) + TBD (mA/V) × Vcc + TBD (mA/MHz) × f + TBD (mA/(MHz • V)) × V_{CC} × f (normal operation)

   I_{CC} max. = TBD (mA) + TBD (mA/V) × Vcc + TBD (mA/MHz) × f + TBD (mA/(MHz • V)) × V_{CC} × f (sleep mode)

Table 24.2 DC Characteristics (3)

Conditions (for H8S/2556 Series):
V_{CC} = 3.0V to 5.5 V, $P1V_{CC}$ = 3.0V to 5.5 V, $P2V_{CC}$ = 3.0V to 5.5 V, AV_{CC} = 3.0V to 5.5V, V_{ref} = 3.0V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –40°C to +85°C (wide-range specifications)*[1]

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30	pF	f = 8 MHz
	P32 to P35		—	—	20	pF	Ta = 25°C
	All input pins except $\overline{RES}$, NMI, and P32 to P35		—	—	15	pF	
Current consumption*[2]	Normal operation	I_{CC}*[4]	—	TBD V_{CC} =5.0 V	TBD V_{CC} = 5.5 V	mA	f = 20 MHz
	Sleep mode		—	TBD V_{CC} = 5.0 V	TBD V_{CC} = 5.5 V	mA	f = 20 MHz
	All modules stopped		—	TBD	—	mA	f = 20 MHz, V_{CC} = 5.0 V (reference values)
	Medium-speed mode (ϕ/32)		—	TBD	—	mA	f = 26 MHz, V_{CC} = 5.0 V (reference values)
	Watch mode		—	TBD	TBD	µA	Using 32.768 kHz crystal resonator Vcc = 3.0 V
	Standby mode*[3]		—	TBD	TBD	µA	$T_a \leq$ 50°C 32.768 kHz not used
			—	—	TBD		50°C < T_a 32.768 kHz not used
Analog power supply current	During A/D conversion, D/A conversion	AI_{CC}	—	TBD	TBD	mA	
	Waiting for A/D conversion, D/A conversion		—	TBD	TBD	µA	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference current	During A/D conversion, D/A conversion	AI_{CC}	—	TBD	TBD	mA	
	Waiting for A/D conversion, D/A conversion		—	TBD	TBD	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Apply a voltage 3.0 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

2. Current consumption values are for $P1V_{CC} = P2V_{CC} = AV_{CC} = V_{CC}$, V_{IH} min. = V_{CC} – TBD V, V_{IL} max. = TBD V with all output pins unloaded and the on-chip pull-up MOS in the off state.

3. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min. = $V_{CC} \times 0.9$, and V_{IL} max. = 0.3 V.

4. I_{CC} depends on V_{CC} and f as follows:

   I_{CC} max. = TBD (mA) + TBD (mA/V) × Vcc + TBD (mA/MHz) × f + TBD (mA/(MHz • V)) × V_{CC} × f (normal operation)

   I_{CC} max. = TBD (mA) + TBD (mA/V) × Vcc + TBD (mA/MHz) × f + TBD (mA/(MHz • V)) × V_{CC} × f (sleep mode)

Table 24.3 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL1, SCL0, SDA1, SDA0	I_{OL}	—	—	10	mA
	Output pins except SCL1, SCL0, SDA1, and SDA0		—	—	1.0	mA
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	60	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	1.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	30	mA

Note: To protect chip reliability, do not exceed the output current values in table 24.3.

* The regular specifications are supported in the H8S/2506 Series only.

Table 24.4 Bus Drive Characteristics

Conditions: V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C (regular specifications)*[1], Ta = –40°C to +85°C (wide-range specifications)*[2], Target pins: SCL1, SCL0, SDA1, SDA0

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	VT^-	$P2V_{CC} \times 0.3$	—	—	V	$P2V_{CC}$ = 3.0 to 5.5 V
	VT^+	—	—	$P2V_{CC} \times 0.7$		$P2V_{CC}$ = 3.0 to 5.5 V
	$VT^+ - VT^-$	$P2V_{CC} \times 0.05$	—	—		$P2V_{CC}$ = 3.0 to 5.5 V
Input high voltage	V_{IH}	$P2V_{CC} \times 0.7$	—	$P2V_{CC} + 0.5$	V	$P2V_{CC}$ = 3.0 to 5.5 V
Input low voltage	V_{IL}	-0.5	—	$P2V_{CC} \times 0.3$	V	$P2V_{CC}$ = 3.0 to 5.5 V
Output low voltage	V_{OL}	—	—	0.5	V	I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA
Input capacitance	C_{IN}	—	—	20	pF	V_{IN} = 0 V, f = 8 MHz, T_a=25°C
Three-state leakage current (off state)	$\lvert I_{STI} \rvert$	—	—	1.0	µA	V_{IN}=0.5 to $P2V_{CC}$-0.5 V
SDL, SDA output fall time	t_{Of}	20 + 0.1Cb	—	250	ns	$P2V_{CC}$ = 3.0 to 5.5 V

Notes: 1. The regular specifications are supported in the H8S/2506 Series only.
2. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Apply a voltage 3.0 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$
3. Test Conditions are for $P1V_{CC}$ = $P2V_{CC}$ = AV_{CC} = V_{CC}.

24.4 AC Characteristics

Figure 24.2 shows the test conditions for the AC characteristics.

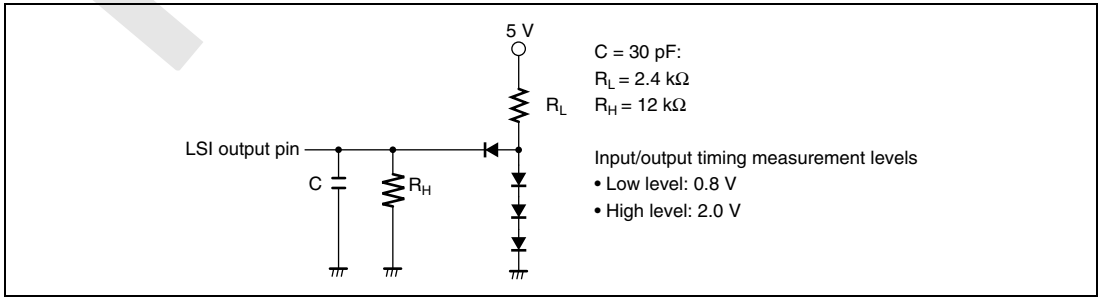


Figure 24.2 Output Load Circuit

24.4.1 Power-On/Off Timing

Table 24.5 Power-On/Off Timing

Condition A (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C (wide-range specifications)

Condition B (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 20 MHz, T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Time taken to switch V_{CC} on	t_{VCCS}	0	—	ms	Figure 24.3
V_{CC} hold time when PV_{CC} is switched off	t_{VCCH}	0	—	ms	

Note:* The regular specifications are supported in the H8S/2506 Series only.

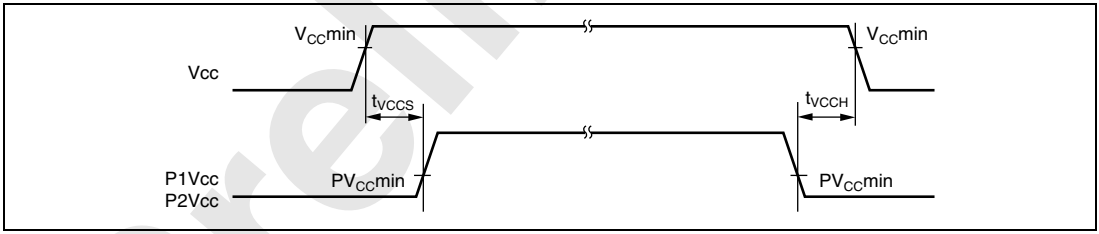


Figure 24.3 Power-On/Off Timing

24.4.2 Clock Timing

Table 24.6 lists the clock timing.

Table 24.6 Clock Timing (1)

Conditions (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Clock cycle time	t_{cyc}	38	125	ns	Figure 24.4
Clock high pulse width	t_{CH}	TBD	—	ns	
Clock low pulse width	t_{CL}	TBD	—	ns	
Clock rise time	t_{Cr}	—	TBD	ns	
Clock fall time	t_{Cf}	—	TBD	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 24.5
Clock oscillator settling time at reset (external clock)		500		µs	
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 24.3
External clock settling delay time	t_{DEXT}	500	—	µs	Figure 24.5
Subclock oscillator settling time	t_{OSC3}	—	2	s	
Subclock oscillator frequency	f_{SUB}		32.768	kHz	
Subclock (ϕ_{SUB}) cycle time	t_{SUB}		30.5	µs	

Note: * The regular specifications are supported in the H8S/2506 Series only.

Table 24.6 Clock Timing (2)

Conditions (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Clock cycle time	t_{cyc}	50	125	ns	Figure 24.4
Clock high pulse width	t_{CH}	TBD	—	ns	
Clock low pulse width	t_{CL}	TBD	—	ns	
Clock rise time	t_{Cr}	—	TBD	ns	
Clock fall time	t_{Cf}	—	TBD	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 24.5
Clock oscillator settling time at reset (external clock)		500		µs	
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 24.4
External clock settling delay time	t_{DEXT}	500	—	µs	Figure 24.5
Subclock oscillator settling time	t_{OSC3}	—	2	s	
Subclock oscillator frequency	f_{SUB}		32.768	kHz	
Subclock (ϕ_{SUB}) cycle time	t_{SUB}		30.5	µs	

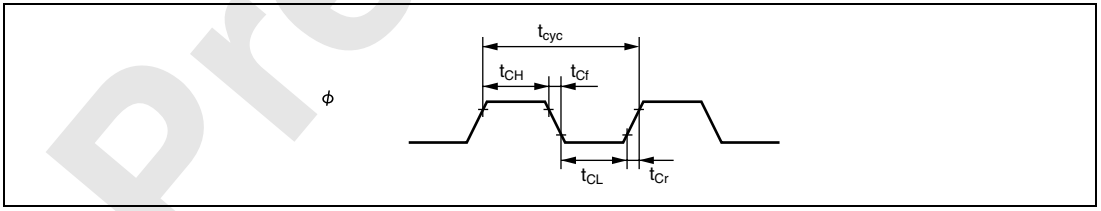


Figure 24.4 System Clock Timing

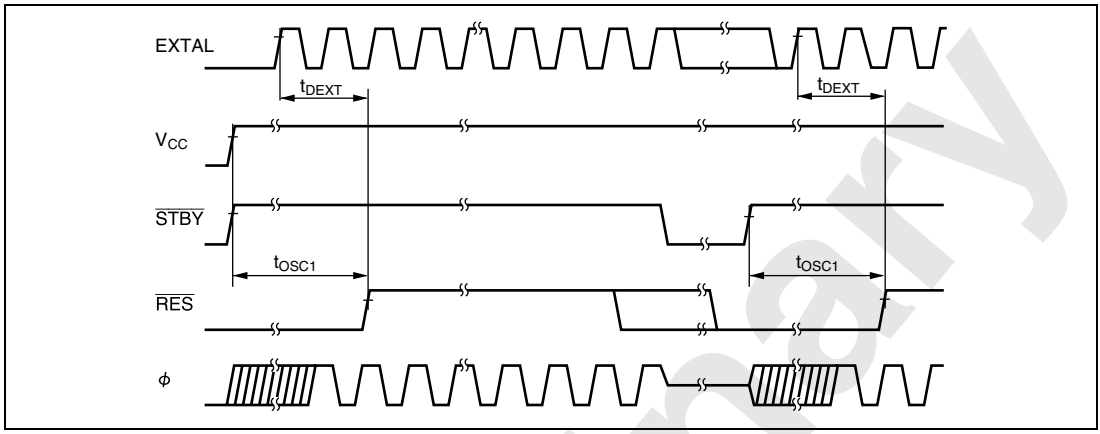


Figure 24.5 Oscillator Settling Timing

24.4.3 Control Signal Timing

Table 24.7 lists the control signal timing.

Table 24.7 Control Signal Timing

Condition A (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, P1V_{CC} = 3.0 V to 5.5 V, P2V_{CC} = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, φ= 32.768kHz,
8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C
(wide-range specifications)

Condition B (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, P1V_{CC} = 3.0 V to 5.5 V, P2V_{CC} = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, φ= 32.768kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	250	—	ns	Figure 24.6
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	250	—	ns	Figure 24.7
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	250	—	ns	

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	Figure 24.7
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

Note: * The regular specifications are supported in the H8S/2506 Series only.

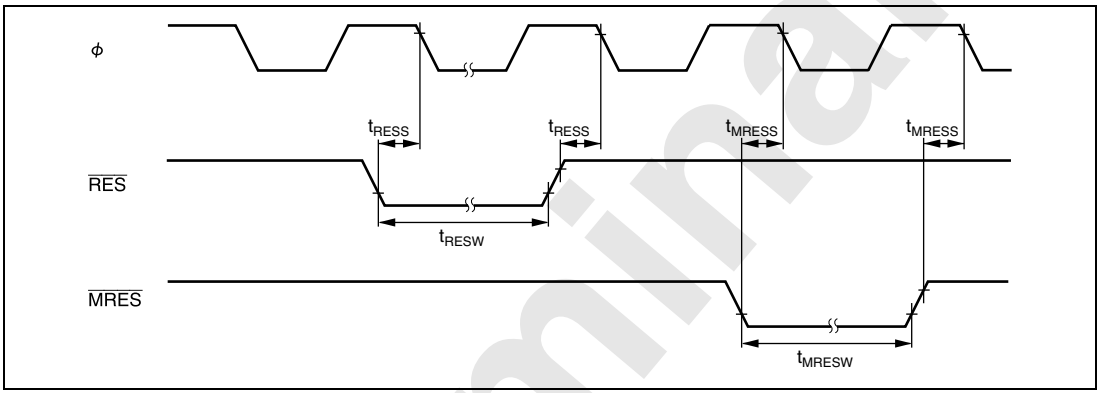


Figure 24.6 Reset Input Timing

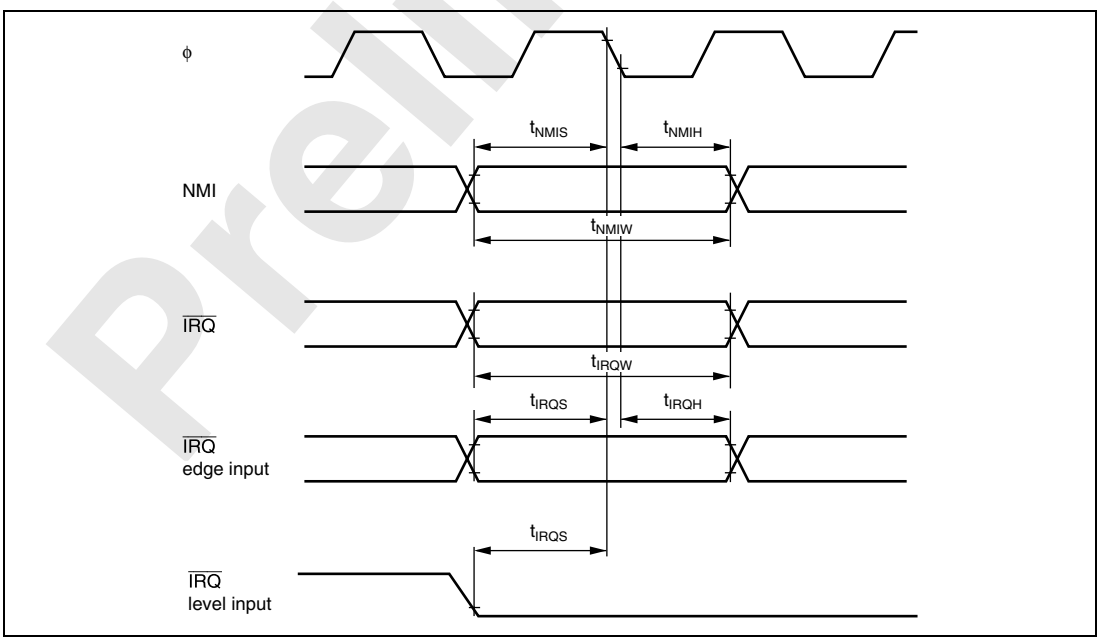


Figure 24.7 Interrupt Input Timing

24.4.4 Bus Timing

Table 24.8 lists the bus timing.

Table 24.8 Bus Timing

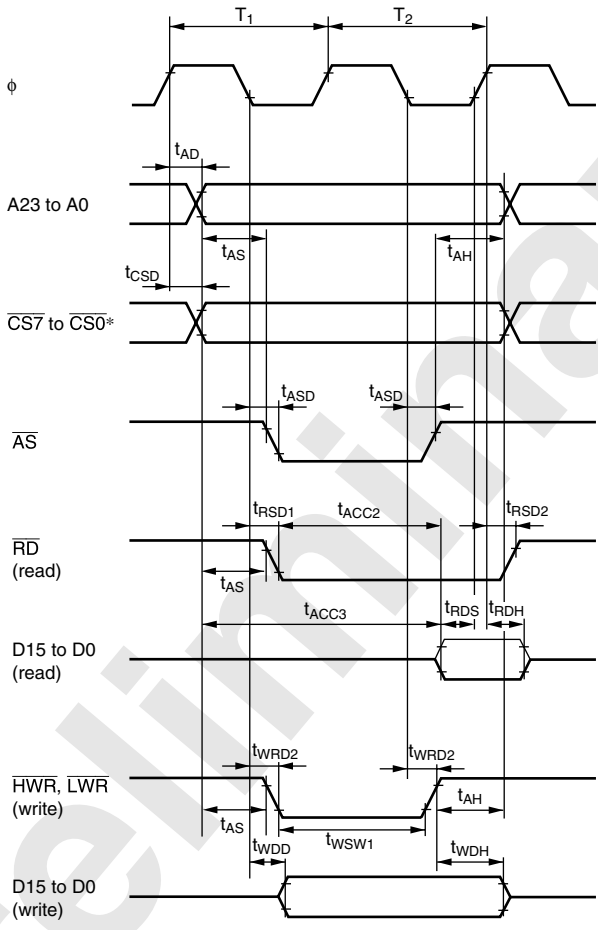
Condition A (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C (wide-range specifications)

Condition B (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			Test
Item	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Conditions**
Address delay time	t_{AD}	—	TBD	—	TBD	ns	Figures 24.8 to 24.12
Address setup time	t_{AS}	$0.5 \times t_{cyc}$ - TBD	—	$0.5 \times t_{cyc}$ - TBD	—	ns	
Address hold time	t_{AH}	$0.5 \times t_{cyc}$ - TBD	—	$0.5 \times t_{cyc}$ - TBD	—	ns	
$\overline{CS}$ delay time	t_{CSD}	—	TBD	—	TBD	ns	
$\overline{AS}$ delay time	t_{ASD}	—	TBD	—	TBD	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	TBD	—	TBD	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	TBD	—	TBD	ns	
Read data setup time	t_{RDS}	TBD	—		—	ns	
Read data hold time	t_{RDH}	TBD	—		—	ns	
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc}$ - TBD	—	$1.0 \times t_{cyc}$ - TBD	ns	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc}$ - TBD	—	$1.5 \times t_{cyc}$ - TBD	ns	
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc}$ - TBD	—	$2.0 \times t_{cyc}$ - TBD	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc}$ - TBD	—	$2.5 \times t_{cyc}$ - TBD	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc}$ - TBD	—	$3.0 \times t_{cyc}$ - TBD	ns	

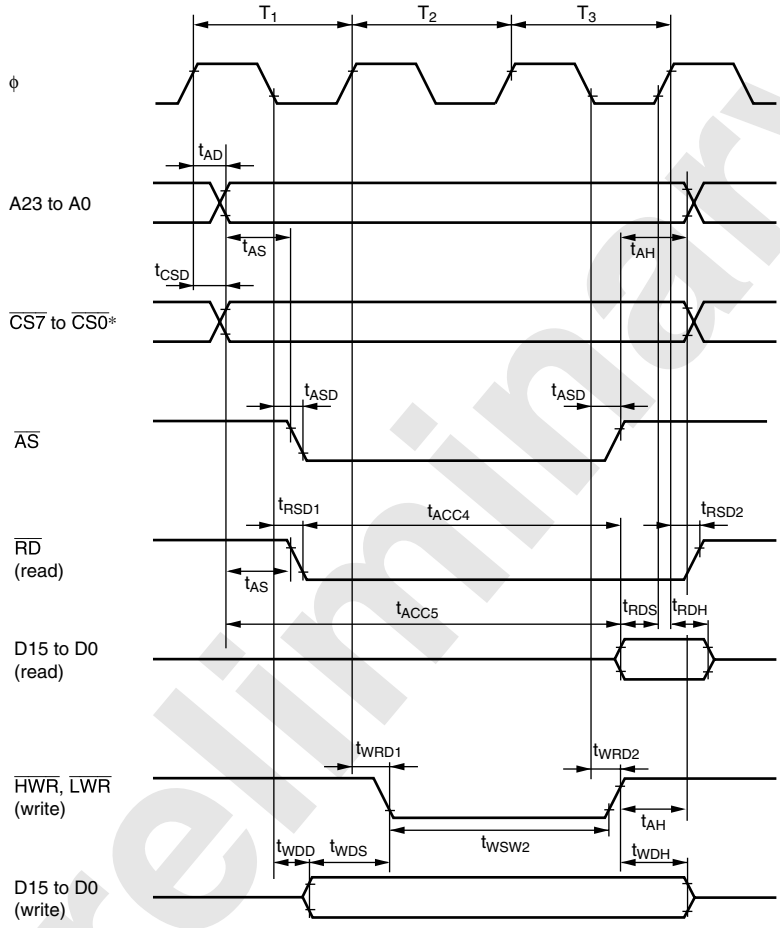
Item	Symbol	Condition A Min.	Condition A Max.	Condition B Min.	Condition B Max.	Unit	Test Conditions
$\overline{WR}$ delay time 1	t_{WRD1}	—	TBD	—	TBD	ns	Figures 24.8 to 24.12
$\overline{WR}$ delay time 2	t_{WRD2}	—	TBD	—	TBD	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc}$-TBD	—	$1.0 \times t_{cyc}$-TBD	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc}$-TBD	—	$1.5 \times t_{cyc}$-TBD	—	ns	
Write data delay time	t_{WDD}	—	TBD	—	TBD	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc}$-TBD	—	$0.5 \times t_{cyc}$-TBD	—	ns	Figure 24.9
Write data hold time	t_{WDH}	$0.5 \times t_{cyc}$-TBD	—	$0.5 \times t_{cyc}$-TBD	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	TBD	—	TBD	—	ns	Figure 24.10
$\overline{WAIT}$ hold time	t_{WTH}	TBD	—	TBD	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	TBD	—	TBD	—	ns	Figure 24.13
$\overline{BACK}$ delay time	t_{BACD}	—	TBD	—	TBD	ns	
Bus floating time	t_{BZD}	—	TBD	—	TBD	ns	

Note: * The regular specifications are supported in the H8S/2506 Series only.



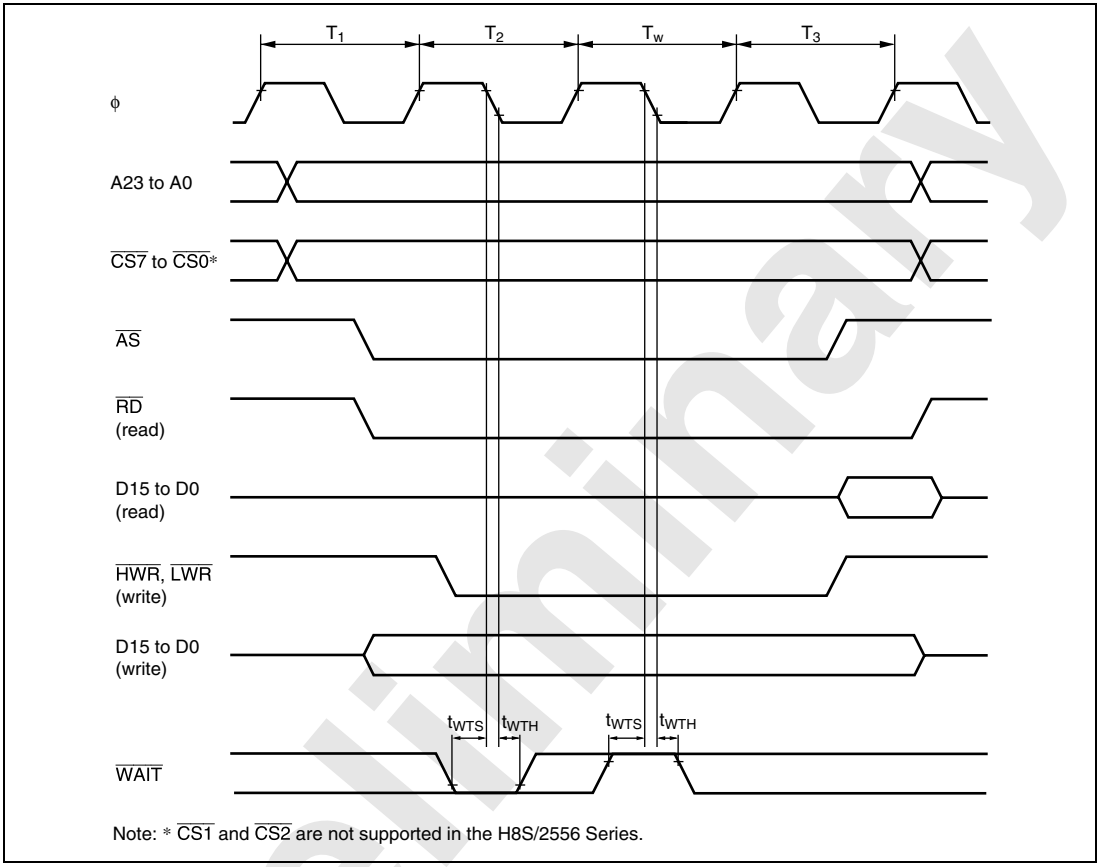
Note: * $\overline{CS1}$ and $\overline{CS2}$ are not supported in the H8S/2556 Series.

Figure 24.8 Basic Bus Timing: Tow-State Access



Note: * $\overline{CS1}$ and $\overline{CS2}$ are not supported in the H8S/2556 Series.

Figure 24.9 Basic Bus Timing: Three-State Access



Note: * CS1 and CS2 are not supported in the H8S/2556 Series.

Figure 24.10 Basic Bus Timing: Three-State Access, One Wait

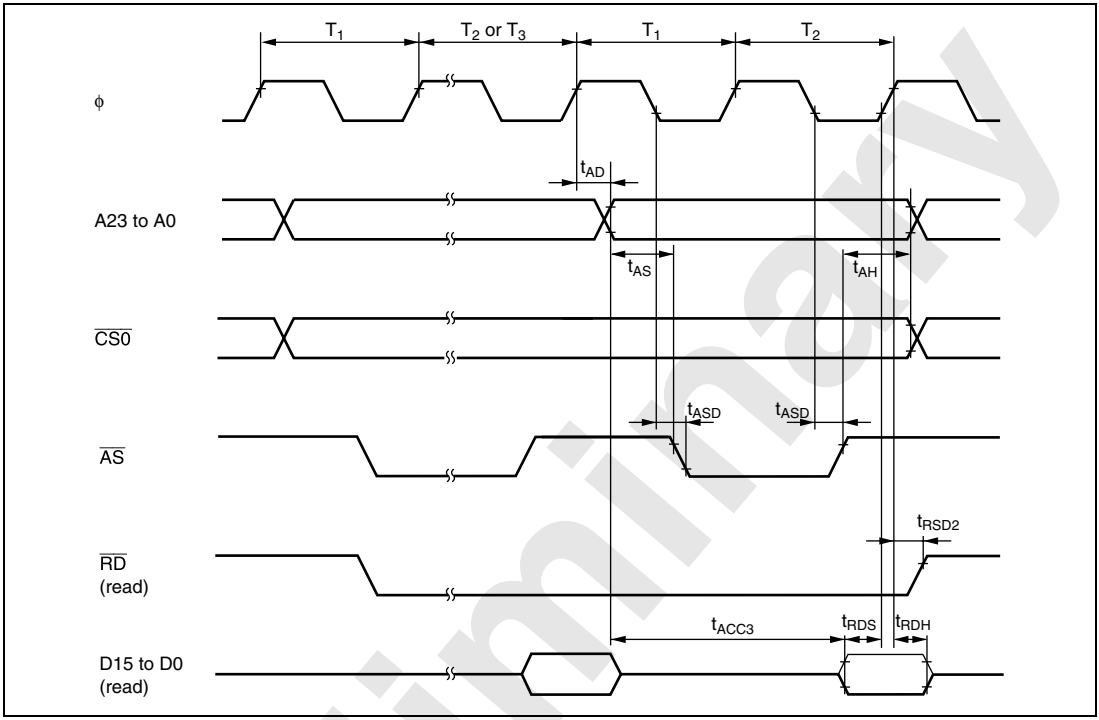


Figure 24.11 Burst ROM Access Timing: Two-State Access

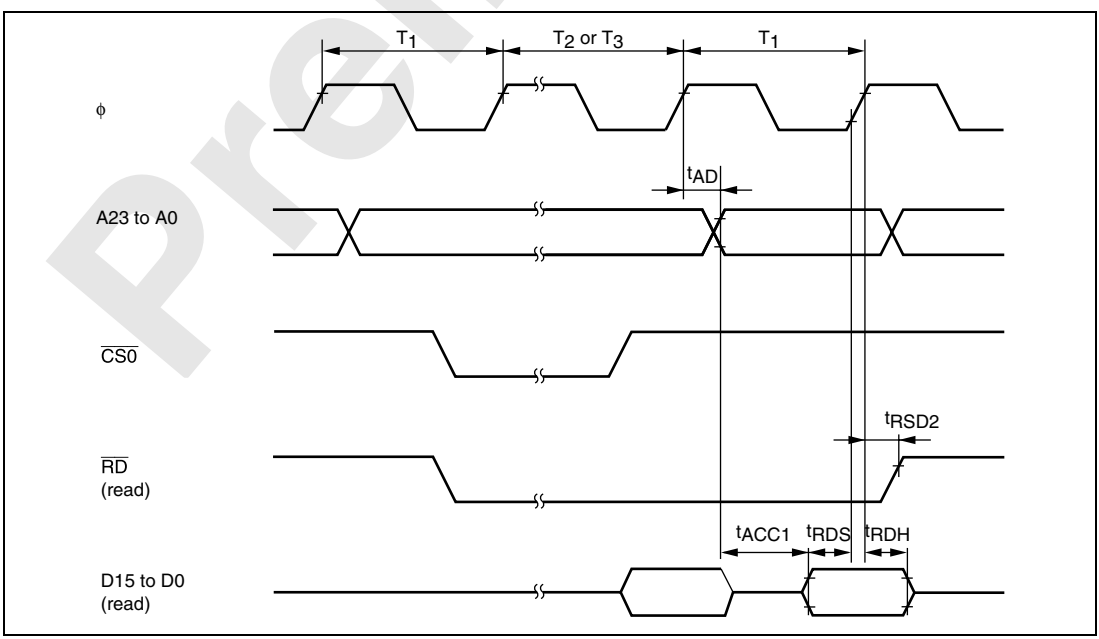
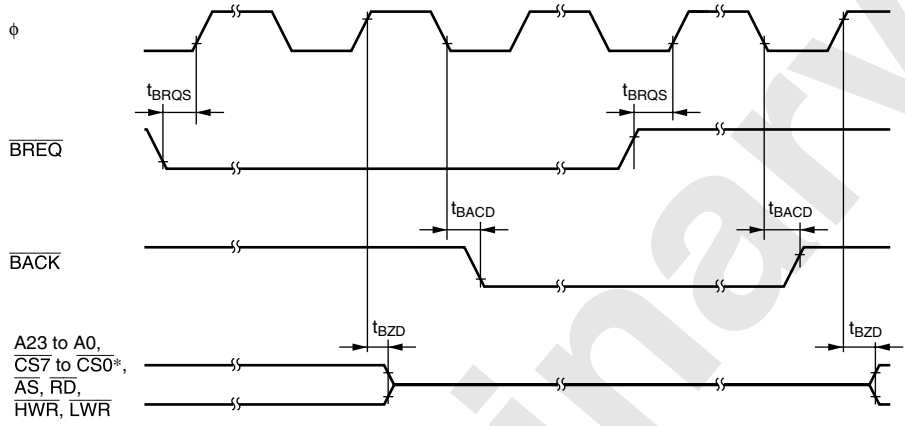


Figure 24.12 Burst ROM Access Timing: One-State Access



Note: * CS1 and CS2 are not supported in the H8S/2556 Series.

Figure 24.13 External Bus-Released Timing

24.4.5 Timing of On-Chip Peripheral Modules

Table 24.9 lists the timing of on-chip peripheral modules. Table 24.10 lists the I²C2 bus timing.

Table 24.9 Timing of On-Chip Peripheral Modules

Condition A (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, P1V_{CC} = 3.0 V to 5.5 V, P2V_{CC} = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, φ= 32.768kHz,
8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*[1], T_a = –40°C to +85°C (wide-range specifications)

Condition B (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, P1V_{CC} = 3.0 V to 5.5 V, P2V_{CC} = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, φ= 32.768kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

			Condition A		Condition B			Test
Item		**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Conditions**
I/O ports	Output data delay time	t_{PWD}	—	TBD	—	TBD	ns	Figure 24.14
	Input data setup time	t_{PRS}	TBD	—	TBD	—		
	Input data hold time	t_{PRH}	TBD	—	TBD	—		
TPU	Timer output delay time	t_{TOCD}	—	TBD	—	TBD	ns	Figure 24.15
	Timer input setup time	t_{TICS}	TBD	—	TBD	—		

Item			Symbol	Condition A Min.	Condition A Max.	Condition B Min.	Condition B Max.	Unit	Test Conditions
TPU	Timer clock input setup time		t_{TCKS}	TBD	—	TBD	—	ns	Figure 24.16
	Timer clock pulse width	Single edge	t_{TCKWH}	TBD	—	—		t_{cyc}	
		Both edges	t_{TCKWL}	TBD	—	—			
TMR	Timer output delay time		t_{TMOD}	—	TBD	—	TBD	ns	Figure 24.17
	Timer reset input setup time		t_{TMRS}	TBD	—	TBD	—	ns	Figure 24.19
	Timer clock input setup time		t_{TMCS}	TBD	—	TBD	—	ns	Figure 24.18
	Timer clock pulse width	Single edge	t_{TMCWH}	TBD	—	—		t_{cyc}	
		Both edges	t_{TMCWL}	TBD	—	—			
WDT_1	BUZZ output delay time		t_{BUZD}	—	TBD	—	TBD	ns	Figure 24.20
SCI	Input clock cycle	Asynchronous	t_{Scyc}	TBD	—	—		t_{cyc}	Figure 24.21
		Synchronous		TBD	—	—			
	Input clock pulse width		t_{SCKW}	TBD	TBD	TBD	TBD	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	TBD	—	TBD	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	TBD	—	TBD		
	Transmit data delay time		t_{TXD}	—	TBD	—	TBD	ns	Figure 24.22
	Receive data setup time (synchronous)		t_{RXS}	TBD	—	TBD	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	TBD	—	TBD	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	TBD	—	TBD	—	ns	Figure 24.23
HCAN*2	Transmit data delay time		t_{HTXD}	—	TBD	—	TBD	ns	Figure 24.24
	Transmit data setup time		t_{HRXS}	TBD	—	TBD	—	ns	
	Transmit data hold time		t_{HRXH}	TBD	—	TBD	—	ns	

Notes: 1. The regular specifications are supported in the H8S/2506 Series only.

2. (For H8S/2556 Series) The HCAN input signal is asynchronous, but checked as if it has been changed at the ϕ clock rise (two clock intervals) shown in figure 24.24. The HCAN output signal is asynchronous, but it changes at the ϕ clock rise (two clock intervals) shown in figure 24.24.

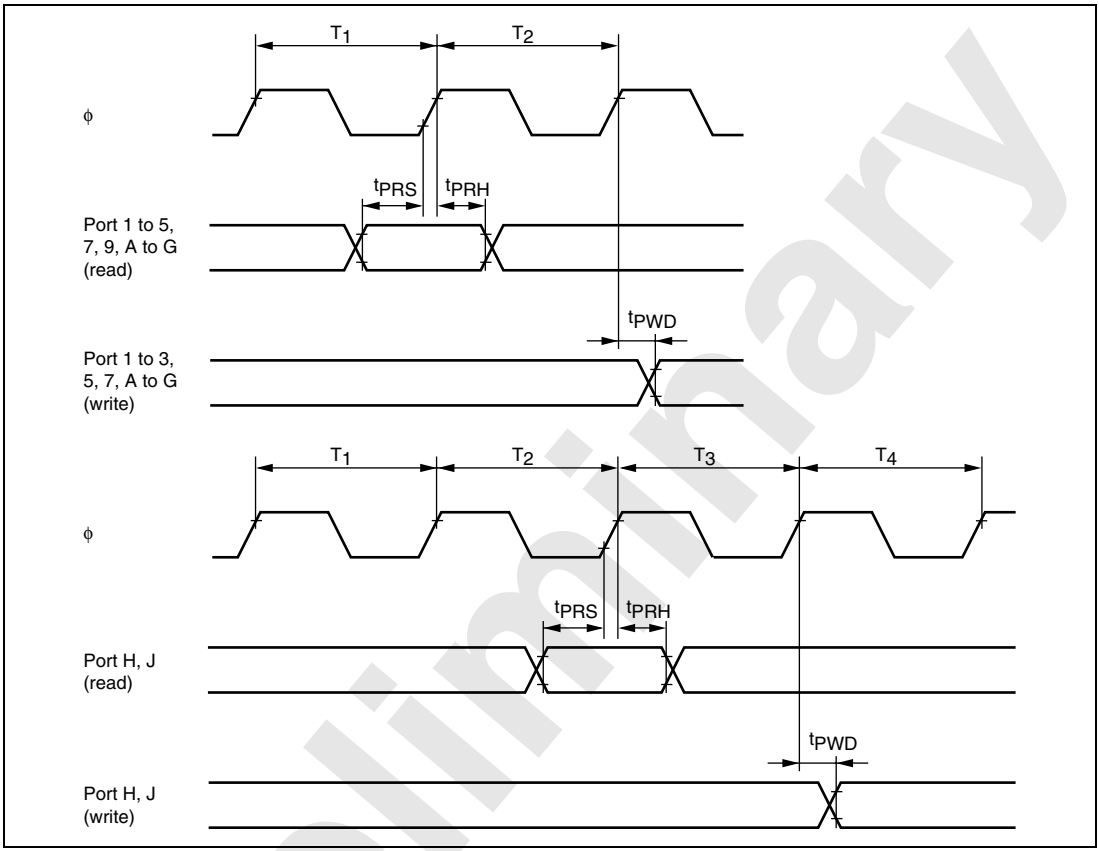


Figure 24.14 I/O Port Input/Output Timing

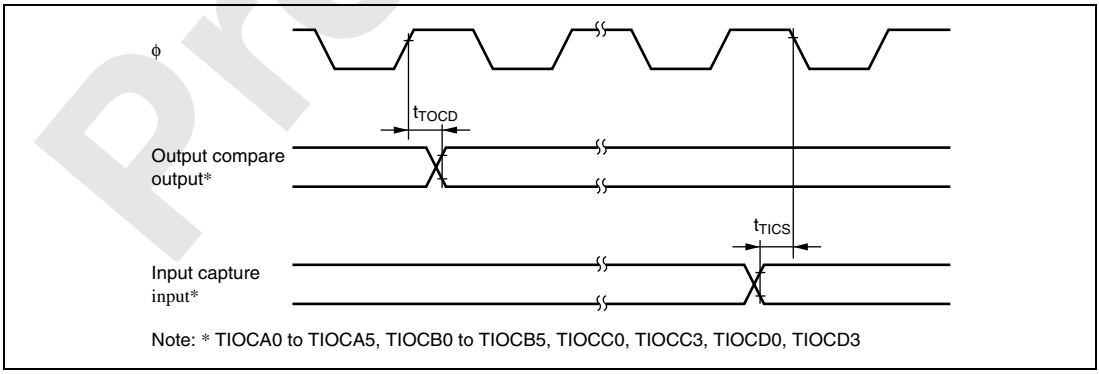


Figure 24.15 TPU Input/Output Timing

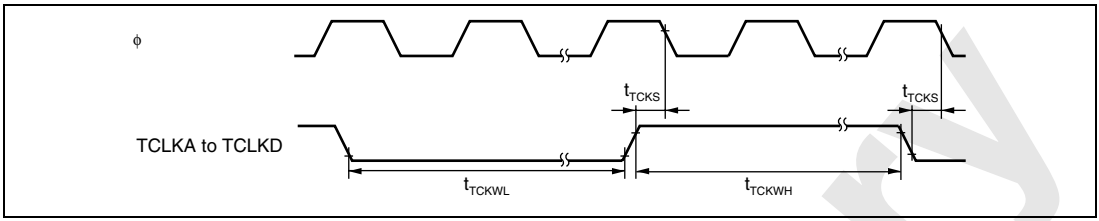


Figure 24.16 TPU Clock Input Timing

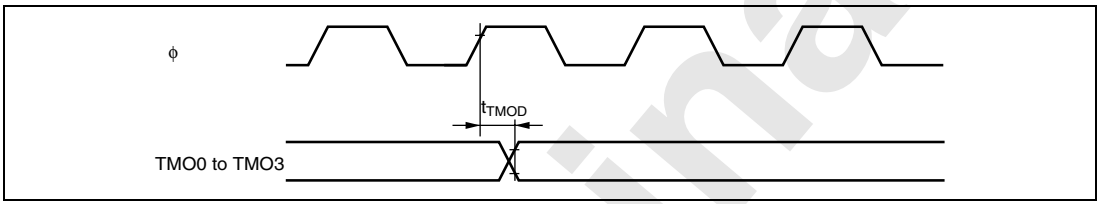


Figure 24.17 8-Bit Timer Output Timing

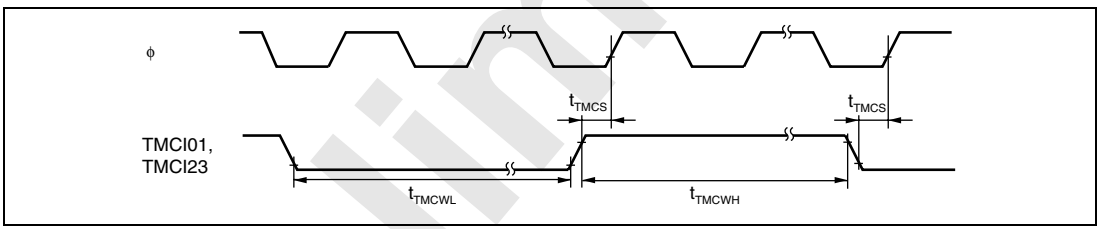


Figure 24.18 8-Bit Timer Clock Input Timing

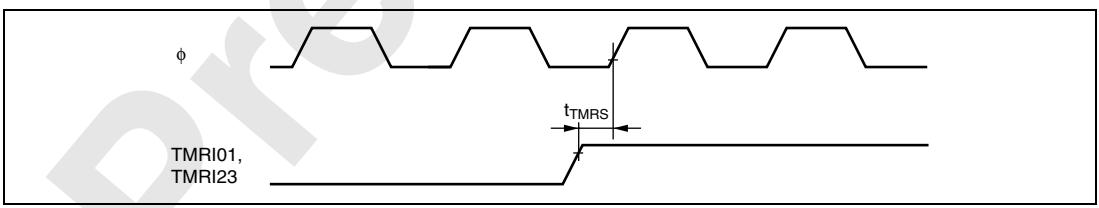


Figure 24.19 8-Bit Timer Reset Input Timing

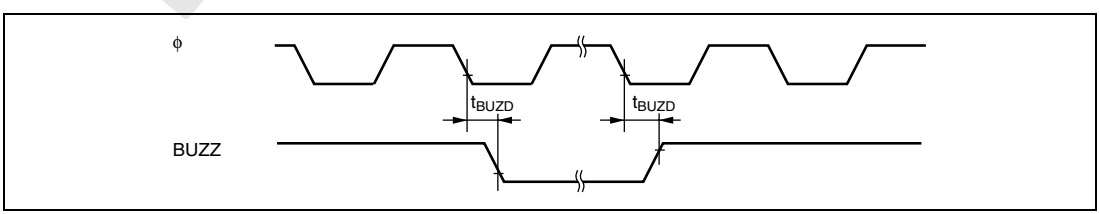


Figure 24.20 WDT_1 Output Timing

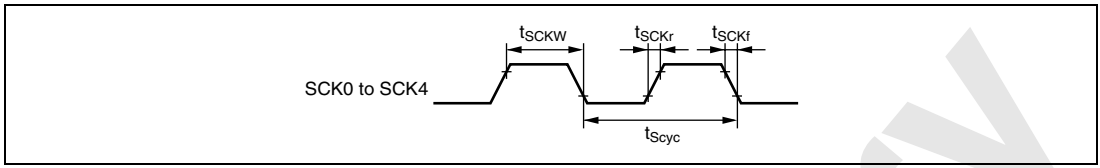


Figure 24.21 SCK Clock Input Timing

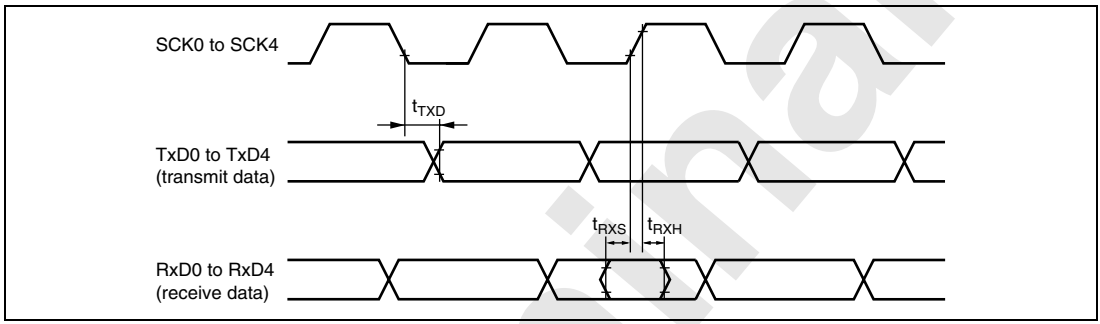


Figure 24.22 SCI Input/Output Timing/Synchronous Mode

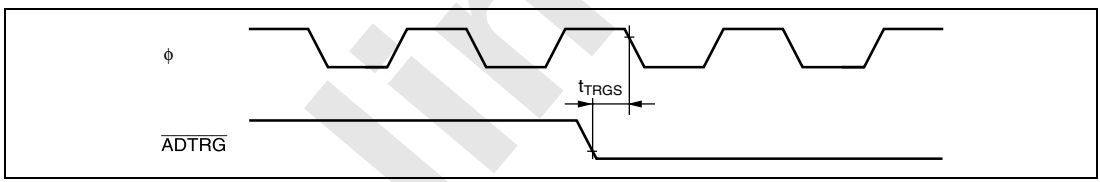


Figure 24.23 External Trigger Input Timing for A/D Converter

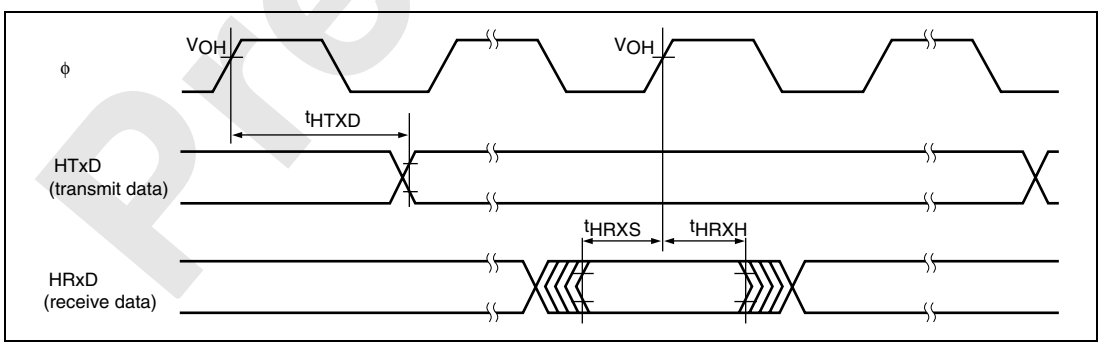
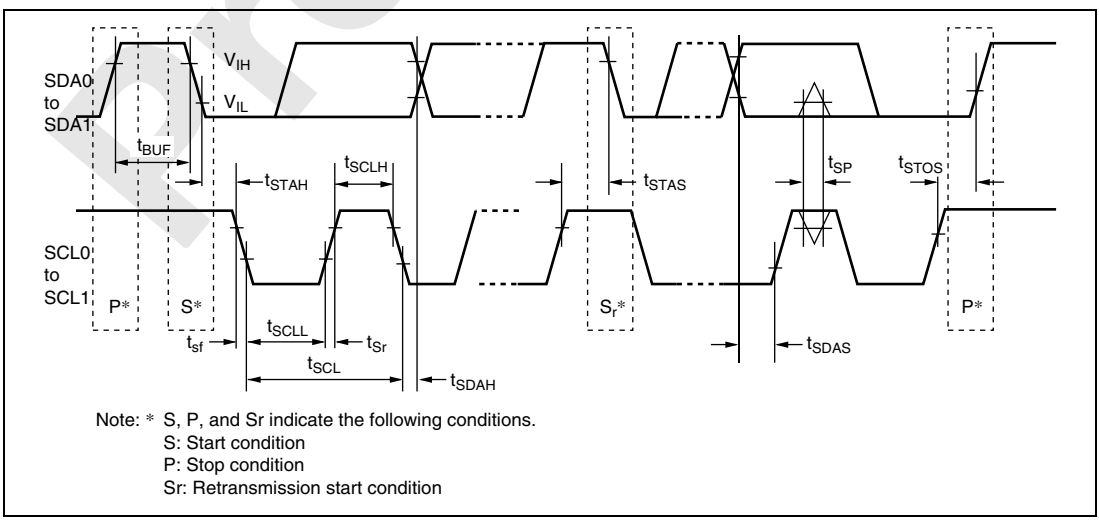


Figure 24.24 HCAN Input/Output Timing

Table 24.10 I²C Bus Interface 2 Timing

Conditions: V_{CC} = 3.0 V to 5.5 V, P1V_{CC} = 3.0 V to 5.5 V, P2V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0 V, ϕ = 8 MHz to maximum operating frequency, T_a = –20°C to +75°C

Item	Symbol	Test Conditions	Standard Value Min.	Typ.	Max.	Unit	Test Conditions
SCL input cycle time	t_{SCL}		$12t_{cyc}$ + 600	—	—	ns	Figure 24.25
SCL input high pulse width	t_{SCLH}		$3t_{cyc}$ + 300	—	—	ns	
SCL input low pulse width	t_{SCLL}		$5t_{cyc}$ + 300	—	—	ns	
SCL, SDA input fall time	t_{Sf}		—	—	300	ns	
SCL, SDA input spike pulse elimination time	t_{SP}		—	—	$1t_{cyc}$	ns	
SDA input bus free time	t_{BUF}		$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}		$3t_{cyc}$	—	—	ns	
Retransmission start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}		$3t_{cyc}$	—	—	ns	
Data input setup time	t_{SDAS}		$1t_{cyc}$ + 20	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
SCL, SDA load capacitance	C_b		0	—	400	pF	
SCL, SDA output fall time	t_{Sf}	V_{CC} = 4.0 to 5.5 V	—	—	250	ns	
			—	—	300	ns	



Note: * S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 24.25 I²C Bus Interface 2 Input/Output Timing

24.5 A/D Conversion Characteristics

Table 24.11 lists the A/D conversion characteristics.

Table 24.11 A/D Conversion Characteristics

Condition A (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz,
8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C
(wide-range specifications)

Condition B (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	—	—	9.9	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Non-linearity error	—	—	±6.0	LSB
Offset error	—	—	±4.0	LSB
Full-scale error	—	—	±4.0	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	LSB

Note: * The regular specifications are supported in the H8S/2506 Series only.

24.6 D/A Conversion Characteristics

Table 24.12 lists the D/A conversion characteristics.

Table 24.12 D/A Conversion Characteristics

Condition A (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz,
8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C
(wide-range specifications)

Condition B (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V,
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	Load capacitance: 20 pF
Absolute accuracy	—	±2.0	±3.0	LSB	Load resistance: 2 MΩ
	—	—	±2.0	LSB	Load resistance: 4 MΩ

Note: * The regular specifications are supported in the H8S/2506 Series only.

24.7 Flash Memory Characteristics

Table 24.13 shows the flash memory characteristics.

Table 24.13 Flash Memory Characteristics

Condition A (for H8S/2552 Series, H8S/2506 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C (wide-range specifications)

Condition B (for H8S/2556 Series):
V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768kHz, 8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Programming time*[1]*[2]*[4]	t_p	—	3	30	ms/128 bytes	
Erase time*[1]*[3]*[5]	t_E	—	80	800	ms/4 kbytes	
		—	500	5000	ms/32 kbytes	
		—	1000	10000	ms/64 kbytes	
Programming time (total) *[1]*[2]*[4]	$\sum_{IP}$	—	10	30	s/512 kbytes	T_a = 25°C, all 0
Erase time (total) *[1]*[2]*[4]	$\sum_{IE}$	—	10	30	s/512 kbytes	T_a = 25°C
Programming/Erase time (total) *[1]*[2]*[4]	$\sum_{IPE}$	—	20	60	s/512 kbytes	T_a = 25°C
Count of reprogramming	N_{WEC}	100*[3]	—	—	Times	
Data hold time*[4]	t_{DRP}	10	—	—	Year	

Notes: 1. Programming/Erase time depends on the data.
2. Programming/Erase time does not include the data transfer time.
3. The minimum times that all characteristics after reprogramming are guaranteed. (The range between 1 and a minimum value is guaranteed.)
4. Data hold characteristics are when reprogramming is performed within the range of specifications including a minimum value.

Appendix A I/O Port States in Each Pin State

Port Name Pin Name	MCU Operating Mode	Power-on Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode, Watch Mode	Bus Release State	Program Execution State, Sleep Mode
Port 1	6, 7	T	keep	T	keep	keep	I/O port
Port 2	6, 7	T	keep	T	keep	keep	I/O port
Port 3	6, 7	T	keep	T	keep	keep	I/O port
Port 4	6, 7	T	T	T	T	T	Input port
Port 5	6, 7	T	keep	T	keep	keep	I/O port
P77 to P74	6, 7	T	keep	T	keep	keep	I/O port
P73/TMO1/$\overline{CS7}$	7	T	keep	T	keep	keep	I/O port
P72/TMO0/$\overline{CS6}$	6	T	keep	T	[DDR•OPE = 0]	T	[DDR = 0]
P71/TMRI23/ TMCI23/$\overline{CS5}$					T		Input port
					[DDR•OPE = 1]		[DDR = 1]
P70/TMRI01/ TMCI01/$\overline{CS4}$					H		$\overline{CS7}$ to $\overline{CS4}$
P97/DA1	6, 7	T	T	T	[DAOEn = 1]	keep	Input port
P96/DA0					keep		
					[DAOEn = 0]		
					T		
P95 to P90	6, 7	T	T	T	T	T	Input port
Port A	7	T	Keep	T	keep	keep	I/O port
Address output selection with AEn bit	6	T	Keep	T	[OPE = 0]	T	Address output
					T		
					[OPE = 1]		
					keep		
Port selection	6	T	keep	T	keep	keep	I/O port
Port B	7	T	keep	T	keep	keep	I/O port
Address output selection with AEn bit	6	T	Keep	T	[OPE = 0]	T	Address output
					T		
					[OPE = 1]		
					keep		
Port selection	6	T	keep	T	keep	keep	I/O port

Port Name Pin Name		MCU Operating Mode	Power-on Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode, Watch Mode	Bus Release State	Program Execution State, Sleep Mode
Port C		6	T	keep	T	[DDR•OPE = 0] T [DDR•OPE = 1] keep	T	[DDR = 0] Input port [DDR = 1] Address output
		7	T	keep	T	keep	keep	I/O port
Port D		6	T	T	T	T	T	Data bus
		7	T	keep	T	keep	keep	I/O port
Port E	8-bit bus	6	T	keep	T	keep	keep	I/O port
	16-bit bus	6	T	T	T	T	T	Data bus
		7	T	keep	T	keep	keep	I/O port
PF7/ϕ		6	Clock output*3	[DDR = 0] Input port [DDR = 1] Clock output	T	[DDR = 0] Input port [DDR = 1] H	[DDR = 0] Input port [DDR = 1] Clock output	[DDR = 0] Input port [DDR = 1] Clock output
		7	T	keep	T	[DDR = 0] Input port [DDR = 1] H	[DDR = 0] Input port [DDR = 1] Clock output	[DDR = 0] Input port [DDR = 1] Clock output
PF6/$\overline{AS}$ PF5/$\overline{RD}$ PF4/ $\overline{HWR}$		6	H*3	H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{AS}$•$\overline{RD}$•$\overline{HWR}$
		7	T	keep	T	keep	keep	I/O port
PF3/$\overline{LWR}$/ $\overline{ADTRG}$/$\overline{IRQ3}$		7	T	keep	T	keep	keep	I/O port
	8-bit bus	6	T	keep	T	keep	keep	I/O port
	16-bit bus	6		H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{LWR}$

Port Name Pin Name	MCU Operating Mode	Power-on Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode, Watch Mode	Bus Release State	Program Execution State, Sleep Mode
PF2/$\overline{\text{WAIT}}$	6	T	keep	T	[WAITE = 0]	[WAITE = 0]	[WAITE = 0]
					keep	keep	I/O port
					[WAITE = 1]	[WAITE = 1]	[WAITE = 1]
					T	T	$\overline{\text{WAIT}}$
	7	T	keep	T	keep	keep	I/O port
PF1/$\overline{\text{BACK}}$/BUZZ	6	T	keep	T	[BRLE = 0]	L	[BRLE = 0]
					keep		I/O port
					[BRLE = 1]		[BRLE = 1]
					H		$\overline{\text{BACK}}$
	7	T	keep	T	keep	keep	I/O port
PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	6	T	keep	T	[BRLE = 0]	T	[BRLE = 0]
					keep		I/O port
					[BRLE = 1]		[BRLE = 1]
					T		$\overline{\text{BREQ}}$
	7	T	keep	T	keep	keep	I/O port
PG4/$\overline{\text{CS0}}$	6	T	keep	T	[DDR•OPE = 0]	T	[DDR = 0]
					T		Input port
					[DDR•OPE = 1]		[DDR = 1]
					H		$\overline{\text{CS0}}$ (H in sleep mode)
	7	T	keep	T	keep	keep	I/O port
PG3/$\overline{\text{RX}}$/$\overline{\text{CS1}}$*[1]	6	T	keep	T	[DDR • OPE = 0]	T	[DDR = 0]
PG2/$\overline{\text{TX}}$/$\overline{\text{CS2}}$*[1]					T		Input port
PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$					[DDR • OPE = 1]		[DDR = 1]
					H		$\overline{\text{CS1}}$ to $\overline{\text{CS3}}$
	7	T	keep	T	keep	keep	I/O port
PG0/$\overline{\text{IRQ6}}$	6, 7	T	keep	T	keep	keep	I/O port
Port H	6, 7	T	keep	T	keep	keep	I/O port
Port J	6, 7	T	keep	T	keep	keep	I/O port
THxD*[2]	6, 7	H*[3]	H	T	H	keep	THxD Output

Port Name Pin Name	MCU Operating Mode	Power-on Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode, Watch Mode	Bus Release State	Program Execution State, Sleep Mode
HRxD*[2]	6, 7	Input	Input	T	T	Input	HRxD Input

Legend

H: High level

L: Low level

T: High-impedance

Keep: Input port becomes high-impedance, output port retains state

DDR: Data direction register

OPE: Output port enable

WAITE: Wait input enable

BRLE: Bus release enable

Notes: 1. PG3 and PG2 are not supported in the H8S/2556 Series.
PG3/$\overline{RX}$/$\overline{CS1}$ and PG2/$\overline{TX}$/$\overline{CS2}$ are supported in the H8S/2552 Series. (When IEE bit is 1, $\overline{RX}$ and $\overline{TX}$ are valid.)
PG3/$\overline{CS1}$ and PG2/$\overline{CS2}$ are supported in the H8S/2506 Series.

2. Supported only in the H8S/2556 Series.

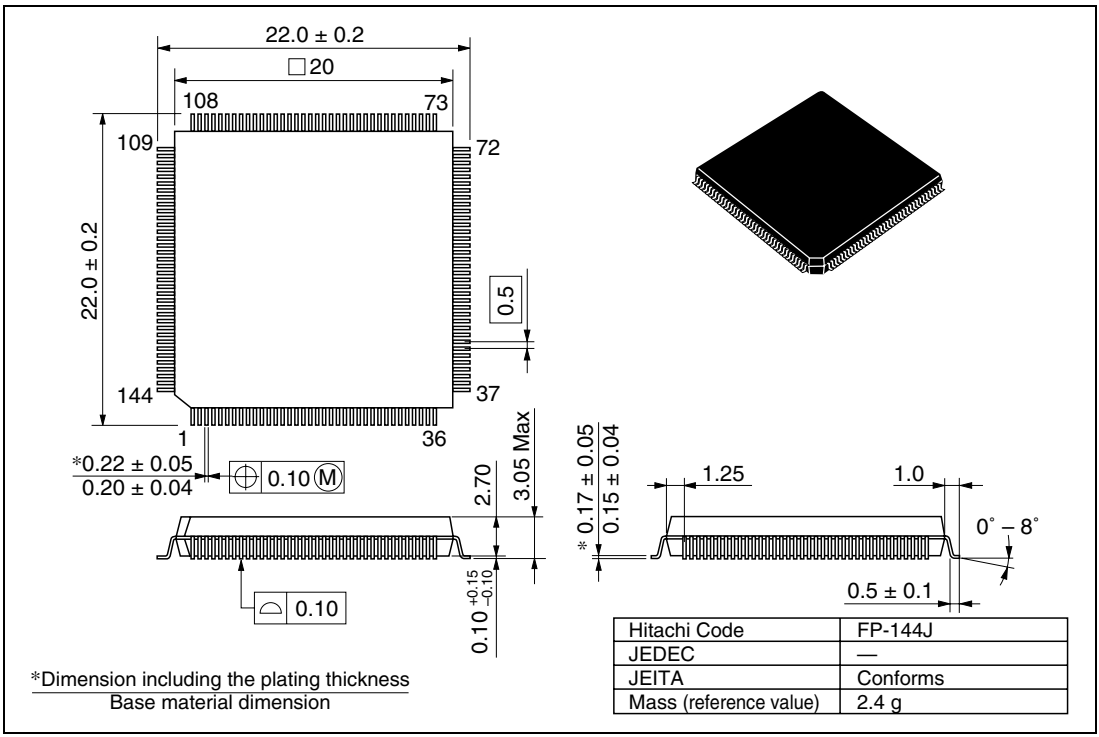
3. Output pins are in the high-impedance state when the power is supplied.

Appendix B Product Codes

Product Type		Product Code	Mark Code	Package (Hitachi Package Code)
H8S/2556	Flash memory version	HD64F2556	HD64F2556FC20	144-pin QFP (FP-144J)
H8S/2552	Flash memory version	HD64F2552	HD64F2552FC26	144-pin QFP (FP-144J)
H8S/2506	Flash memory version	HD64F2506	HD64F2506FC26	144-pin QFP (FP-144J)

Note: Some products above are in the developing or planning stage. Please contact Hitachi agency to confirm the present state of each product.

Appendix C Package Dimensions



Hitachi Code	FP-144J
JEDEC	—
JEITA	Conforms
Mass (reference value)	2.4 g

Figure C.1 FP-144J Package Dimensions

Index

H8S/2556 Series, H8S/2552 Series, H8S/2506 Series
Hardware Manual

Publication Date: 1st Edition, March 2003
Published by: Business Operation Division
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.