REJ09B0281-0300

Everywhere you imagine. RENESAS

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

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

16

H8S/2668 Group, H8S/2667 F-ZTAT™ Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer H8S Family/H8S/2600 Series

H8S/2667 HD64F2667

Hardware Manual

Rev. 3.00
Revision Date: Feb 22, 2006

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are 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. Main Revisions in This Edition

   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.
5. Contents
6. Overview
7. 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.

8. List of Registers
9. Electrical Characteristics
10. Appendix
11. Index

Preface

The H8S/2668 Group are microcomputers (MCU) made up of the H8S/2600 CPU employing Renesas' original architecture as their cores, and the peripheral functions required to configure a system.

The H8S/2600 CPU has an internal 32-bit configuration, sixteen 16-bit general registers, and a simple and optimized instruction set for high-speed operation. The H8S/2600 CPU can handle a 16-Mbyte linear address space.

This LSI is equipped with data transfer controller (DTC) bus masters, ROM and RAM memory, a 16-bit timer pulse unit (TPU), a programmable pulse generator (PPG), an 8-bit timer (TMR), a watchdog timer (WDT), a serial communication interface (SCI and IrDA), a 10-bit A/D converter, an 8-bit D/A converter, and I/O ports as on-chip peripheral modules required for system configuration.

A high functionality bus controller is also provided, enabling fast and easy connection of DRAM, SDRAM and other kinds of memory.

A single-power flash memory (F-ZTAT™*) version and masked ROM version are 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.

This manual describes this LSI's hardware.

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

Target Users: This manual was written for users who will be using this LSI 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 this LSI to the target users.
Refer to the H8S/2600 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 20, 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.renesas.com

H8S/2668 Group manuals:

Document Title	Document No.
H8S/2668 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	REJ09B139

User's manuals for development tools:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0058
H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-037
H8S, H8/300 Series High-performance Embedded Workshop, High-performance Debugging Interface Tutorial	ADE-702-231
High-performance Embedded Workshop User's Manual	ADE-702-201

Main Revisions in This Edition

Item	Page	Revision (See Manual for Details)
All	—	All references to Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names changed to Renesas Technology Corp. Designation for categories changed from “series” to “group”
5.6.5 DTC Activation by Interrupt Figure 5.6 DTC and Interrupt Controller	95	Figure title amended
7.8.3 DTCE Bit Setting	171	Description of “DMAC Transfer End Interrupt” deleted
Section 8 I/O Port Table 8.1 Port Functions	174	Table 8.1 amended

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 1	General I/O port also functioning as PPG outputs, and TPU I/Os	P17/PO15/TIOCB2/TCLKD P16/PO14/TIOCA2 P15/PO13/TIOCB1/TCLKC P14/PO12/TIOCA1 P13/PO11/TIOCD0/TCLKB P12/PO10/TIOCC0/TCLKA P11/PO9/TIOCB0 P10/PO8/TIOCA0					Schmitt-triggered input
Port 2	General I/O port also functioning as PPG outputs, TPU I/Os, and interrupt inputs	P27/PO7/TIOCB5 P26/PO6/TIOCA5 P25/PO5/TIOCB4 P24/PO4/TIOCA4 P23/PO3/TIOCD3 P22/PO2/TIOCC3 P21/PO1/TIOCB3 P20/PO0/TIOCA3					Schmitt-triggered input
Port 3	General I/O port also functioning as SCI I/Os	P35/SCK1 P34/SCK0 P33/RxD1 P32/RxD0/IrRxD P31/TxD1 P30/TxD0/IrTxD					Open-drain output enable

Item	Page	Revision (See Manual for Details)
Section 8 I/O Port Table 8.1 Port Functions	175	Table 8.1 amended

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 5	General I/O port also functioning as interrupt inputs, A/D converter analog inputs, and D/A converter analog outputs	P57/AN15/DA3/$\overline{IRQ7}$ P56/AN14/DA2/$\overline{IRQ6}$ P55/AN13/$\overline{IRQ5}$ P54/AN12/$\overline{IRQ4}$					Schmitt-triggered input when used as IRQ input
	General I/O port also functioning as interrupt inputs, A/D converter analog inputs, and SCI I/Os	P53/$\overline{ADTRG}$/$\overline{IRQ3}$ P52/SCK2/$\overline{IRQ2}$ P51/RxD2/$\overline{IRQ1}$ P50/TxD2/$\overline{IRQ0}$					Schmitt-triggered input when used as IRQ input
Port 6	General I/O port also functioning as interrupt inputs, and TMR I/Os	P65/TMO1 P64/TMO0 P63/TMCI1 P62/TMCI0 P61/TMRI1 P60/TMRI0					
Port 7	General I/O port	P75 P74 P73 P72 P71 P70			P75 P74 P73 P72 P71 P70	P75 P74 P73 P72 P71 P70	
Port 8	General I/O port as interrupt inputs	P85/$\overline{IRQ5}$ P84/$\overline{IRQ4}$ P83/$\overline{IRQ3}$ P82/$\overline{IRQ2}$ P81/$\overline{IRQ1}$ P80/$\overline{IRQ0}$			P85/$\overline{IRQ5}$ P84/$\overline{IRQ4}$ P83/$\overline{IRQ3}$ P82/$\overline{IRQ2}$ P81/$\overline{IRQ1}$ P80/$\overline{IRQ0}$	P85/$\overline{IRQ5}$ P84/$\overline{IRQ4}$ P83/$\overline{IRQ3}$ P82/$\overline{IRQ2}$ P81/$\overline{IRQ1}$ P80/$\overline{IRQ0}$	Schmitt-triggered input when used as IRQ input

176

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port A	General I/O port also functioning as address outputs	PA7/A23 PA6/A22 PA5/A21 A20 A19 A18 A17 A16		PA7/A23 PA6/A22 PA5/A21 PA4/A20 PA3/A19 PA2/A18 PA1/A17 PA0/A16	PA7/A23 PA6/A22 PA5/A21 PA4/A20 PA3/A19 PA2/A18 PA1/A17 PA0/A16	PA7 PA6 PA5 PA4 PA3 PA2 PA1 PA0	Built-in input pull-up MOS Open-drain output enable

177

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port H	General I/O port also functioning as interrupt inputs and bus control I/Os	PH3/$\overline{CS7}$/($\overline{IRQ7}$) PH2/$\overline{CS6}$/($\overline{IRQ6}$) PH1/$\overline{CS5}$ PH0/$\overline{CS4}$			PH3/$\overline{CS7}$/($\overline{IRQ7}$) PH2/$\overline{CS6}$/($\overline{IRQ6}$) PH1/$\overline{CS5}$ PH0/$\overline{CS4}$	PH3/($\overline{IRQ7}$) PH2/($\overline{IRQ6}$) PH1 PH0	Schmitt-triggered input when used as IRQ input

Item	Page	Revision (See Manual for Details)
9.3.9 Timer Synchronous Register (TSYR)	288	Bit 7 and 6 initial value amended (Before) — → (After) 0
13.3.7 Serial Status Register (SSR)	406	Normal Serial Communication Interface Mode (When SMIF in SCMR is 0) Bit 2 Clearing condition amended • When the DTC is activated by a TXI interrupt and writes data to TDR
13.3.9 Bit Rate Register (BRR) Table13.2 Relationships between N Setting in BRR and Bit Rate B	411	Table 13.2 amended **Mode** / **Bit Rate** / **Error** Smart Card Interface Mode $B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N + 1)}$ $\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} - 1 \right\} \times 100$
14.3.2 A/D Control/Status Register (ADCSR)	477	Bit 7 Clearing condition amended • When the DTC is activated by an ADI interrupt and ADDR is read
18.5.1 Notes on Clock Pulse Generator	540	Description amended Note that the frequency of ϕ will be changed when setting SCKCR or PLLCR while executing the external bus cycle with the Write-data-buffer function.
Section 19 Power-Down Mode Table 19.1 Operating Mode	544	EXDMAC and DMAC description deleted from table 19.1
20.2 Register Bits	566	Table amended

Register Name	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*1
SAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
MRB	CHNE	DISEL	CHNS	—	—	—	—	—	
DAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRB	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

Item	Page	Revision (See Manual for Details)
20.2 Register Bits	567	Table amended

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ITSR	—	—	—	—	—	—	—	—	INt
	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0	
SSIER	—	—	—	—	—	—	—	—	
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0	

568

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PFCR2	—	—	—	—	ASOE	LWROE	—	—	PORT

569 Description amended

BROMCRH, BROMCRL, DRAMCR, DRACCR, REFCR, RTCNT, and RTCOR deleted from table

BCR (Before) ICIS2*7 → (After) ICIS2

570 Table amended

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	—	—	—	—	—	—	—	—	
DTCERC	—	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	—	—	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	—	—	—	—	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	DTCEG7	DTCEG6	—	—	—	—	—	—	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—	INT
IER	—	—	—	—	—	—	—	—	
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	—	—	—	—	—	—	—	—	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
NDRH*1	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	PPG
NDRL*1	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
NDRH*1	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
NDRL*1	—	—	—	—	NDR3	NDR2	NDR1	NDR0	

572

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SMR_2	C/$\overline{A}$/ GM*2	CHR/ BLK*3	PE	O/$\overline{E}$	STOP/ BCP1*4	MP/ BCP0*5	CKS1	CKS0	SCI_2, Smart card interface_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	TDRE	RDRF	ORER	FER/ ERS*6	PER	TEND	MPB	MPBT	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	

Item	Page	Revision (See Manual for Details)
20.2 Register Bits	573	Table amended

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CH3	—	—	
DADR0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	D/A

574

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	FLASH (F-ZTAT version)
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	EB13	EB12	EB11	EB10	EB9	EB8	

575 Notes amended

Notes: 1. If the PCR setting specifies the same output trigger for pulse output group 2 and pulse output group 3, the address is H'FF4C. If the triggers are different, the NDRH address corresponding to pulse output group 2 is H'FF4E and the NDRH address corresponding to pulse output group 3 is H'FF4C. In like manner, if the PCR setting specifies the same output trigger for pulse output group 0 and pulse output group 1, the address is H'FF4D. If the triggers are different, the NDRH address corresponding to pulse output group 0 is H'FF4F and the NDRH address corresponding to pulse output group 1 is H'FF4D.

2. Functions as C/$\overline{A}$ for SCI use, ...

3. Functions as CHR for SCI use, ...

4. Functions as STOP for SCI use, ...

5. Functions as MP for SCI use, ...

6. Functions as FER for SCI use, ...

7. Loaded in on-chip RAM. The bus width is 32 bits when the DTC accesses this area as register information, and 16 bits otherwise.

20.3 Register Stated in Each Operating Mode	576	Note *1 deleted (Before) SEMR*1 → (After) SEMR
	578	(Before) RAMER*1 → (After) RAMER
	582	(Before) FLMCR1*1 → (After) FLMCR1 (Before) FLMCR2*1 → (After) FLMCR2 (Before) EBR1*1 → (After) EBR1 (Before) EBR2*1 → (After) EBR2

Item	Page	Revision (See Manual for Details)
21.1.6 Flash Memory Characteristics Table 21.12 Flash Memory Characteristics	607	Table 21.12 amended

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Programming time*1 *2 *4	t_P	—	10	200	ms/ 128 bytes	
Erase time*1 *3 *5	t_E	—	50	1000	ms/ 128 bytes	
Rewrite times	N_{WEC}	100*1	10000*2	—	Times	
Data retention time*3	t_{DRP}	10	—	—	Years	
Programming Wait time after SWE bit setting*1	x	1	—	—	µs	

Contents

Figures

Section 6 Bus Controller (BSC)

Section 7 Data Transfer Controller (DTC)

Section 9 16-Bit Timer Pulse Unit (TPU)

Section 10 Programmable Pulse Generator (PPG)

Section 11 8-Bit Timers (TMR)

Section 12 Watchdog Timer

Section 13 Serial Communication Interface (SCI, IrDA)

Section 14 A/D Converter

Section 15 D/A Converter

Section 17 Flash Memory (F-ZTAT Version)

Section 18 Clock Pulse Generator

Section 19 Power-Down Modes

Section 21 Electrical Characteristics

Appendix

Tables

Section 6 Bus Controller (BSC)

Section 7 Data Transfer Controller (DTC)

Section 8 I/O Ports

Section 9 16-Bit Timer Pulse Unit (TPU)

Section 14 A/D Converter

Section 15 D/A Converter

Section 17 Flash Memory (F-ZTAT Version)

Section 18 Clock Pulse Generator

Section 1 Overview

1.1 Features

- High-speed H8S/2600 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
  69 basic instructions
- Various peripheral functions
  Data transfer controller (DTC)
  16-bit timer-pulse unit (TPU)
  Programmable pulse generator (PPG)
  8-bit timer (TMR)
  Watchdog timer (WDT)
  Asynchronous or clocked synchronous serial communication interface (SCI)
  10-bit A/D converter
  8-bit D/A converter
  Clock pulse generator
- On-chip memory

ROM Type	Model	ROM	RAM
Flash memory version	HD64F2667	384 kbytes	16 kbytes

- General I/O ports
  I/O pins: 103
  Input-only pins: 12
- Supports various power-down states
- Compact package

Package	(Code)	Body Size	Pin Pitch
LQFP-144	FP-144H	22.0 × 22.0 mm	0.5 mm

1.2 Block Diagram

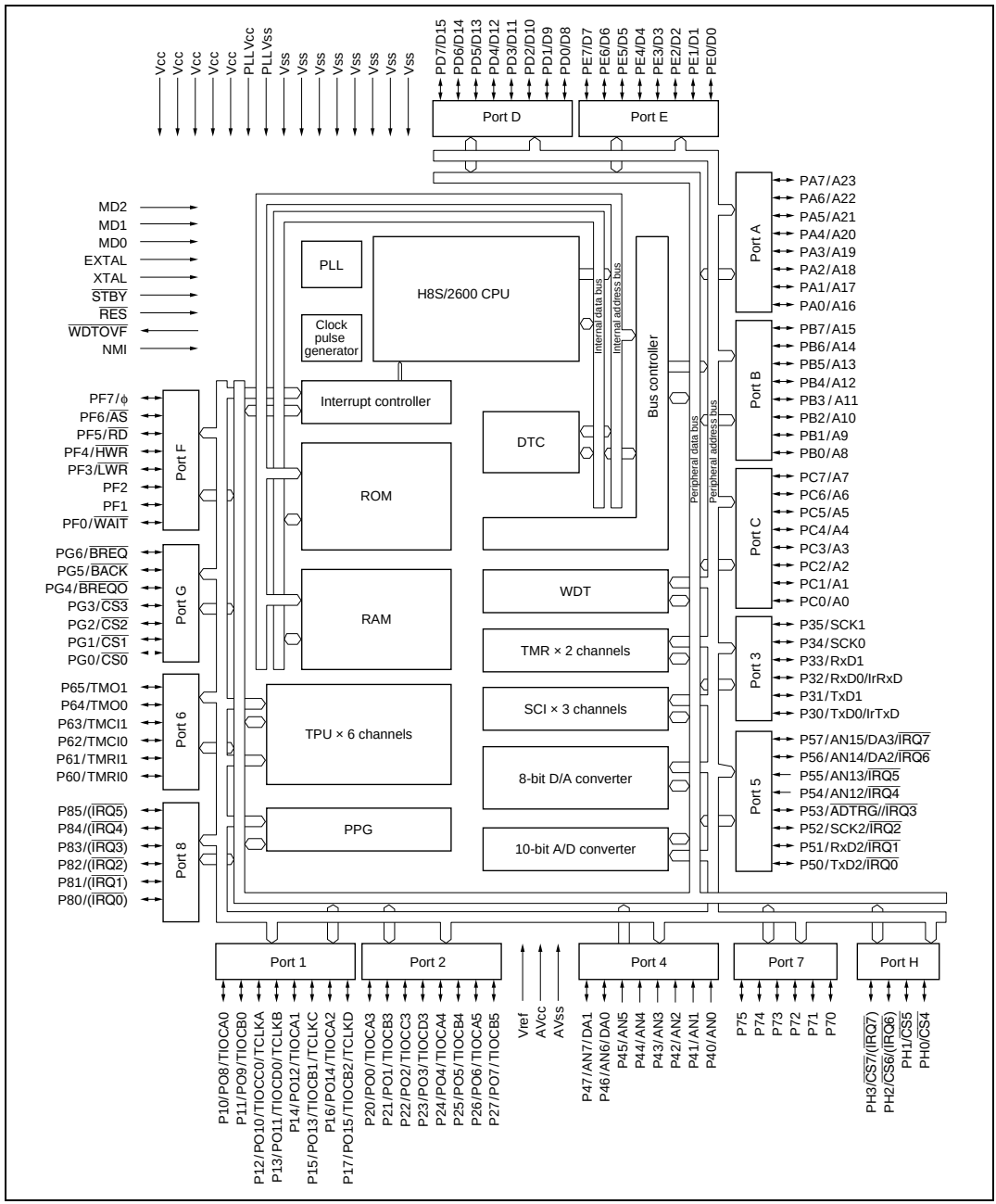
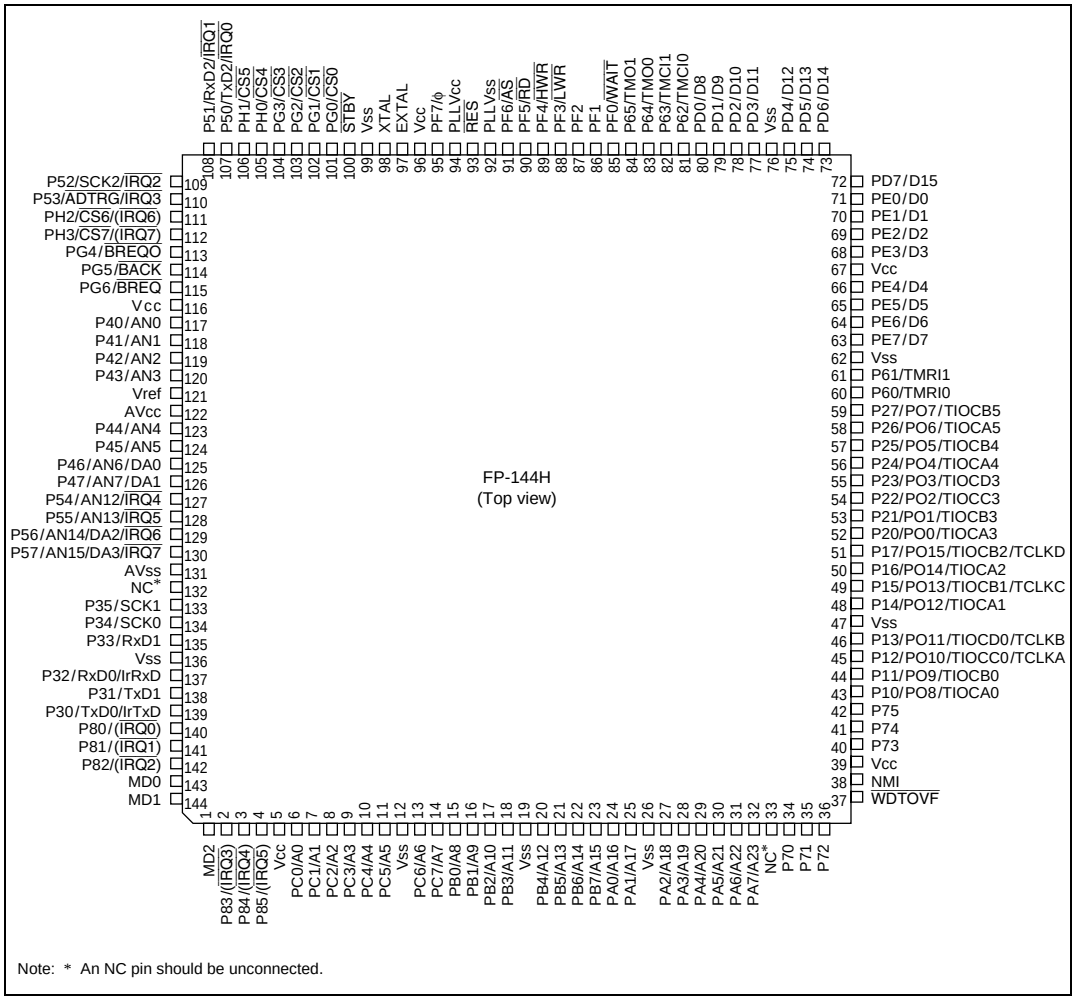


Figure 1.1 Internal Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement



Note: * An NC pin should be unconnected.

Figure 1.2 Pin Arrangement

1.3.2 Pin Arrangement in Each Operating Mode

Table 1.1 Pin Arrangement in Each Operating Mode

	Pin Name					
				Mode 7		Flash Memory Programmer Mode
Pin No.	Modes 1 and 5	Modes 2 and 6	Mode 4	EXPE = 1	EXPE = 0	
1	MD2	MD2	MD2	MD2	MD2	Vss
2	P83/($\overline{\text{IRQ3}}$)	P83/($\overline{\text{IRQ3}}$)	P83/($\overline{\text{IRQ3}}$)	P83/($\overline{\text{IRQ3}}$)	P83/($\overline{\text{IRQ3}}$)	NC
3	P84/($\overline{\text{IRQ4}}$)	P84/($\overline{\text{IRQ4}}$)	P84/($\overline{\text{IRQ4}}$)	P84/($\overline{\text{IRQ4}}$)	P84/($\overline{\text{IRQ4}}$)	NC
4	P85/($\overline{\text{IRQ5}}$)	P85/($\overline{\text{IRQ5}}$)	P85/($\overline{\text{IRQ5}}$)	P85/($\overline{\text{IRQ5}}$)	P85/($\overline{\text{IRQ5}}$)	NC
5	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
6	A0	A0	PC0/A0	PC0/A0	PC0	A0
7	A1	A1	PC1/A1	PC1/A1	PC1	A1
8	A2	A2	PC2/A2	PC2/A2	PC2	A2
9	A3	A3	PC3/A3	PC3/A3	PC3	A3
10	A4	A4	PC4/A4	PC4/A4	PC4	A4
11	A5	A5	PC5/A5	PC5/A5	PC5	A5
12	Vss	Vss	Vss	Vss	Vss	Vss
13	A6	A6	PC6/A6	PC6/A6	PC6	A6
14	A7	A7	PC7/A7	PC7/A7	PC7	A7
15	A8	A8	PB0/A8	PB0/A8	PB0	A8
16	A9	A9	PB1/A9	PB1/A9	PB1	A9
17	A10	A10	PB2/A10	PB2/A10	PB2	A10
18	A11	A11	PB3/A11	PB3/A11	PB3	A11
19	Vss	Vss	Vss	Vss	Vss	Vss
20	A12	A12	PB4/A12	PB4/A12	PB4	A12
21	A13	A13	PB5/A13	PB5/A13	PB5	A13
22	A14	A14	PB6/A14	PB6/A14	PB6	A14
23	A15	A15	PB7/A15	PB7/A15	PB7	A15
24	A16	A16	PA0/A16	PA0/A16	PA0	A16
25	A17	A17	PA1/A17	PA1/A17	PA1	A17
26	Vss	Vss	Vss	Vss	Vss	Vss
27	A18	A18	PA2/A18	PA2/A18	PA2	A18
28	A19	A19	PA3/A19	PA3/A19	PA3	NC

Pin No.	Pin Name					
	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Flash Memory Programmer Mode
				EXPE = 1	EXPE = 0	
29	A20	A20	PA4/A20	PA4/A20	PA4	NC
30	PA5/A21	PA5/A21	PA5/A21	PA5/A21	PA5	NC
31	PA6/A22	PA6/A22	PA6/A22	PA6/A22	PA6	NC
32	PA7/A23	PA7/A23	PA7/A23	PA7/A23	PA7	NC
33	NC	NC	NC	NC	NC	NC
34	P70	P70	P70	P70	P70	NC
35	P71	P71	P71	P71	P71	NC
36	P72	P72	P72	P72	P72	NC
37	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	NC
38	NMI	NMI	NMI	NMI	NMI	Vcc
39	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
40	P73	P73	P73	P73	P73	NC
41	P74	P74	P74	P74	P74	NC
42	P75	P75	P75	P75	P75	NC
43	P10/PO8/ TIOCA0	P10/PO8/ TIOCA0	P10/PO8/ TIOCA0	P10/PO8/ TIOCA0	P10/PO8/ TIOCA0	NC
44	P11/PO9/ TIOCB0	P11/PO9/ TIOCB0	P11/PO9/ TIOCB0	P11/PO9/ TIOCB0	P11/PO9/ TIOCB0	NC
45	P12/PO10/ TIOCC0/ TCLKA	P12/PO10/ TIOCC0/ TCLKA	P12/PO10/ TIOCC0/ TCLKA	P12/PO10/ TIOCC0/ TCLKA	P12/PO10/ TIOCC0/ TCLKA	NC
46	P13/PO11/ TIOCD0/ TCLKB	P13/PO11/ TIOCD0/ TCLKB	P13/PO11/ TIOCD0/ TCLKB	P13/PO11/ TIOCD0/ TCLKB	P13/PO11/ TIOCD0/ TCLKB	NC
47	Vss	Vss	Vss	Vss	Vss	Vss
48	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	P14/PO12/ TIOCA1	NC
49	P15/PO13/ TIOCB1/ TCLKC	P15/PO13/ TIOCB1/ TCLKC	P15/PO13/ TIOCB1/ TCLKC	P15/PO13/ TIOCB1/ TCLKC	P15/PO13/ TIOCB1/ TCLKC	NC
50	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	NC
51	P17/PO15/ TIOCB2/ TCLKD	P17/PO15/ TIOCB2/ TCLKD	P17/PO15/ TIOCB2/ TCLKD	P17/PO15/ TIOCB2/ TCLKD	P17/PO15/ TIOCB2/ TCLKD	NC

Pin No.	Pin Name: Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7: EXPE = 1	Mode 7: EXPE = 0	Flash Memory Programmer Mode
52	P20/PO0/ TIOCA3	P20/PO0/ TIOCA3	P20/PO0/ TIOCA3	P20/PO0/ TIOCA3	P20/PO0/ TIOCA3	NC
53	P21/PO1/ TIOCB3	P21/PO1/ TIOCB3	P21/PO1/ TIOCB3	P21/PO1/ TIOCB3	P21/PO1/ TIOCB3	NC
54	P22/PO2/ TIOCC3	P22/PO2/ TIOCC3	P22/PO2/ TIOCC3	P22/PO2/ TIOCC3	P22/PO2/ TIOCC3	$\overline{OE}$
55	P23/PO3/ TIOCD3	P23/PO3/ TIOCD3	P23/PO3/ TIOCD3	P23/PO3/ TIOCD3	P23/PO3/ TIOCD3	$\overline{CE}$
56	P24/PO4/ TIOCA4	P24/PO4/ TIOCA4	P24/PO4/ TIOCA4	P24/PO4/ TIOCA4	P24/PO4/ TIOCA4	$\overline{WE}$
57	P25/PO5/ TIOCB4	P25/PO5/ TIOCB4	P25/PO5/ TIOCB4	P25/PO5/ TIOCB4	P25/PO5/ TIOCB4	Vss
58	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	NC
59	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	NC
60	P60/TMRI0	P60/TMRI0	P60/TMRI0	P60/TMRI0	P60/TMRI0	NC
61	P61/TMRI1	P61/TMRI1	P61/TMRI1	P61/TMRI1	P61/TMRI1	NC
62	Vss	Vss	Vss	Vss	Vss	Vss
63	D7	PE7/D7	PE7/D7	PE7/D7	PE7	NC
64	D6	PE6/D6	PE6/D6	PE6/D6	PE6	NC
65	D5	PE5/D5	PE5/D5	PE5/D5	PE5	NC
66	D4	PE4/D4	PE4/D4	PE4/D4	PE4	NC
67	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
68	D3	PE3/D3	PE3/D3	PE3/D3	PE3	NC
69	D2	PE2/D2	PE2/D2	PE2/D2	PE2	NC
70	D1	PE1/D1	PE1/D1	PE1/D1	PE1	NC
71	D0	PE0/D0	PE0/D0	PE0/D0	PE0	NC
72	D15	D15	D15	D15	PD7	I/O7
73	D14	D14	D14	D14	PD6	I/O6
74	D13	D13	D13	D13	PD5	I/O5
75	D12	D12	D12	D12	PD4	I/O4
76	Vss	Vss	Vss	Vss	Vss	Vss

Pin No.	Pin Name: Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7: EXPE = 1	Mode 7: EXPE = 0	Flash Memory Programmer Mode
77	D11	D11	D11	D11	PD3	I/O3
78	D10	D10	D10	D10	PD2	I/O2
79	D9	D9	D9	D9	PD1	I/O1
80	D8	D8	D8	D8	PD0	I/O0
81	P62/TMCI0	P62/TMCI0	P62/TMCI0	P62/TMCI0	P62/TMCI0	NC
82	P63/TMCI1	P63/TMCI1	P63/TMCI1	P63/TMCI1	P63/TMCI1	NC
83	P64/TMO0	P64/TMO0	P64/TMO0	P64/TMO0	P64/TMO0	NC
84	P65/TMO1	P65/TMO1	P65/TMO1	P65/TMO1	P65/TMO1	NC
85	PF0/$\overline{\text{WAIT}}$	PF0/$\overline{\text{WAIT}}$	PF0/$\overline{\text{WAIT}}$	PF0/$\overline{\text{WAIT}}$	PF0	NC
86	PF1	PF1	PF1	PF1	PF1	NC
87	PF2	PF2	PF2	PF2	PF2	NC
88	PF3/$\overline{\text{LWR}}$	PF3/$\overline{\text{LWR}}$	PF3/$\overline{\text{LWR}}$	PF3/$\overline{\text{LWR}}$	PF3	NC
89	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF4	NC
90	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF5	NC
91	PF6/$\overline{\text{AS}}$	PF6/$\overline{\text{AS}}$	PF6/$\overline{\text{AS}}$	PF6/$\overline{\text{AS}}$	PF6	NC
92	PLLVss	PLLVss	PLLVss	PLLVss	PLLVss	Vss
93	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
94	PLLVcc	PLLVcc	PLLVcc	PLLVcc	PLLVcc	Vcc
95	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ	NC
96	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
97	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
98	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
99	Vss	Vss	Vss	Vss	Vss	Vss
100	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	Vcc
101	PG0/$\overline{\text{CS0}}$	PG0/$\overline{\text{CS0}}$	PG0/$\overline{\text{CS0}}$	PG0/$\overline{\text{CS0}}$	PG0	NC
102	PG1/$\overline{\text{CS1}}$	PG1/$\overline{\text{CS1}}$	PG1/$\overline{\text{CS1}}$	PG1/$\overline{\text{CS1}}$	PG1	NC
103	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2	NC
104	PG3/$\overline{\text{CS3}}$	PG3/$\overline{\text{CS3}}$	PG3/$\overline{\text{CS3}}$	PG3/$\overline{\text{CS3}}$	PG3	NC
105	PH0/$\overline{\text{CS4}}$	PH0/$\overline{\text{CS4}}$	PH0/$\overline{\text{CS4}}$	PH0/$\overline{\text{CS4}}$	PH0	NC
106	PH1/$\overline{\text{CS5}}$	PH1/$\overline{\text{CS5}}$	PH1/$\overline{\text{CS5}}$	PH1/$\overline{\text{CS5}}$	PH1	NC

Pin No.	Pin Name: Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7: EXPE = 1	Mode 7: EXPE = 0	Flash Memory Programmer Mode
107	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	Vss
108	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	Vss
109	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	Vcc
110	P53/ $\overline{\text{ADTRG}}$/ $\overline{\text{IRQ3}}$	P53/ $\overline{\text{ADTRG}}$/ $\overline{\text{IRQ3}}$	P53/ $\overline{\text{ADTRG}}$/ $\overline{\text{IRQ3}}$	P53/ $\overline{\text{ADTRG}}$/ $\overline{\text{IRQ3}}$	P53/ $\overline{\text{ADTRG}}$/ $\overline{\text{IRQ3}}$	NC
111	PH2/ $\overline{\text{CS6}}$/ ($\overline{\text{IRQ6}}$)	PH2/ $\overline{\text{CS6}}$/ ($\overline{\text{IRQ6}}$)	PH2/ $\overline{\text{CS6}}$/ ($\overline{\text{IRQ6}}$)	PH2/ $\overline{\text{CS6}}$/ ($\overline{\text{IRQ6}}$)	PH2/($\overline{\text{IRQ6}}$)	NC
112	PH3/ $\overline{\text{CS7}}$/ ($\overline{\text{IRQ7}}$)	PH3/ $\overline{\text{CS7}}$/ ($\overline{\text{IRQ7}}$)	PH3/ $\overline{\text{CS7}}$/ ($\overline{\text{IRQ7}}$)	PH3/ $\overline{\text{CS7}}$/ ($\overline{\text{IRQ7}}$)	PH3/($\overline{\text{IRQ7}}$)	NC
113	PG4/ $\overline{\text{BREQO}}$	PG4/ $\overline{\text{BREQO}}$	PG4/ $\overline{\text{BREQO}}$	PG4/ $\overline{\text{BREQO}}$	PG4	NC
114	PG5/ $\overline{\text{BACK}}$	PG5/ $\overline{\text{BACK}}$	PG5/ $\overline{\text{BACK}}$	PG5/ $\overline{\text{BACK}}$	PG5	NC
115	PG6/ $\overline{\text{BREQ}}$	PG6/ $\overline{\text{BREQ}}$	PG6/ $\overline{\text{BREQ}}$	PG6/ $\overline{\text{BREQ}}$	PG6	NC
116	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
117	P40/AN0	P40/AN0	P40/AN0	P40/AN0	P40/AN0	NC
118	P41/AN1	P41/AN1	P41/AN1	P41/AN1	P41/AN1	NC
119	P42/AN2	P42/AN2	P42/AN2	P42/AN2	P42/AN2	NC
120	P43/AN3	P43/AN3	P43/AN3	P43/AN3	P43/AN3	NC
121	Vref	Vref	Vref	Vref	Vref	NC
122	AVcc	AVcc	AVcc	AVcc	AVcc	Vcc
123	P44/AN4	P44/AN4	P44/AN4	P44/AN4	P44/AN4	NC
124	P45/AN5	P45/AN5	P45/AN5	P45/AN5	P45/AN5	NC
125	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	NC
126	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	NC
127	P54/AN12/ $\overline{\text{IRQ4}}$	P54/AN12/ $\overline{\text{IRQ4}}$	P54/AN12/ $\overline{\text{IRQ4}}$	P54/AN12/ $\overline{\text{IRQ4}}$	P54/AN12/ $\overline{\text{IRQ4}}$	NC
128	P55/AN13/ $\overline{\text{IRQ5}}$	P55/AN13/ $\overline{\text{IRQ5}}$	P55/AN13/ $\overline{\text{IRQ5}}$	P55/AN13/ $\overline{\text{IRQ5}}$	P55/AN13/ $\overline{\text{IRQ5}}$	NC
129	P56/AN14/ DA2/ $\overline{\text{IRQ6}}$	P56/AN14/ DA2/ $\overline{\text{IRQ6}}$	P56/AN14/ DA2/ $\overline{\text{IRQ6}}$	P56/AN14/ DA2/ $\overline{\text{IRQ6}}$	P56/AN14/ DA2/ $\overline{\text{IRQ6}}$	NC
130	P57/AN15/ DA3/ $\overline{\text{IRQ7}}$	P57/AN15/ DA3/ $\overline{\text{IRQ7}}$	P57/AN15/ DA3/ $\overline{\text{IRQ7}}$	P57/AN15/ DA3/ $\overline{\text{IRQ7}}$	P57/AN15/ DA3/ $\overline{\text{IRQ7}}$	NC

Pin No.	Pin Name					
	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Flash Memory Programmer Mode
				EXPE = 1	EXPE = 0	
131	AVss	AVss	AVss	AVss	AVss	Vss
132	NC	NC	NC	NC	NC	NC
133	P35/SCK1	P35/SCK1	P35/SCK1	P35/SCK1	P35/SCK1	NC
134	P34/SCK0	P34/SCK0	P34/SCK0	P34/SCK0	P34/SCK0	NC
135	P33/RxD1	P33/RxD1	P33/RxD1	P33/RxD1	P33/RxD1	NC
136	Vss	Vss	Vss	Vss	Vss	Vss
137	P32/RxD0/ IrRxD	P32/RxD0/ IrRxD	P32/RxD0/ IrRxD	P32/RxD0/ IrRxD	P32/RxD0/ IrRxD	Vcc
138	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	NC
139	P30/TxD0/ IrTxD	P30/TxD0/ IrTxD	P30/TxD0/ IrTxD	P30/TxD0/ IrTxD	P30/TxD0/ IrTxD	NC
140	P80/($\overline{\text{IRQ0}}$)	P80/($\overline{\text{IRQ0}}$)	P80/($\overline{\text{IRQ0}}$)	P80/($\overline{\text{IRQ0}}$)	P80/($\overline{\text{IRQ0}}$)	NC
141	P81/($\overline{\text{IRQ1}}$)	P81/($\overline{\text{IRQ1}}$)	P81/($\overline{\text{IRQ1}}$)	P81/($\overline{\text{IRQ1}}$)	P81/($\overline{\text{IRQ1}}$)	NC
142	P82/($\overline{\text{IRQ2}}$)	P82/($\overline{\text{IRQ2}}$)	P82/($\overline{\text{IRQ2}}$)	P82/($\overline{\text{IRQ2}}$)	P82/($\overline{\text{IRQ2}}$)	NC
143	MD0	MD0	MD0	MD0	MD0	Vss
144	MD1	MD1	MD1	MD1	MD1	Vss

1.3.3 Pin Functions

Table 1.2 Pin Functions

Type	Symbol	Pin No. FP-144H	I/O	Function
Power	V_{CC}	5, 39, 67, 96, 116	Input	For connection to the power supply. All V_{CC} pins should be connected to the system power supply.
	V_{SS}	12, 19, 26, 47, 76, 99, 136	Input	For connection to ground. All V_{SS} pins should be connected to the system power supply (0 V).
	$PLLV_{CC}$	94	Input	Power supply pin for the on-chip PLL oscillator.
	$PLLV_{SS}$	92	Input	Ground pin for the on-chip PLL oscillator.
Clock	XTAL	98	Input	For connection to a crystal oscillator. See section 18, Clock Pulse Generator for typical connection diagrams for a crystal oscillator and external clock input.
	EXTAL	97	Input	For connection to a crystal oscillator. The EXTAL pin can also input an external clock. See section 18, Clock Pulse Generator for typical connection diagrams for a crystal oscillator and external clock input.
	ϕ	95	Output	Supplies the system clock to external devices.
Operating mode control	MD2 MD1 MD0	1, 144, 143	Input	These pins set the operating mode. These pins should not be changed while the MCU is operating.
System control	$\overline{RES}$	93	Input	When this pin is driven low, the chip is reset.
	$\overline{STBY}$	100	Input	When this pin is driven low, a transition is made to hardware standby mode.
	$\overline{BREQ}$	115	Input	Requests chip to release the bus to an external bus master.
	$\overline{BREQO}$	113	Output	External bus request signal used when an internal bus master accesses external space when the external bus is released.
	$\overline{BACK}$	114	Output	Indicates that the bus has been released to an external bus master.

Type	Symbol	Pin No. FP-144H	I/O	Function
Address bus	A23 to A0	32 to 27, 25 to 20, 18 to 13, 11 to 6	Output	These pins output an address.
Data bus	D15 to D0	72 to 75, 77 to 80, 63 to 66, 68 to 71	Input/ output	These pins constitute a bidirectional data bus.
Bus control	$\overline{CS7}$ to $\overline{CS0}$	112, 111, 106 to 101	Output	Signals that select division areas 7 to 0 in the external address space.
	$\overline{AS}$	91	Output	When this pin is low, it indicates that address output on the address bus is valid.
	$\overline{RD}$	90	Output	When this pin is low, it indicates that the external address space is being read.
	$\overline{HWR}$	89	Output	Strobe signal indicating that external address space is to be written, and the upper half (D15 to D8) of the data bus is enabled. Write enable signal for DRAM interface space.
	$\overline{LWR}$	88	Output	Strobe signal indicating that external address space is to be written, and the lower half (D7 to D0) of the data bus is enabled.
	$\overline{WAIT}$	85	Input	Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.
Interrupt signals	NMI	38	Input	Nonmaskable interrupt request pin. Fix high when not used.
	$\overline{IRQ7}$ to $\overline{IRQ0}$	130 to 127, 110 to 107	Input	These pins request a maskable interrupt. The input pins of $\overline{DREQn}$ and ($\overline{DREQn}$) are selected by the IRQ pin select register (ITSR) of the interrupt controller. (n = 0 to 7)
	($\overline{IRQ7}$) to ($\overline{IRQ0}$)	112, 111, 4 to 2, 142 to 140		

Type	Symbol	Pin No. FP-144H	I/O	Function
16-bit timer pulse unit (TPU)	TCLKA TCLKB TCLKC TCLKD	45, 46, 49, 51	Input	External clock input pins.
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	43, 44, 45, 46	Input/ output	TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOCA1 TIOCB1	48, 49	Input/ output	TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOCA2 TIOCB2	50, 51	Input/ output	TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOCA3 TIOCB3 TIOCC3 TIOCD3	52, 53, 54, 55	Input/ output	TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOCA4 TIOCB4	56, 57	Input/ output	TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.
	TIOCA5, TIOCB5	58, 59	Input/ output	TGRA_5 and TGRB_5 input capture input/output compare output/PWM output pins.
Programmable pulse generator (PPG)	PO15 to PO0	51 to 48, 46 to 43, 59 to 52	Output	Pulse output pins.
8-bit timer	TMO0 TMO1	83, 84	Output	Waveform output pins with output compare function.
	TMCI0 TMCI1	81, 82	Input	External event input pins.
	TMRI0 TMRI1	60, 61	Input	Counter reset input pins.
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$	37	Output	Counter overflow signal output pin in watchdog timer mode.

Type	Symbol	Pin No. FP-144H	I/O	Function
Serial communication interface (SCI)/smart card interface (SCI_0 with IrDA function)	TxD2 TxD1 TxD0/IrTxD	107, 138, 139	Output	Data output pins.
	RxD2 RxD1 RxD0/ IrRxD	108, 135, 137	Input	Data input pins.
	SCK2 SCK1 SCK0	109, 133, 134	Input/ output	Clock input/output pins.
A/D converter	AN15 to AN12, AN7 to AN0	130 to 127, 126 to 123, 120 to 117	Input	Analog input pins for the A/D converter.
	$\overline{\text{ADTRG}}$	110	Input	Pin for input of an external trigger to start A/D conversion.
D/A converter	DA3 to DA0	130, 129, 126, 125	Output	Analog input pins for the D/A converter.
A/D converter, D/A converter	AV_{CC}	122	Input	The analog power-supply pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).
	AV_{SS}	131	Input	The ground pin for the A/D converter and D/A converter. This pin should be connected to the system power supply (0 V).
	Vref	121	Input	The reference voltage input pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).

Type	Symbol	Pin No. FP-144H	I/O	Function
I/O ports	P17 to P10	51 to 48, 46 to 43	Input/ output	Eight input/output pins.
	P27 to P20	59 to 52	Input/ output	Eight input/output pins.
	P35 to P30	133 to 135, 137 to 139	Input/ output	Six input/output pins.
	P47 to P40	126 to 123, 120 to 117	Input	Eight input pins.
	P57 to P54	130 to 127	Input	Four input pins.
	P53 to P50	110 to 107	Input/ output	Four input/output pins.
	P65 to P60	84 to 81, 61, 60	Input/ output	Six input/output pins.
	P75 to P70	42 to 40, 36 to 34	Input/ output	Six input/output pins.
	P85 to P80	4 to 2, 142 to 140	Input/ output	Six input/output pins.
	PA7 to PA0	32 to 27, 25, 24	Input/ output	Eight input/output pins.
	PB7 to PB0	23 to 20, 18 to 15	Input/ output	Eight input/output pins.
	PC7 to PC0	14, 13, 11 to 6	Input/ output	Eight input/output pins.
	PD7 to PD0	72 to 75, 77 to 80	Input/ output	Eight input/output pins.
	PE7 to PE0	63 to 66, 68 to 71	Input/ output	Eight input/output pins.
	PF7 to PF0	95, 91 to 85	Input/ output	Eight input/output pins.
	PG6 to PG0	115 to 113, 104 to 101	Input/ output	Seven input/output pins.
	PH3 to PH0	112, 111, 106, 105	Input/ output	Four input/output pins.

Section 2 CPU

The H8S/2600 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/2600 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/2600 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 CPUs
  Can execute H8/300 and H8/300H object programs
- General-register architecture
  Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-nine basic instructions
  8/16/32-bit arithmetic and logic instructions
  Multiply and divide instructions
  Powerful bit-manipulation instructions
  Multiply-and-accumulate instruction
- 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: 3 states
  16 ÷ 8-bit register-register divide: 12 states

16 × 16-bit register-register multiply: 4 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 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 as shown below.

- Register configuration
  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions
  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- 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, 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/2600 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.
  A multiply-and-accumulate instruction has been added.
  Two-bit shift and rotate 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.

Note: Normal mode is not available in this LSI.

2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhancements.

- Additional control register
  One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  Addressing modes of bit-manipulation instructions have been enhanced.
  A multiply-and-accumulate instruction has been added.
  Two-bit shift and rotate 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/2600 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

  The H8S/2600 CPU provides linear access to a maximum 64-kbyte 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.

  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. The exception vector table in normal mode is shown in figure 2.1. 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.

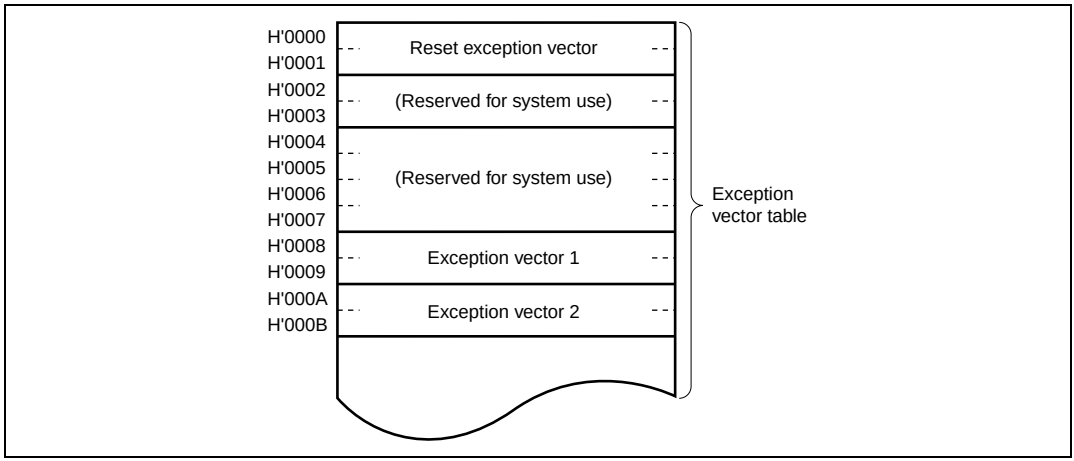
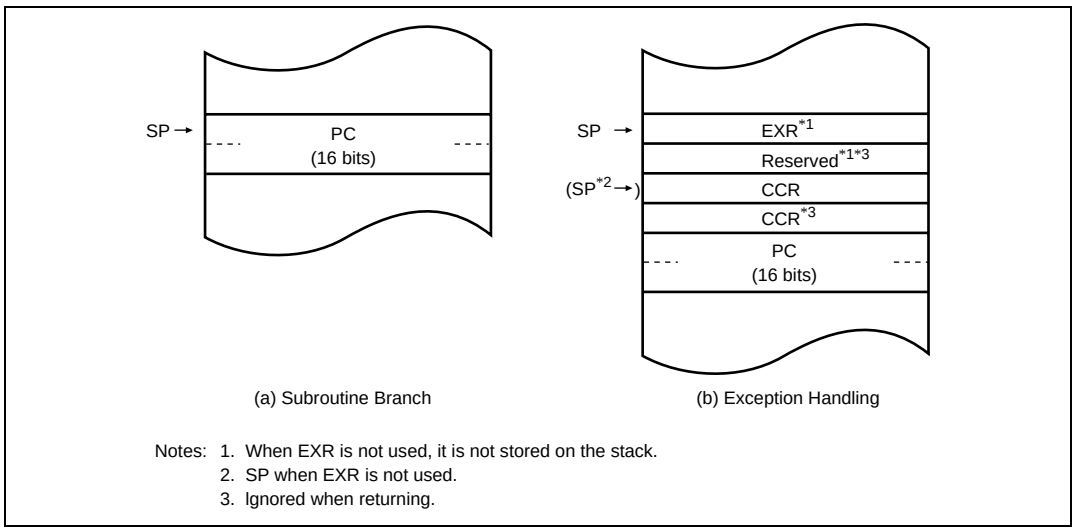


Figure 2.1 Exception Vector Table (Normal Mode)



Notes: 1. When EXR is not used, it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access is provided to a 16-Mbyte maximum 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.

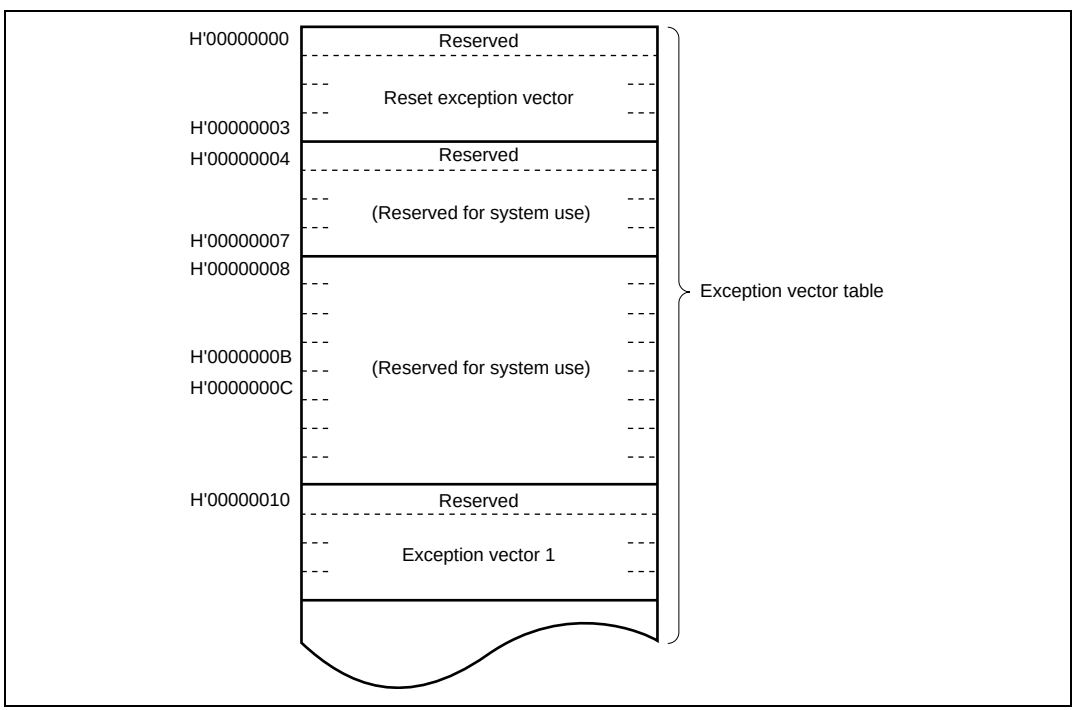


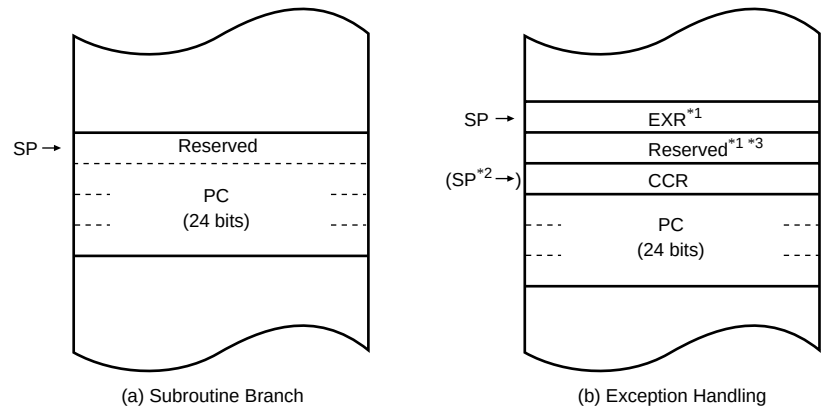
Figure 2.3 Exception Vector Table (Advanced Mode)

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 are 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 used for 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. EXR is not pushed onto the stack in interrupt control mode 0. 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.
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/2600 CPU. The H8S/2600 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.

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

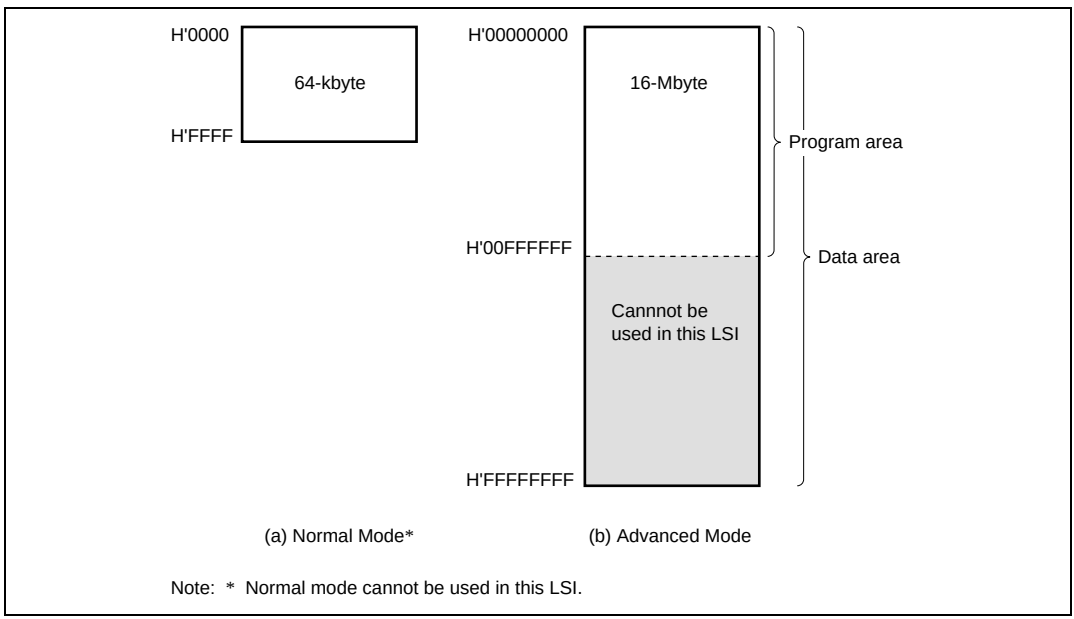
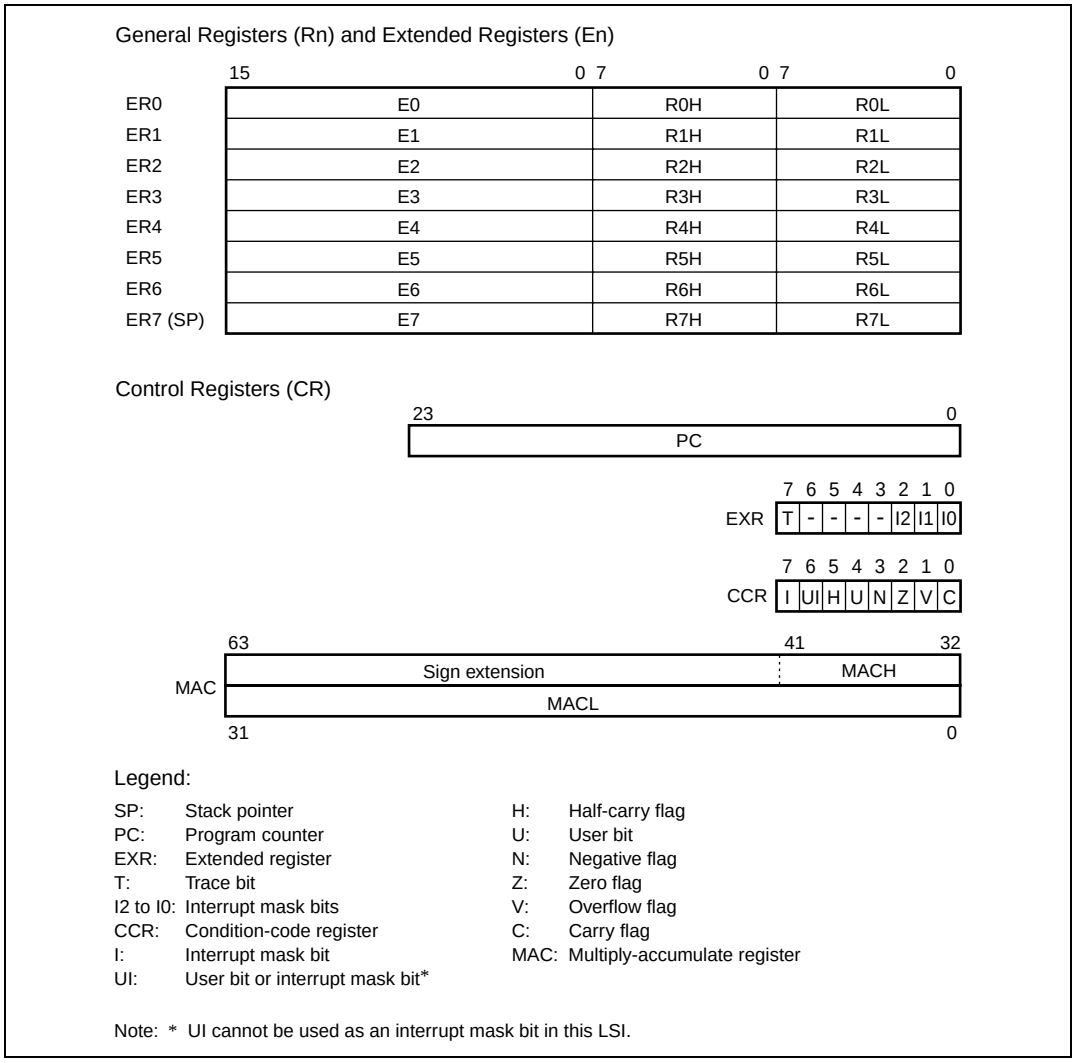


Figure 2.5 Memory Map

2.4 Registers

The H8S/2600 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 register (EXR), an 8-bit condition code register (CCR), and a 64-bit multiply-accumulate register (MAC).



Legend:

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit	MAC:	Multiply-accumulate register
UI:	User bit or interrupt mask bit*		

Note: * UI cannot be used as an interrupt mask bit in this LSI.

Figure 2.6 CPU Registers

2.4.1 General Registers

The H8S/2600 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 registers.

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.

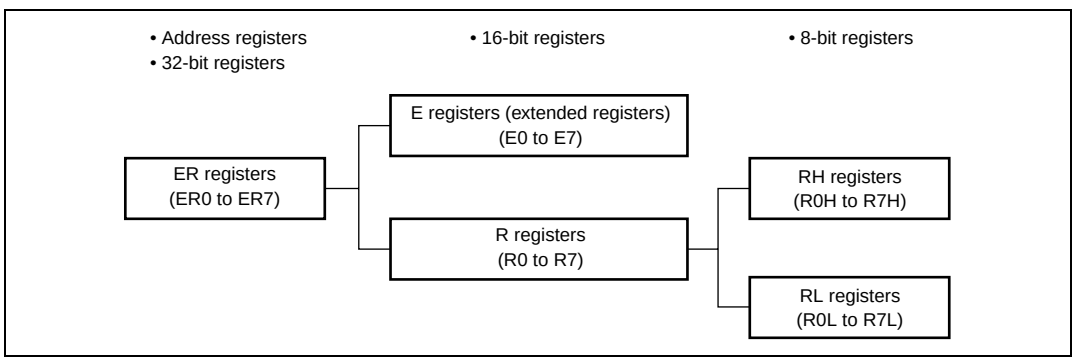


Figure 2.7 Usage of General Registers

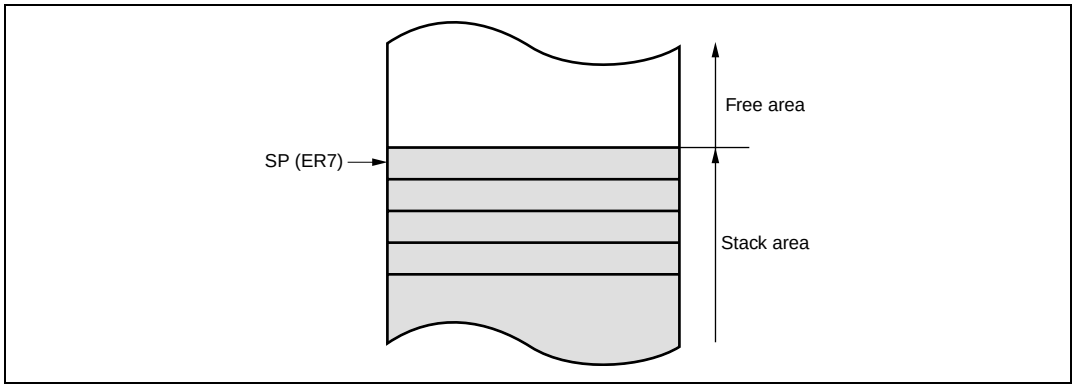


Figure 2.8 Stack

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 Register (EXR)

EXR is an 8-bit register that can be manipulated by 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 started each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6	—	All 1	—	Reserved They are always read as 1.
5				
4				
3				
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 otherwise.
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 Multiply-Accumulate Register (MAC)

This 64-bit register stores the results of multiply-and-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upper bits are a sign extension.

2.4.6 Initial Values of CPU Internal Registers

When the reset exception handling loads the start address from the vector address, PC is initialized, the T bit in EXR is cleared to 0, and the I bits in EXR and CCR are set to 1. However, the general registers and the other CCR bits are not initialized. The initial value of SP (ER7) is undefined. SP should therefore be initialized by using the MOV.L instruction immediately after a reset.

2.5 Data Formats

The H8S/2600 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.

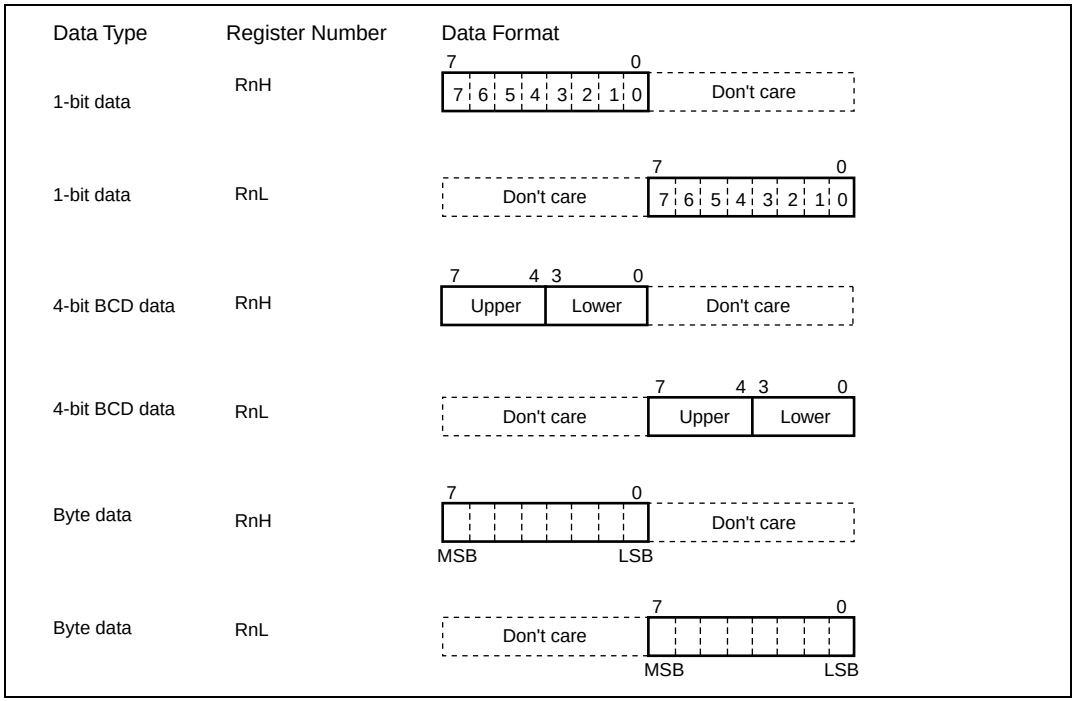


Figure 2.9 General Register Data Formats (1)

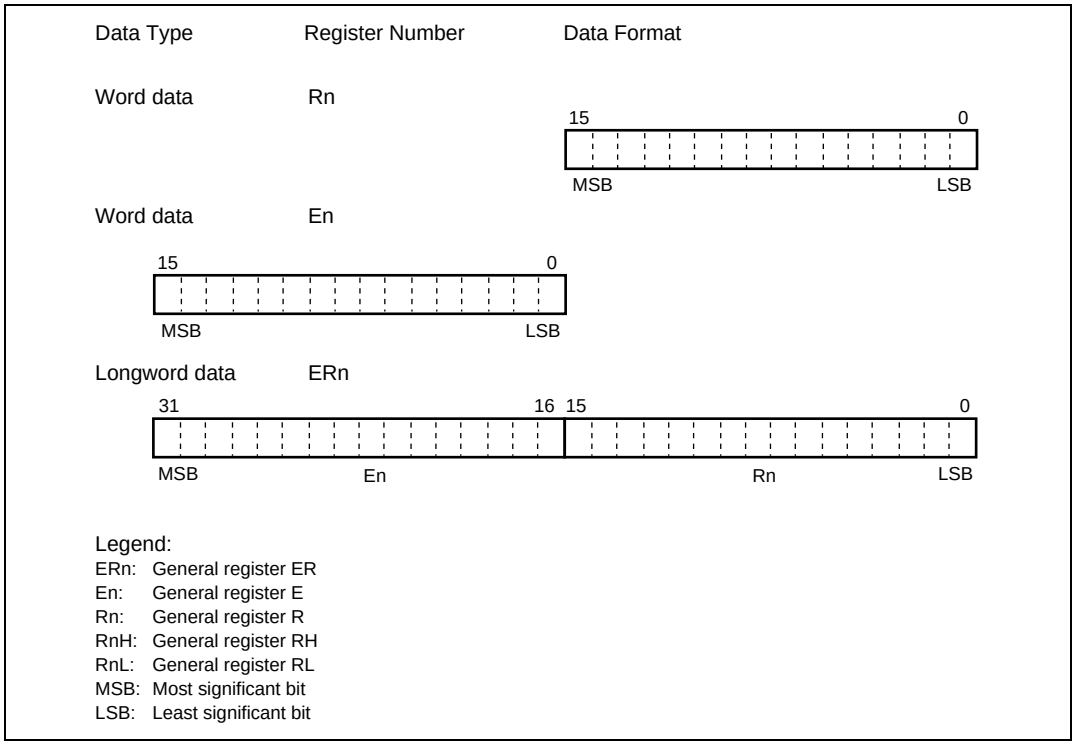


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2600 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 size or longword size.

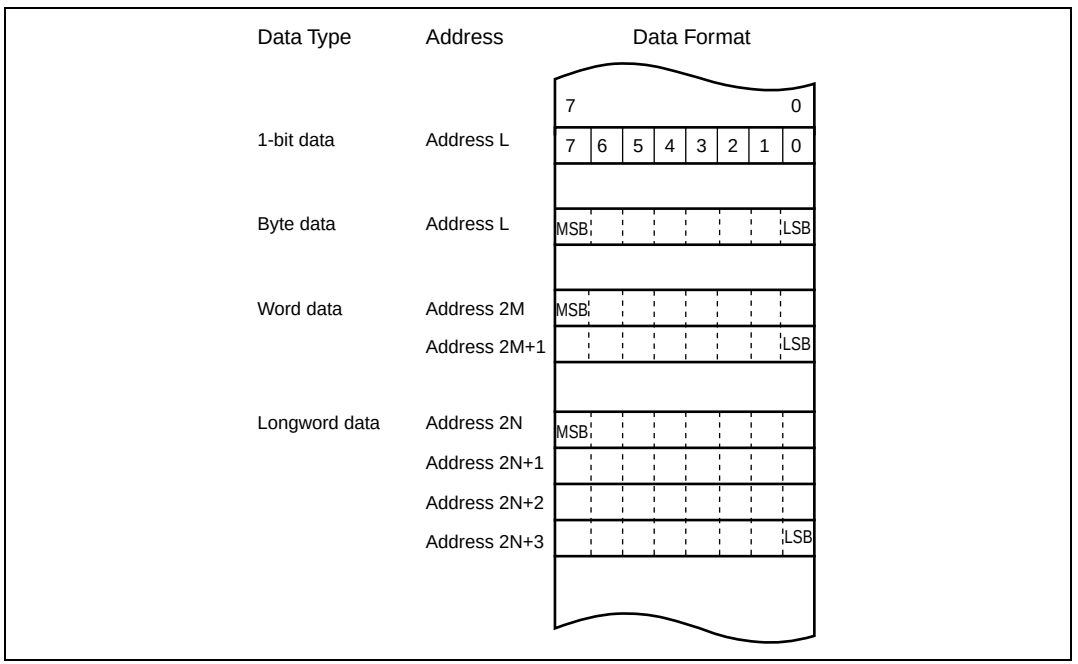


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2600 CPU has 69 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	23
	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	
	MAC, LDMAC, STMAC, CLRMAC	—	
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: 69

Legend: B: Byte
W: Word
L: Longword

Notes: 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)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended 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 exclusive OR
→	Move
¬	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit 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: * Size 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 or borrow 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: * Size 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.
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating
CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accumulate register.

Notes: 1. Size 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 (logical complement) of general register contents.

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

Table 2.6 Shift Instructions

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

Note: * Size 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: * Size 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 Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ ¬ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	(<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: * Size 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 contents of a general register or memory, 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 exclusive-ORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

Table 2.10 Block 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

The H8S/2600 Series instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), 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, 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.

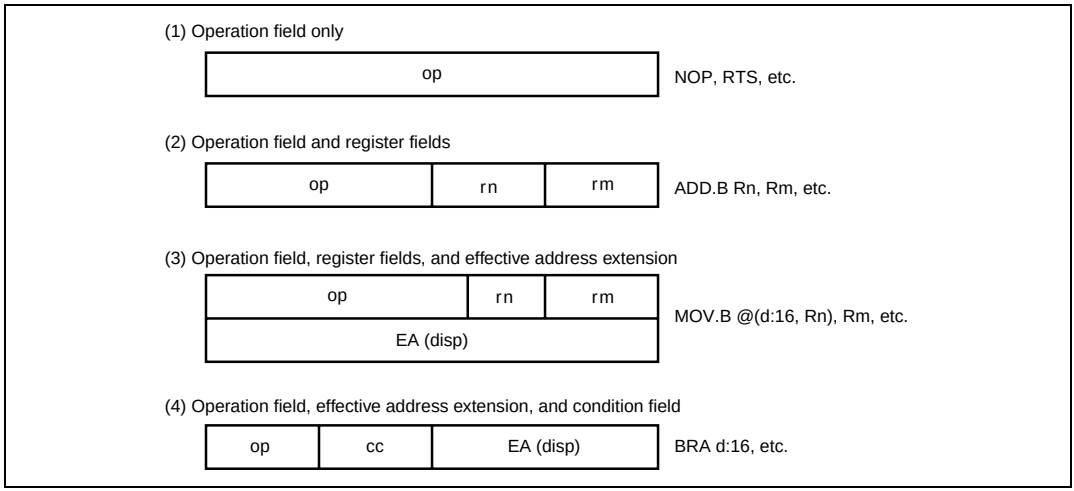


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2600 CPU supports the eight addressing modes listed in table 2.11. The usable address modes are different in each instruction.

Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute 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 code 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 code, 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 word or longword transfer instruction, 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 becomes 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 word or longword transfer instruction, 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: * Not available in this LSI.

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

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

The 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 code 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 all 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.

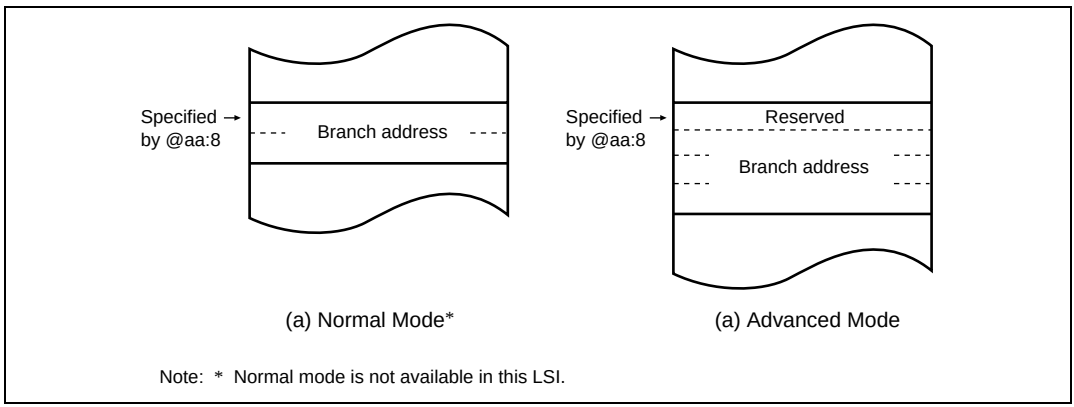


Figure 2.12 Branch Address Specification in Memory Indirect Mode

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.

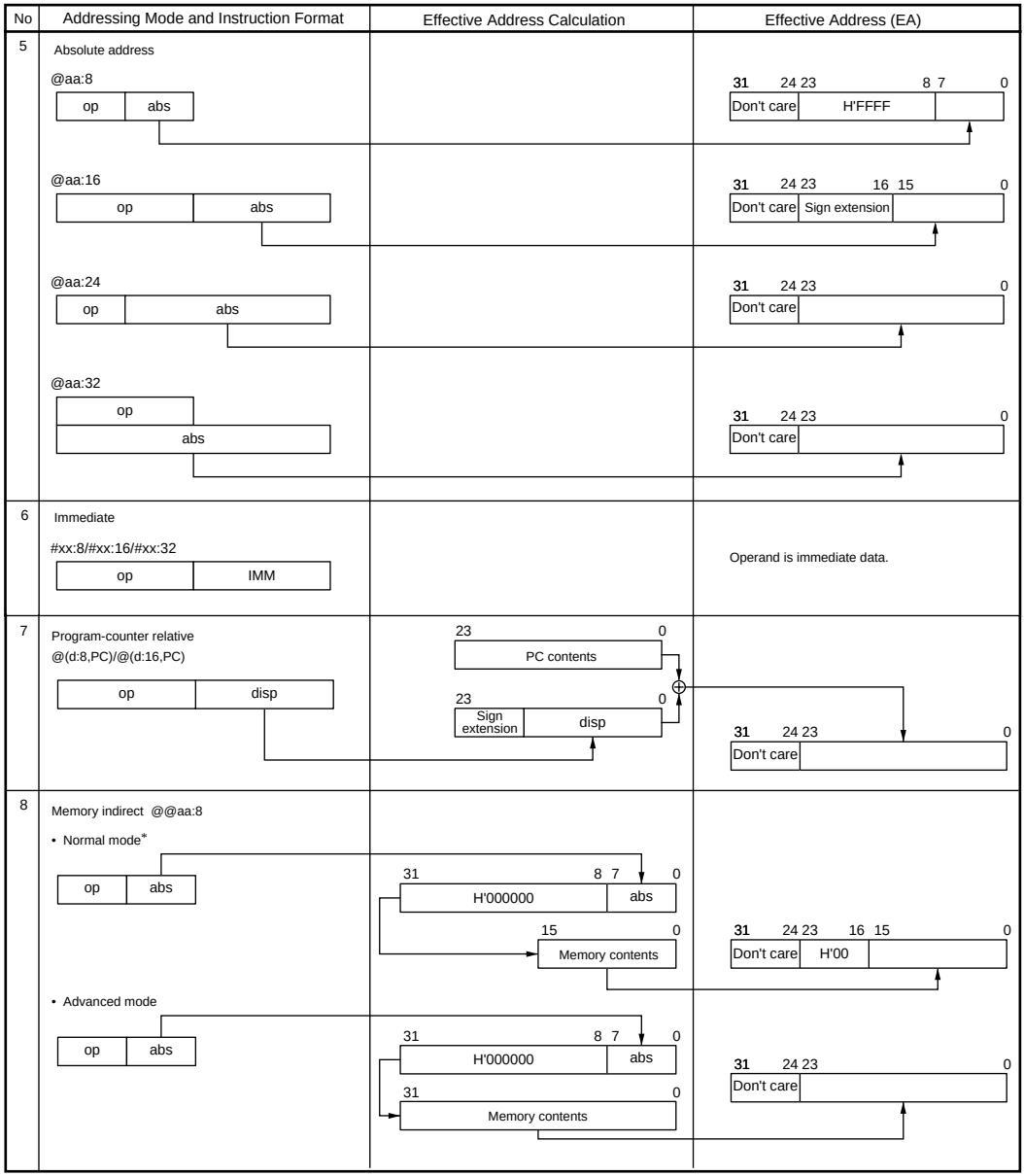
Note: Normal mode is not available in this LSI.

Table 2.13 Effective Address Calculation (1)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) op | rm | rn		Operand is general register contents.
2	Register indirect (@ERn) op | r	31 … 0: General register contents	31 … 24 23 … 0: Don't care
3	Register indirect with displacement @(d:16,ERn)/@(d:32,ERn) op | r | disp	31 … 0: General register contents 31 … 0: Sign extension | disp	31 … 24 23 … 0: Don't care
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+ op | r • Register indirect with pre-decrement @-ERn op | r	31 … 0: General register contents ± 1, 2, or 4	31 … 24 23 … 0: Don't care

Operand Size	Offset
Byte	1
Word	2
Longword	4

Table 2.13 Effective Address Calculation (2)



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

2.8 Processing States

The H8S/2600 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and program stop state. Figure 2.13 indicates the state transitions.

- Reset State

  The CPU and on-chip peripheral modules are all initialized and stop. 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

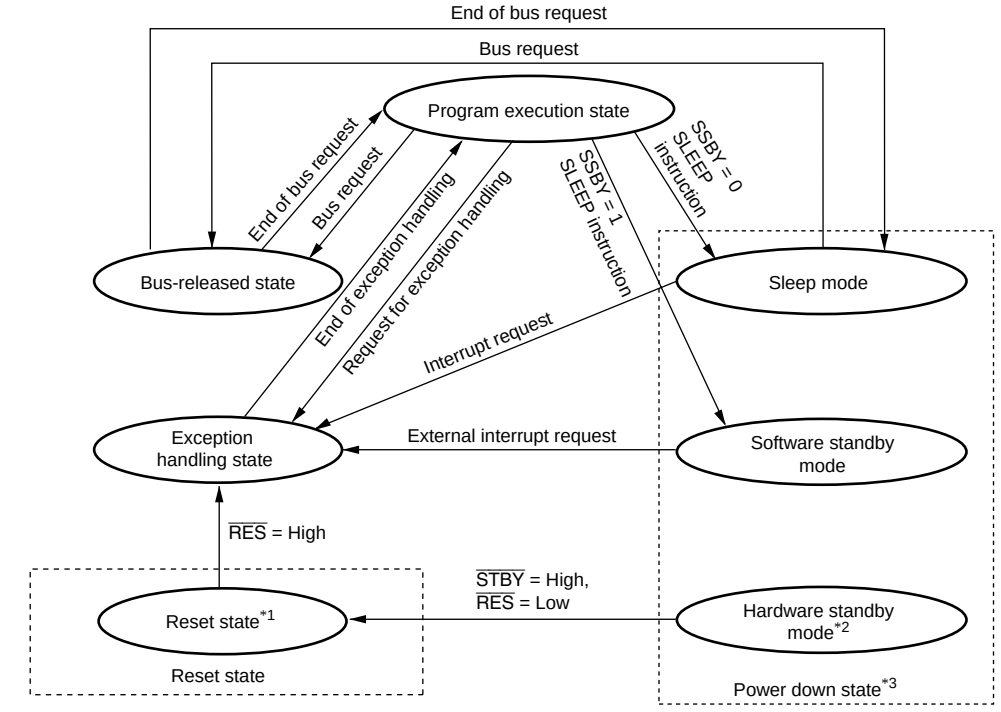
  In this state the CPU executes program instructions in sequence.

- Bus-Released State

  In a product which has a bus master other than the CPU, such as a direct memory access controller (DMAC) and 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.

- Program stop 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 19, 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. In every state, when the $\overline{STBY}$ pin becomes low, the hardware standby mode is entered.
3. For details, refer to section 19, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Note

2.9.1 Usage Notes on Bit-wise Operation 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 the flags in the internal I/O registers to 0. 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

The H8S/2668 Group has twelve operating modes (modes 1 to 7).

These modes are determined by the mode pin (MD2 to MD0) setting.

Modes 1, 2, and 4 to 6 are externally expanded modes in which the CPU can access an external memory and peripheral devices. In the externally expanded mode, each area can be switched to 8-bit or 16-bit address space by the bus controller. If one of areas is set to 16-bit address space, the bus mode is 16 bits. If all areas are set to 8-bit address space, the bus mode is 8 bits.

Mode 7 is a single-chip activation externally expanded mode in which the CPU can switch to access an external memory and peripheral devices at the beginning of a program execution.

Mode 3 is a boot mode in which the flash memory can be accessed.

For details, refer to section 17, Flash Memory (F-ZTAT version).

Do not change MD2 to MD0 pin settings during operation.

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. Value
1	0	0	1	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
2	0	1	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
3	0	1	1	Advanced	Boot mode	Enabled	—	16 bits
4	1	0	0	Advanced	Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
5	1	0	1	Advanced	Expanded mode with on-chip ROM enabled	Enabled	16 bits	16 bits
6	1	1	0	Advanced	Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7	1	1	1	Advanced	Single-chip mode	Enabled	—	16 bits

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 of the H8S/2668 Group chip.

Bit	Bit Name	Initial Value	R/W	Descriptions
7 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 pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits and they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a reset.
1	MDS1	—*	R	
0	MDS0	—*	R	

Note: * Determined by pins MD2 to MD0.

3.2.2 System Control Register (SYSCR)

SYSCR selects saturating or non-saturating calculation for the MAC instruction, controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2), sets external bus mode, and enables or disables on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Descriptions
7, 6	—	All 1	R/W	Reserved The initial value should not be modified.
5	MACS	0	R/W	MAC Saturation Selects either saturating or non-saturating calculation for the MAC instruction. 0: Non-saturating calculation for MAC instruction 1: Saturating calculation for MAC instruction

Bit	Bit Name	Initial Value	R/W	Descriptions
4	—	0	R/W	Reserved The initial value should not be modified.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). If this bit is set to 1, the flash memory control registers can be read/written to. If this bit is cleared to 0, the flash memory control registers are not selected. At this time, the contents of the flash memory control registers are maintained. 0: Flash memory control registers are not selected for area H'FFFFC8 to H'FFFFCB 1: Flash memory control registers are selected for area H'FFFFC8 to H'FFFFCB
2	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
1	EXPE	—	R/W	External Bus Mode Enable Sets external bus mode. In modes 1, 2, and 4 to 6, this bit is fixed at 1 and cannot be modified. In mode 3 and 7, this bit has an initial value of 0, and can be read and written. Writing of 0 to EXPE when its value is 1 should only be carried out when an external bus cycle is not being executed. 0: External bus disabled 1: External bus enabled
0	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

3.3 Operating Mode Descriptions

3.3.1 Mode 1

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports A to C function as an address bus, ports D and E function as a data bus, and parts of ports F to H carry bus control signals.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, if 8-bit access is designated for all areas by the bus controller, the bus mode switches to 8 bits.

3.3.2 Mode 2

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports A to C function as an address bus, ports D and E function as a data bus, and parts of ports F to H carry bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for all areas by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus.

3.3.3 Mode 3

This mode is a boot mode of the flash memory. This mode is the same as mode 7, except for accessing to the flash memory.

3.3.4 Mode 4

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The program in the on-chip ROM connected to the first half of area 0 is executed.

Ports A to C function as input ports immediately after a reset, but can be set to function as an address bus. For details, see section 8, I/O Ports. Ports D and E function as a data bus, and parts of ports F to H carry bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for any area by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus. User program mode is entered by setting 1 to the SWE bit of FLMCR1.

3.3.5 Mode 5

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The program in an external ROM connected to the first half of area 0 is executed.

Ports A to C function as an address bus, ports D and E function as a data bus, and parts of ports F to H carry bus control signals.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, if 8-bit access is designated for any area by the bus controller, the bus mode switches to 8 bits.

User program mode is entered by setting 1 to the SWE bit of FLMCR1.

3.3.6 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The program in an external ROM connected to the first half of area 0 is executed.

Ports A to C function as an address bus, ports D and E function as a data bus, and parts of ports F to H carry bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for any area by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus.

User program mode is entered by setting 1 to the SWE bit of FLMCR1.

3.3.7 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, and the chip starts up in single-chip mode. External addresses cannot be used in single-chip mode.

The initial mode after a reset is single-chip mode, with all I/O ports available for use as input/output ports. However, the mode can be switched to externally expanded mode by setting 1 to the EXPE bit of SYSCR and then the external address space is enabled. When externally expanded mode is selected, all areas are initially designated as 16-bit access space. The function of pins in ports A to H is the same as in externally expanded mode with on-chip ROM enabled.

In the flash memory version, user program mode is entered by setting 1 to the SWE bit of FLMCR1.

3.3.8 Pin Functions

The pin functions of ports A to H are switched according to operating mode. Table 3.2 shows the pin functions in each operating mode.

Table 3.2 Pin Functions in Each Operating Mode

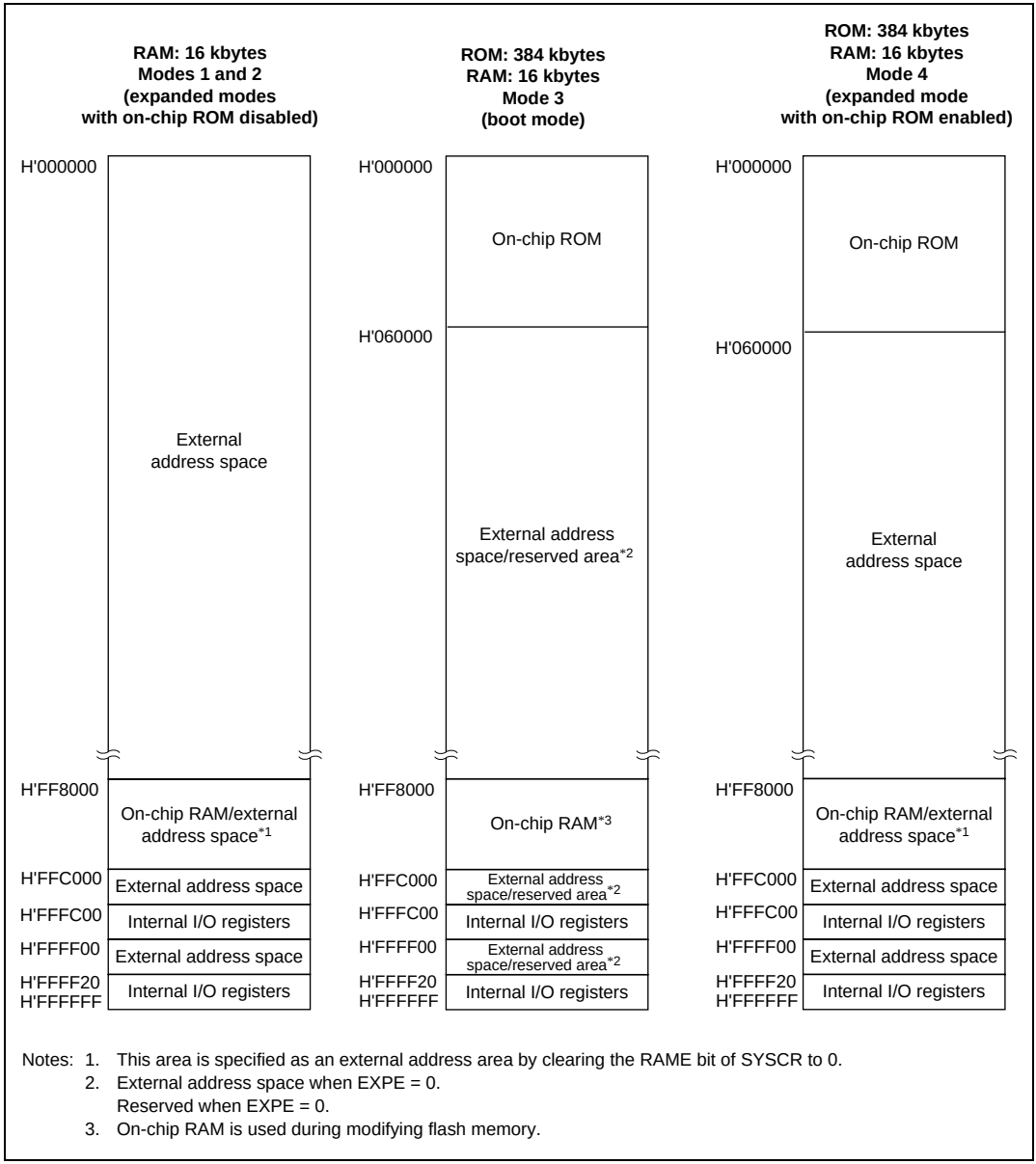
Port		Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port A	PA7 to PA5	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A	P*/A
	PA4 to PA0	A	A			A	A	
Port B		A	A	P*/A	P*/A	A	A	P*/A
Port C		A	A	P*/A	P*/A	A	A	P*/A
Port D		D	D	P*/D	P*/D	D	D	P*/D
Port E		P/D*	P*/D	P*/D	P*/D	P/D*	P*/D	P*/D
Port F	PF7, PF6	P/C*	P*/C	P*/C	P/C*	P/C*	P/C*	P*/C
	PF5, PF4	C	C		C	C	C	
	PF3	P/C*	P/C*		P/C*	P/C*	P/C*	
	PF2 to PF0	P*/C	P*/C		P*/C	P*/C	P*/C	
Port G	PG6 to PG1	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C
	PG0	P/C*	P/C*		P*/C	P/C*	P/C*	
Port H		P*/C	P*/C	P*/C	P*/C	P*/C	P*/C	P*/C

Legend: P: I/O port
A: Address bus output
D: Data bus input/output
C: Control signals, clock input/output

Note: * After reset

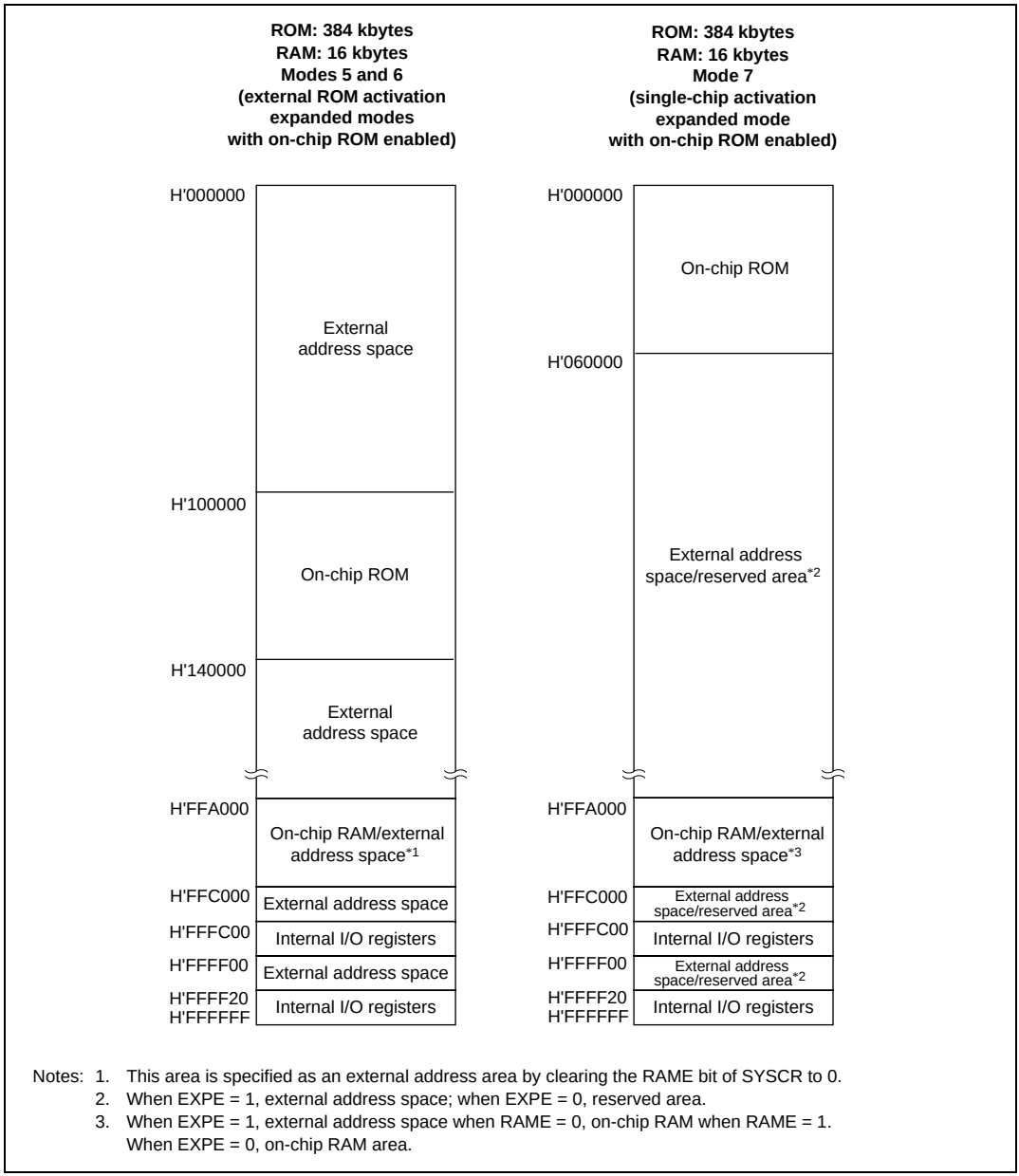
3.4 Memory Map in Each Operating Mode

Figure 3.1 shows memory maps.



Notes: 1. This area is specified as an external address area by clearing the RAME bit of SYSCR to 0.
2. External address space when EXPE = 0.
Reserved when EXPE = 0.
3. On-chip RAM is used during modifying flash memory.

Figure 3.1 Memory Map (1)



Notes: 1. This area is specified as an external address area by clearing the RAME bit of SYSCR to 0.
2. When EXPE = 1, external address space; when EXPE = 0, reserved area.
3. When EXPE = 1, external address space when RAME = 0, on-chip RAM when RAME = 1. When EXPE = 0, on-chip RAM area.

Figure 3.1 Memory Map (2)

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, interrupt, or trap instruction. 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. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, refer to section 5, Interrupt Controller.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ▲	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low.
	Trace*1	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit in the EXR is set to 1.
	Direct transition*2	Starts when the direct transition occurs by execution of the SLEEP instruction.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued.*3
Low	Trap instruction*4	Started by execution of a trap instruction (TRAPA)

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.
2. Not available in this LSI.
3. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
4. Trap instruction exception handling requests are accepted at all times in program execution state.

4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses. Since the usable modes differ depending on the product, for details on each product, refer to section 3, MCU Operating Modes.

Table 4.2 Exception Handling Vector Table

Exception Source		Vector Number	Vector Address*1	
			Normal Mode*2	Advanced Mode
Power-on reset		0	H'0000 to H'0001	H'0000 to H'0003
Manual reset*2		1	H'0002 to H'0003	H'0004 to H'0007
Reserved for system use		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0019	H'0010 to H'0013
Trace		5	H'000A to H'000B	H'0014 to H'0017
Interrupt (direct transition)*2		6	H'000C to H'000D	H'0018 to H'001B
Interrupt (NMI)		7	H'000E to H'000F	H'001C to H'001F
Trap instruction (#0)		8	H'0010 to H'0011	H'0020 to H'0023
(#1)		9	H'0012 to H'0013	H'0024 to H'0027
(#2)		10	H'0014 to H'0015	H'0028 to H'002B
(#3)		11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
	IRQ6	22	H'002C to H'002D	H'0058 to H'005B
	IRQ7	23	H'002E to H'002F	H'005C to H'005F

Exception Source	Vector Number	Vector Address*1 Normal Mode*2	Vector Address*1 Advanced Mode
Reserved by system	24	H'0030 to H'0031	H'0060 to H'0063
	25	H'0032 to H'0033	H'0064 to H'0067
	26	H'0034 to H'0035	H'0068 to H'006B
	27	H'0036 to H'0037	H'006C to H'006F
	28	H'0038 to H'0039	H'0070 to H'0073
	29	H'003A to H'003B	H'0074 to H'0077
	30	H'003C to H'003D	H'0078 to H'007B
	31	H'003E to H'003F	H'007C to H'007F
Internal interrupt*3	32	H'0040 to H'0041	H'0080 to H'0083
	⎮	⎮	⎮
	99	H'00C6 to H'00C7	H'018C to H'018F

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.

4.3 Reset

A reset has the highest exception priority. When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules.

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

The interrupt control mode is 0 immediately after reset.

4.3.1 Reset exception handling

When the $\overline{\text{RES}}$ 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 is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
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 and 4.2 show examples of the reset sequence.

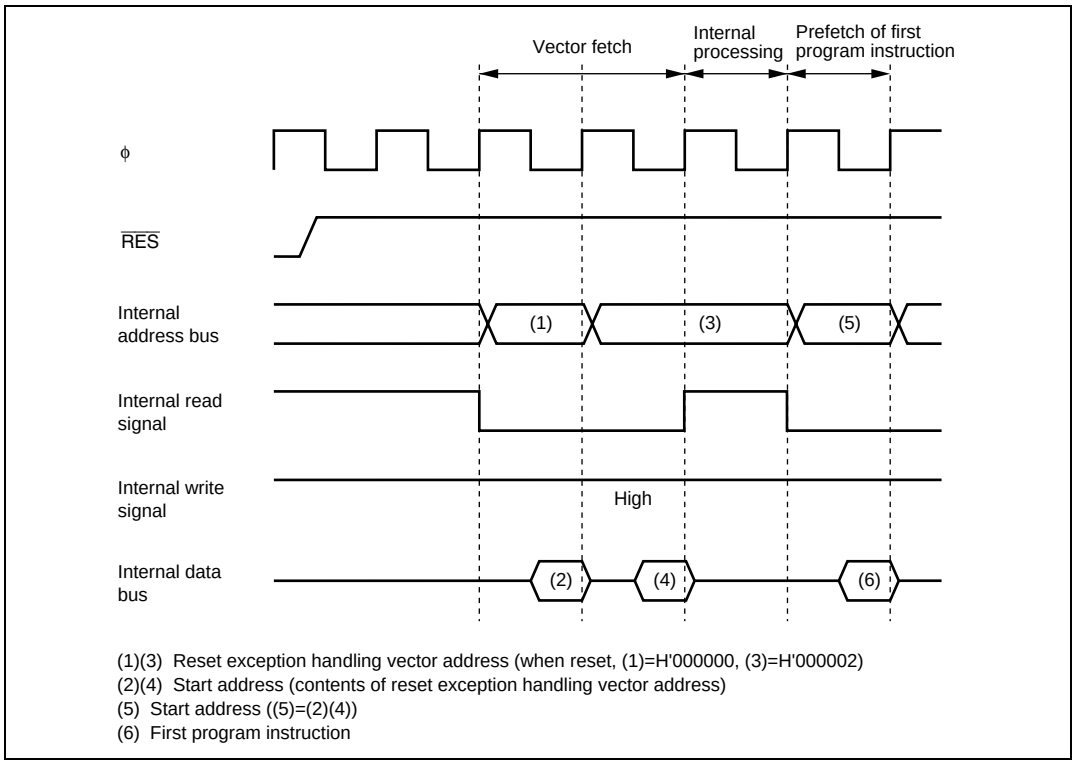
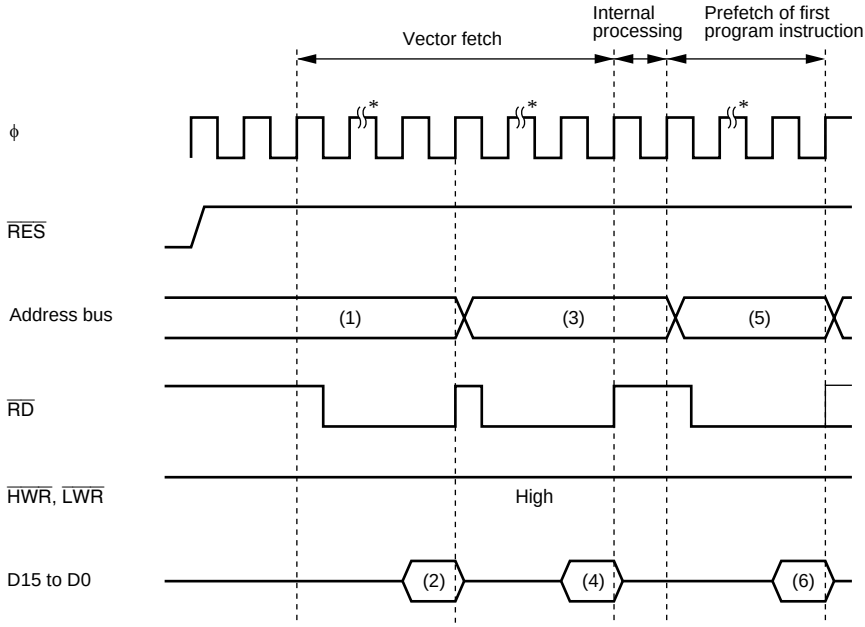


Figure 4.1 Reset Sequence (Advanced Mode with On-chip ROM Enabled)



(1)(3) Reset exception handling vector address (when 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 program instruction

Note: * Seven program wait states are inserted.

Figure 4.2 Reset Sequence (Advanced Mode with On-chip ROM Disabled)

4.3.2 Interrupts after Reset

If an interrupt is accepted after a reset but 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: 32, SP).

4.3.3 On-Chip Peripheral Functions after Reset Release

After reset release, MSTPCR is initialized to H'0FFF 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 module stop mode is exited.

4.4 Traces

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

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction. Trace mode is not affected by interrupt masking. Table 4.3 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. 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.

Interrupts are accepted even within the trace exception handling routine.

Table 4.3 Status of CCR and EXR after Trace Exception Handling

Interrupt Control Mode	CCR		EXR	
	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 Interrupts

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. The source to start interrupt exception handling and the vector address differ depending on the product. For details, refer to section 5, Interrupt Controller.

The interrupt exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in 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.

4.6 Trap Instruction

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.

The trap instruction exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in 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 4.4 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4.4 Status of CCR and EXR after Trap Instruction Exception Handling

Interrupt Control Mode	CCR		EXR	
	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 Status after Exception Handling

Figure 4.3 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

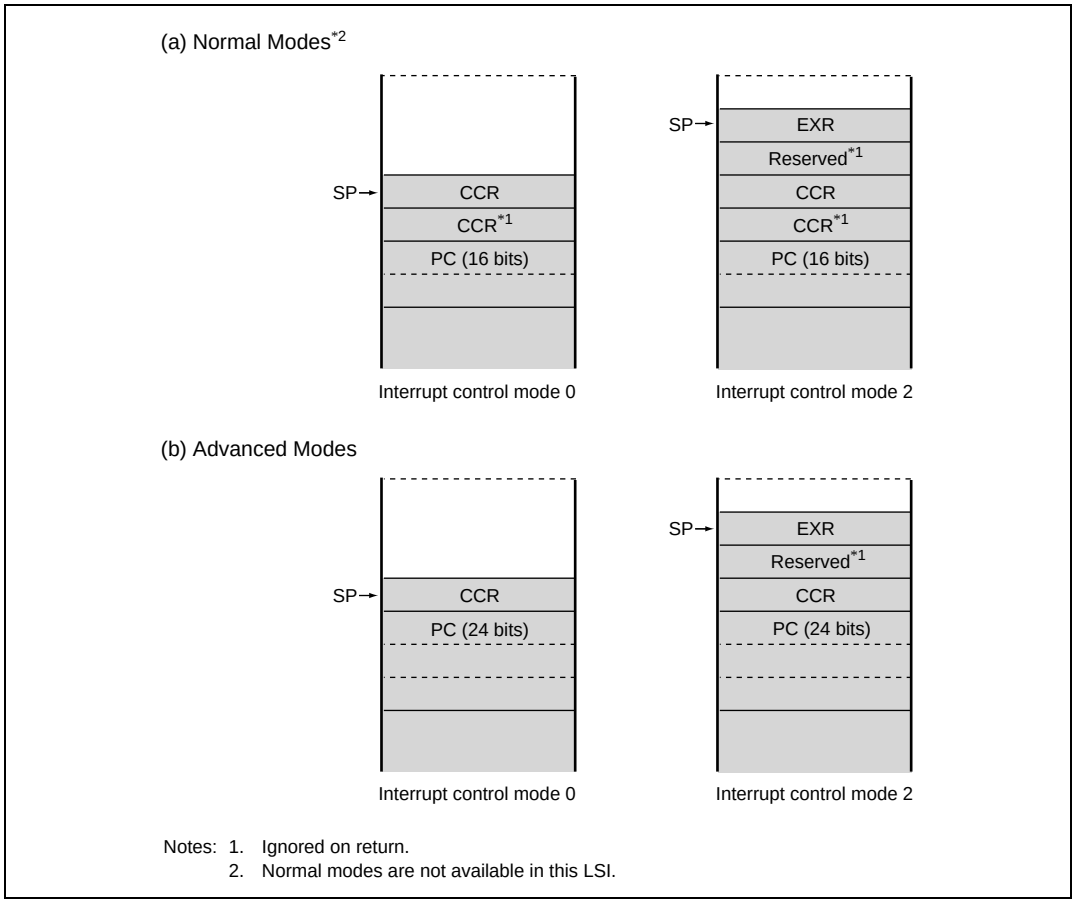


Figure 4.3 Stack Status after Exception Handling

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 SP to an odd value may lead to a malfunction. Figure 4.4 shows an example of operation when the SP value is odd.

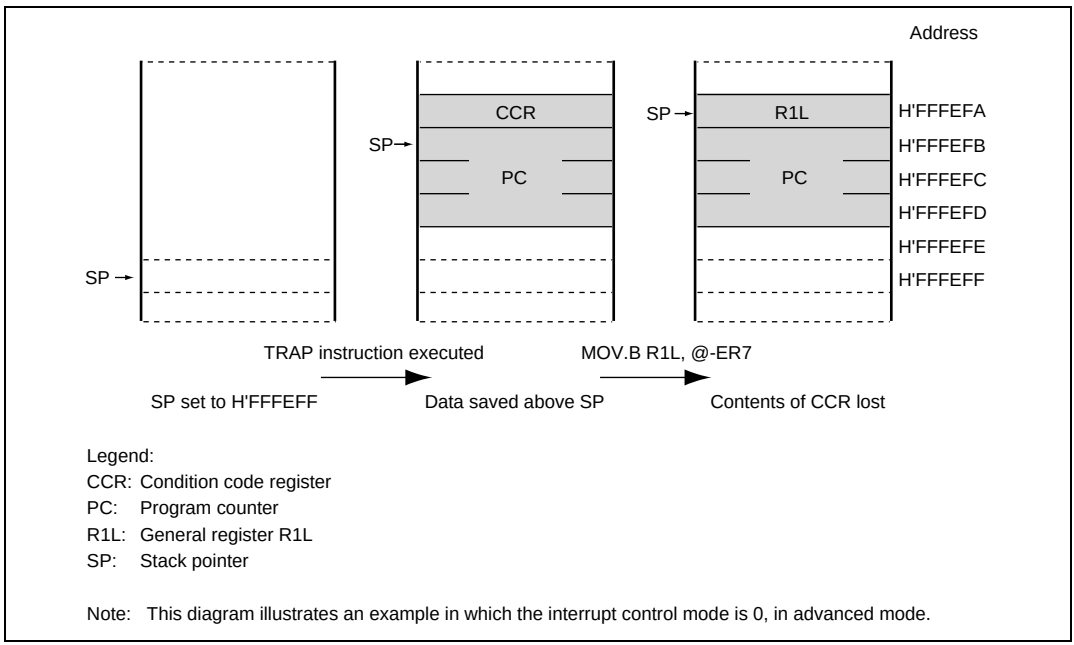


Figure 4.4 Operation when SP Value Is Odd

Section 5 Interrupt Controller

5.1 Features

- Two interrupt control modes

  Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the interrupt control register (INTCR).
- 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 interrupts

  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, or both edge detection, or level sensing, can be selected for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- DTC control

  DTC activations are performed by means of interrupts.

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

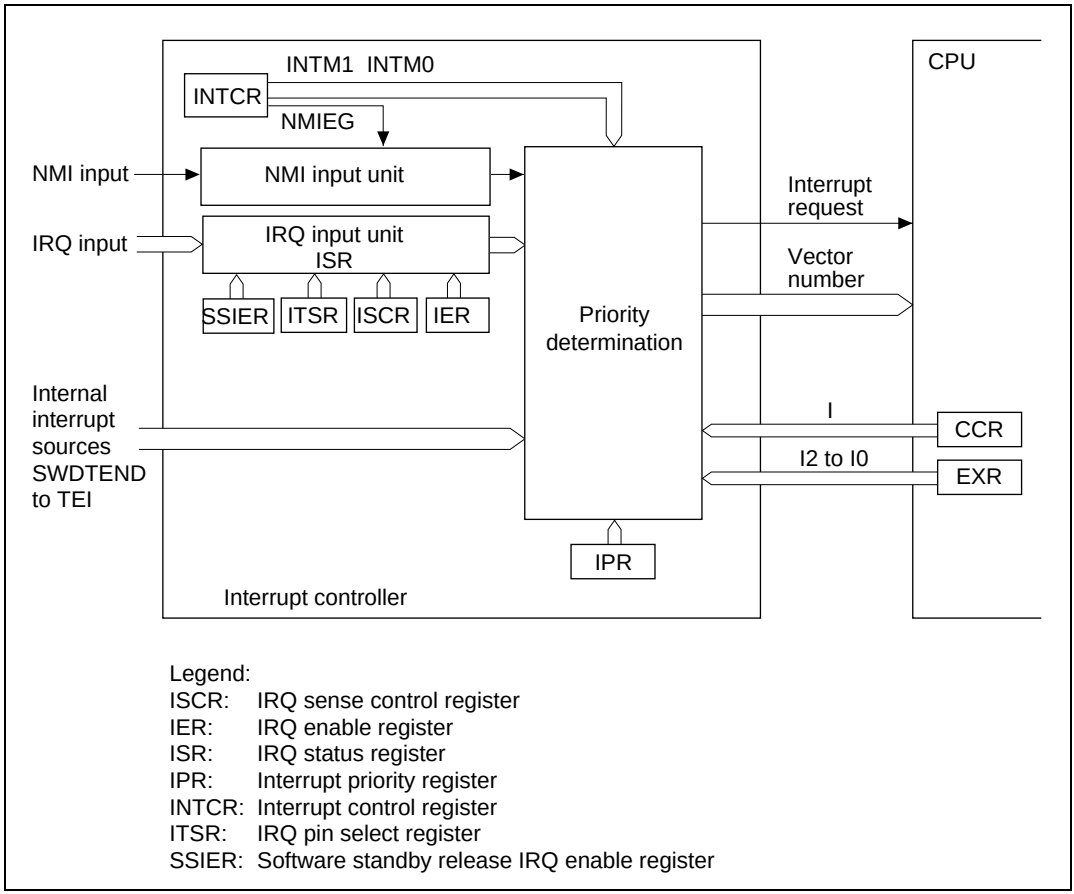


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 shows the pin configuration of the interrupt controller.

Table 5.1 Pin Configuration

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

5.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- IRQ pin select register (ITSR)
- Software standby release IRQ enable register (SSIER)
- 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)

5.3.1 Interrupt Control Register (INTCR)

INTCR selects the interrupt control mode, and the detected edge for NMI.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.
5	INTM1	0	R/W	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select either of two interrupt control modes for the interrupt controller. 00: Interrupt control mode 0 Interrupts are controlled by I bit. 01: Setting prohibited. 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0, and IPR. 11: Setting prohibited.
3	NMIEG	0	R/W	NMI Edge Select Selects the input edge for the NMI pin. 0: Interrupt request generated at falling edge of NMI input 1: Interrupt request generated at rising edge of NMI input
2 to 0	—	All 0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.

5.3.2 Interrupt Priority Registers A to K (IPRA to IPRK)

IPR are eleven 16-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 table 5.2 (Interrupt Sources, Vector Addresses, and Interrupt Priorities). Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 14 to 12, 10 to 8, 6 to 4, and 2 to 0 sets the priority of the corresponding interrupt. IPR should be read in word size.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
14	IPR14	1	R/W	Sets 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)
13	IPR13	1	R/W	
12	IPR12	1	R/W	
11	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
10	IPR10	1	R/W	Sets 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)
9	IPR9	1	R/W	
8	IPR8	1	R/W	

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	Sets 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	Sets 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	

5.3.3 IRQ Enable Register (IER)

IER controls enabling and disabling of interrupt requests IRQ7 to IRQ0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	IRQ15E	0	R/W	Reserved These bits can be read and modified. The write value should always be 0.
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.

5.3.4 IRQ Sense Control Registers (ISCR)

ISCR select the source that generates an interrupt request at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

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 generated at $\overline{\text{IRQ7}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ input
13	IRQ6SCB	0	R/W	IRQ6 Sense Control B
12	IRQ6SCA	0	R/W	IRQ6 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ6}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ6}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ6}}$ input 11: Interrupt request 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 generated at $\overline{\text{IRQ5}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input

Bit	Bit Name	Initial Value	R/W	Description
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ4}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ4}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ4}}$ input 11: Interrupt request 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 generated at $\overline{\text{IRQ3}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ3}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ3}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ2}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ2}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ2}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input

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

5.3.5 IRQ Status Register (ISR)

ISR is an IRQ7 to IRQ0 interrupt request flag register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
7	IRQ7F	0	R/(W)*	[Setting conditions] When the interrupt source selected by ISCR occurs [Clearing conditions] • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0 (n=15 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 clear the flag.

5.3.6 IRQ Pin Select Register (ITSR)

ITSR selects input pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R/W	Reserved These bits can be read and modified. The write value should always be 0.
7	ITS7	0	R/W	Selects $\overline{\text{IRQ7}}$ input pin. 0: P57 1: PH3
6	ITS6	0	R/W	Selects $\overline{\text{IRQ6}}$ input pin. 0: P56 1: PH2
5	ITS5	0	R/W	Selects $\overline{\text{IRQ5}}$ input pin. 0: P55 1: P85
4	ITS4	0	R/W	Selects $\overline{\text{IRQ4}}$ input pin. 0: P54 1: P84
3	ITS3	0	R/W	Selects $\overline{\text{IRQ3}}$ input pin. 0: P53 1: P83
2	ITS2	0	R/W	Selects $\overline{\text{IRQ2}}$ input pin. 0: P52 1: P82
1	ITS1	0	R/W	Selects $\overline{\text{IRQ1}}$ input pin. 0: P51 1: P81
0	ITS0	0	R/W	Selects $\overline{\text{IRQ0}}$ input pin. 0: P50 1: P80

5.3.7 Software Standby Release IRQ Enable Register (SSIER)

SSIER selects the $\overline{\text{IRQ}}$ pins used to recover from the software standby state.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R/W	Reserved These bits can be read and modified. The write value should always be 0.
7	SSI7	0	R/W	Software Standby Release IRQ Setting These bits select the $\overline{\text{IRQ}}$n pins used to recover from the software standby state. 0: IRQn requests are not sampled in the software standby state (Initial value when n = 7 to 3) 1: When an IRQn request occurs in the software standby state, the chip recovers from the software standby state after the elapse of the oscillation settling time (Initial value when n = 2 to 0) (n = 15 to 0)
6	SSI6	0	R/W	
5	SSI5	0	R/W	
4	SSI4	0	R/W	
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	
0	SSI0	0	R/W	

5.4 Interrupt Sources

5.4.1 External Interrupts

There are nine external interrupts: NMI and IRQ7 to IRQ0. These interrupts can be used to restore the chip from software standby mode.

NMI Interrupt: Nonmaskable interrupt request (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 INTCR 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: Interrupts IRQ7 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. Interrupts IRQ7 to IRQ0 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 pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- Enabling or disabling of interrupt requests IRQ7 to IRQ0 can be selected with IER.
- The interrupt priority level can be set with IPR.

- The status of interrupt requests IRQ7 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

When IRQ7 to IRQ0 interrupt requests occur at low level of $\overline{\text{IRQn}}$, the corresponding $\overline{\text{IRQ}}$ should be held low until an interrupt handling starts. Then the corresponding $\overline{\text{IRQ}}$ should be set to high in the interrupt handling routine and clear the IRQnF bit (n = 0 to 7) in ISR to 0. Interrupts may not be executed when the corresponding $\overline{\text{IRQ}}$ is set to high before the interrupt handling starts.

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 and use the pin as an I/O pin for another function.

A block diagram of interrupts IRQ15 to IRQ0 is shown in figure 5.2.

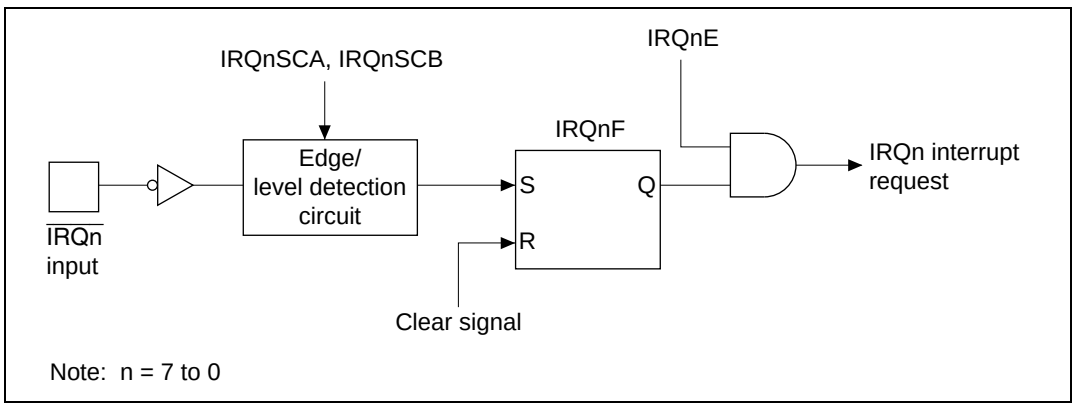


Figure 5.2 Block Diagram of Interrupts IRQ7 to IRQ0

5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

- 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. They can be controlled independently. When the enable bit is set to 1, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DTC can be activated by a TPU, SCI, or other interrupt request.
- When the DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

5.5 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. When interrupt control mode 2 is set, 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

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority	DTC Activation
			Advanced Mode			
External pin	NMI	7	H'001C	—	High	—
	IRQ0	16	H'0040	IPRA14 to IPRA12	↑	○
	IRQ1	17	H'0044	IPRA10 to IPRA8		○
	IRQ2	18	H'0048	IPRA6 to IPRA4		○
	IRQ3	19	H'004C	IPRA2 to IPRA0		○
	IRQ4	20	H'0050	IPRB14 to IPRB12		○
	IRQ5	21	H'0054	IPRB10 to IPRB8		○
	IRQ6	22	H'0058	IPRB6 to IPRB4		○
	IRQ7	23	H'005C	IPRB2 to IPRB0		○
Reserved by system		24	H'0060	IPRC14 to IPRC12		—
		25	H'0064	IPRC10 to IPRC8		—
		26	H'0068	IPRC6 to IPRC4		—
		27	H'006C	IPRC2 to IPRC0		—
		28	H'0070	IPRD14 to IPRD12		—
		29	H'0074	IPRD10 to IPRD8		—
		30	H'0078	IPRD6 to IPRD4		—
		31	H'007C	IPRD2 to IPRD0		—
DTC	SWDTEND	32	H'0080	IPRE14 to IPRE12		○
WDT	WOVI	33	H'0084	IPRE10 to IPRE8		—
—	Reserved for system use	34	H'0088	IPRE6 to IPRE4		—
		35	H'008C	IPRE2 to IPRE0		—
	Reserved for system use	36	H'0090	IPRF14 to IPRF12		—
		37	H'0094		Low	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority	DTC Activation
			Advanced Mode			
A/D	ADI	38	H'0098	IPRF10 to IPRF8	High	○
—	Reserved for system use	39	H'009C			—
TPU_0	TGI0A	40	H'00A0	IPRF6 to IPRF4		○
	TGI0B	41	H'00A4			○
	TGI0C	42	H'00A8			○
	TGI0D	43	H'00AC			○
	TCI0V	44	H'00B0			—
—	Reserved for system use	45	H'00B4	IPRF6 to IPRF4		—
		46	H'00B8			—
		47	H'00BC			—
TPU_1	TGI1A	48	H'00C0	IPRF2 to IPRF0		○
	TGI1B	49	H'00C4			○
	TCI1V	50	H'00C8			—
	TCI1U	51	H'00CC			—
TPU_2	TGI2A	52	H'00D0	IPRG14 to IPRG12		○
	TGI2B	53	H'00D4			○
	TCI2V	54	H'00D8			—
	TCI2U	55	H'00DC			—
TPU_3	TGI3A	56	H'00E0	IPRG10 to IPRG8		○
	TGI3B	57	H'00E4			○
	TGI3C	58	H'00E8			○
	TGI3D	59	H'00EC			○
	TCI3V	60	H'00F0			—
—	Reserved for system use	61	H'00F4			—
		62	H'00F8			—
		63	H'00FC			—
TPU_4	TGI4A	64	H'0100	IPRG6 to IPRG4		○
	TGI4B	65	H'0104			○
	TCI4V	66	H'0108			—
	TCI4U	67	H'010C		Low	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*			DTC Activation
			Advanced Mode	IPR	Priority	
TPU_5	TGI5A	68	H'0110	IPRG2 to IPRG0	High	○
	TGI5B	69	H'0114			○
	TCI5V	70	H'0118			—
	TCI5U	71	H'011C			—
TMR_0	CMIA0	72	H'0120	IPRH14 to IPRH12		○
	CMIB0	73	H'0124			○
	OVI0	74	H'0128			—
—	Reserved for system use	75	H'012C			—
TMR_1	CMIA1	76	H'0130	IPRH10 to IPRH8		○
	CMIB1	77	H'0134			○
	OVI1	78	H'0138			—
—	Reserved for system use	79	H'013C			—
		80	H'0140	IPRH6 to IPRH4		○
		81	H'0144			○
		82	H'0148			○
		83	H'014C			○
		84	H'0150	IPRH0 to IPRH0		—
		85	H'0154	IPRI14 to IPRI12		—
		86	H'0158	IPRI10 to IPRI8		—
		87	H'015C	IPRI6 to IPRI4		—
SCI_0	ERI0	88	H'0160	IPRI2 to IPRI0		—
	RXI0	89	H'0164			○
	TXI0	90	H'0168			○
	TEI0	91	H'016C			—
SCI_1	ERI1	92	H'0170	IPRJ14 to IPRJ12		—
	RXI1	93	H'0174			○
	TXI1	94	H'0178			○
	TEI1	95	H'017C			—
SCI_2	ERI2	96	H'0180	IPRJ10 to IPRJ8		—
	RXI2	97	H'0184			○
	TXI2	98	H'0188			○
	TEI2	99	H'018C		Low	—

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority	DTC Activation
—	Reserved for system use	100	H'0190	IPRJ6 to IPRJ4	High	—
		101	H'0194			—
		102	H'0198			—
		103	H'019C			—
		104	H'01A0	IPRJ2 to IPRJ0		—
		105	H'01A4			—
		106	H'01A8			—
		107	H'01AC			—
		108	H'01B0	IPRK14 to IPRK12		—
		109	H'01B4			—
		110	H'01B8			—
		111	H'01BC			—
		112	H'01C0	IPRK10 to IPRK8		—
		113	H'01C4			—
		114	H'01C8			—
		115	H'01CC			—
		116	H'01D0	IPRK6 to IPRK4		—
		117	H'01D4			—
		118	H'01D8			—
		119	H'01DC			—
		120	H'01E0	IPRK2 to IPRK0		—
		121	H'01E4			—
		122	H'01E8			—
		123	H'01EC			—
		124	H'01F0			—
		125	H'01F4			—
		126	H'01F8			—
		127	H'01EC		Low	—

Note: * Lower 16 bits of the start address.

5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 2. Interrupt operations differ depending on the interrupt control mode. The interrupt control mode is selected by INTCR. Table 5.3 shows the differences between interrupt control mode 0 and interrupt control mode 2.

Table 5.3 Interrupt Control Modes

Interrupt Control Mode	Priority Setting Registers	Interrupt Mask Bits	Description
0	Default	I	The priorities of interrupt sources are fixed at the default settings. Interrupt sources except for NMI is masked by the I bit.
2	IPR	I2 to I0	8 priority levels except for NMI can be set with IPR. 8-level interrupt mask control is performed by bits I2 to I0.

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI is masked by the I bit of CCR in the CPU. Figure 5.3 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. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
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 in the vector table.

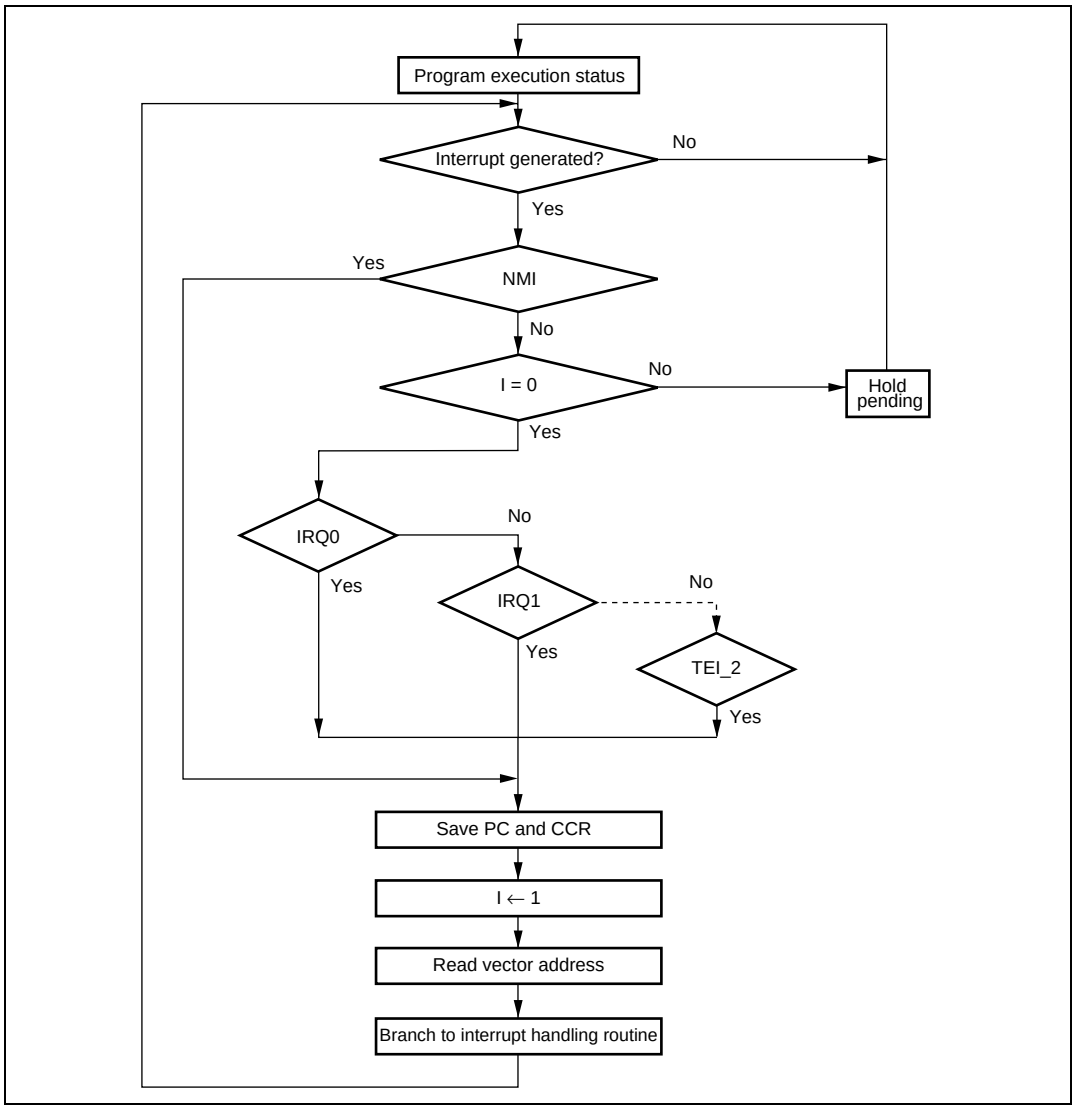


Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is done in eight levels for interrupt requests except for NMI by comparing the EXR interrupt mask level (I2 to I0 bits) in the CPU and the IPR setting. Figure 5.4 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 in the vector table.

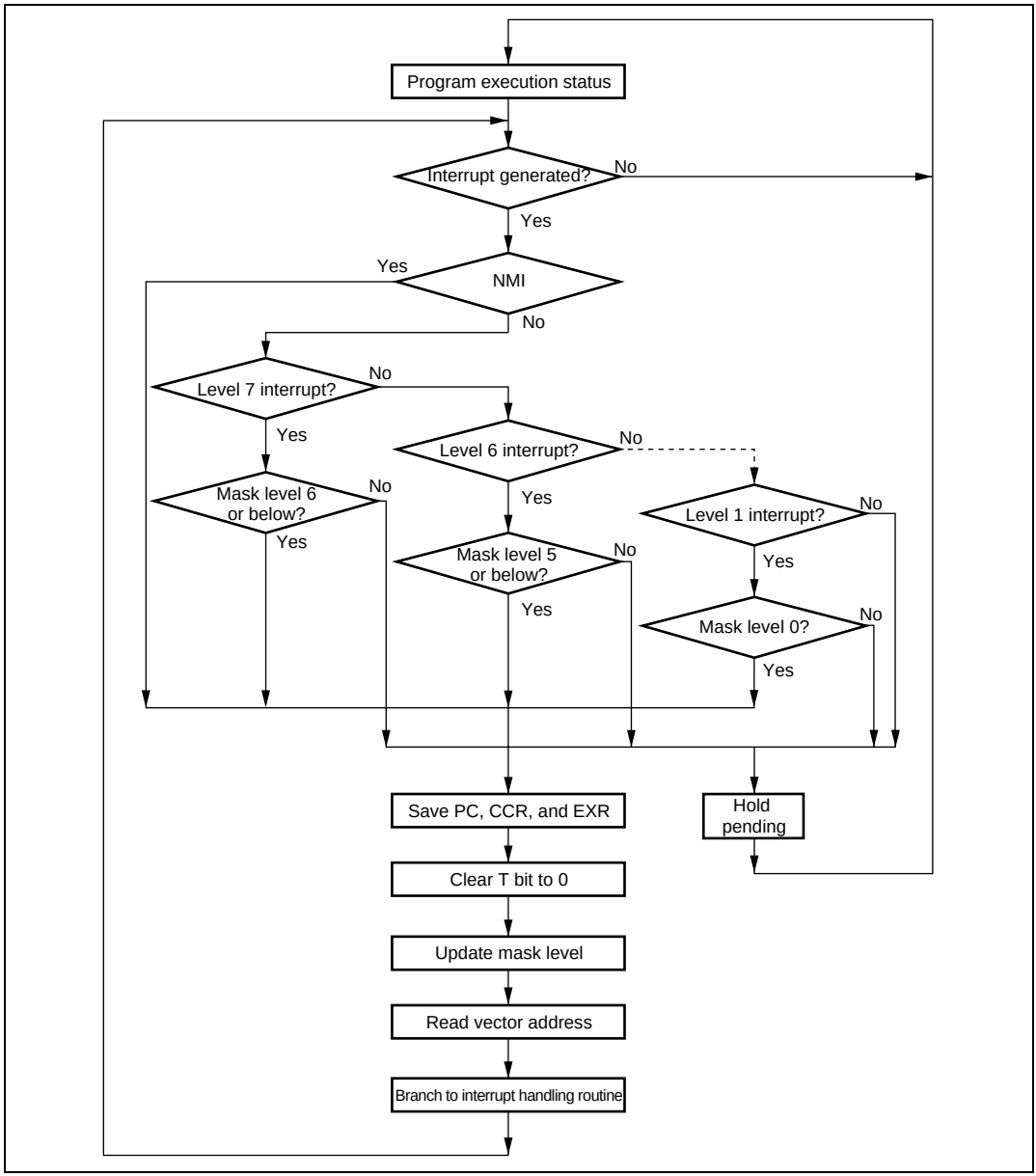


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.6.3 Interrupt Exception Handling Sequence

Figure 5.5 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.

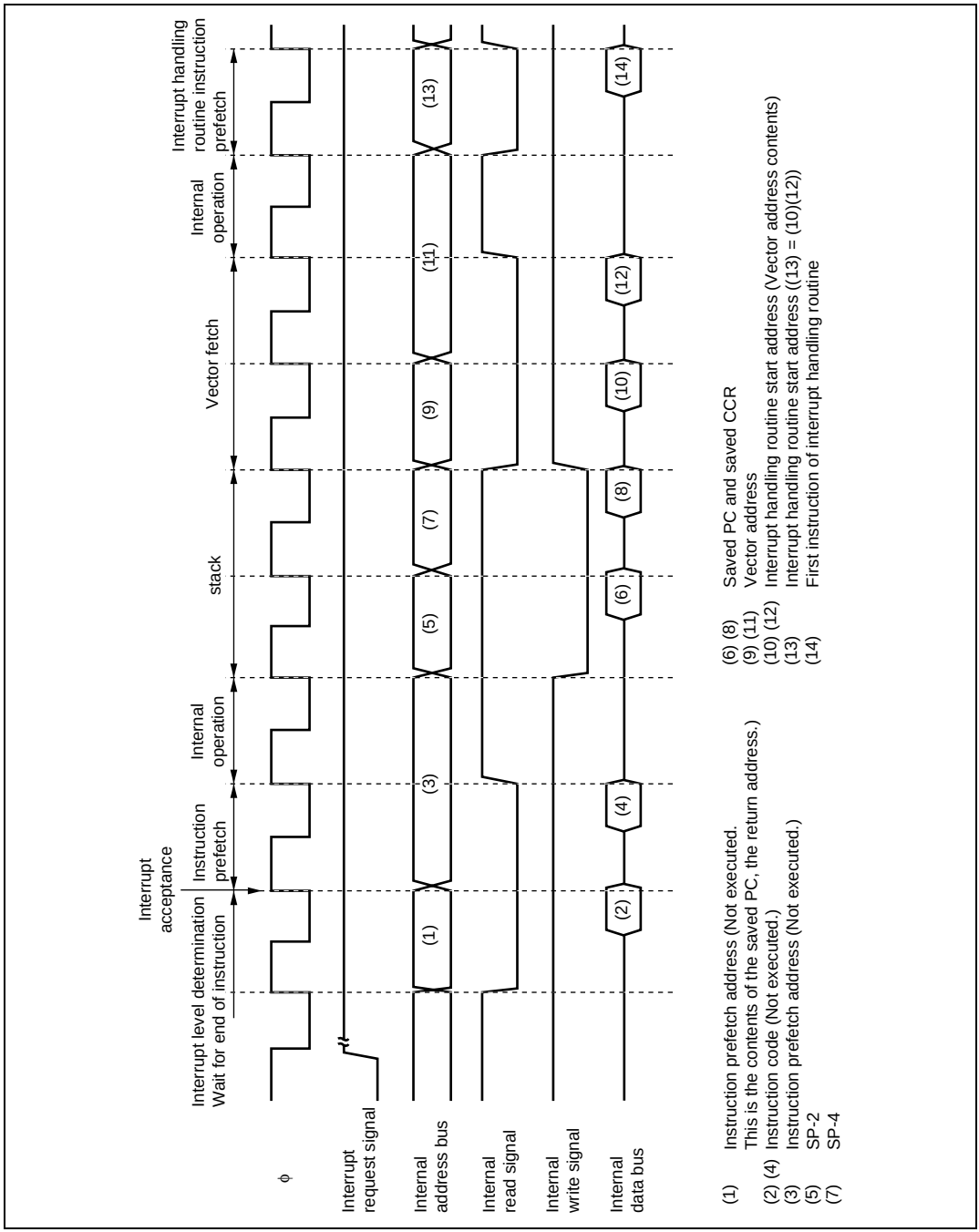


Figure 5.5 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.4 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 5.4 are explained in table 5.5.

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

Table 5.4 Interrupt Response Times

		Normal Mode*5		Advanced Mode	
No.	Execution Status	Interrupt control mode 0	Interrupt control mode 2	Interrupt control mode 0	Interrupt control mode 2
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 5.5 Number of States in Interrupt Handling Routine Execution Statuses

	Object of Access				
		External Device			
		8 Bit Bus		16 Bit Bus	
Symbol	Internal 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.

5.6.5 DTC Activation by Interrupt

The DTC can be activated by an interrupt. In this case, the following options are available:

- Interrupt request to CPU
- Activation request to DTC
- Selection of a number of the above

For details of interrupt requests that can be used to activate the DTC, see table 5.2 and section 7, Data Transfer Controller.

Figure 5.6 shows a block diagram of the DTC and interrupt controller.

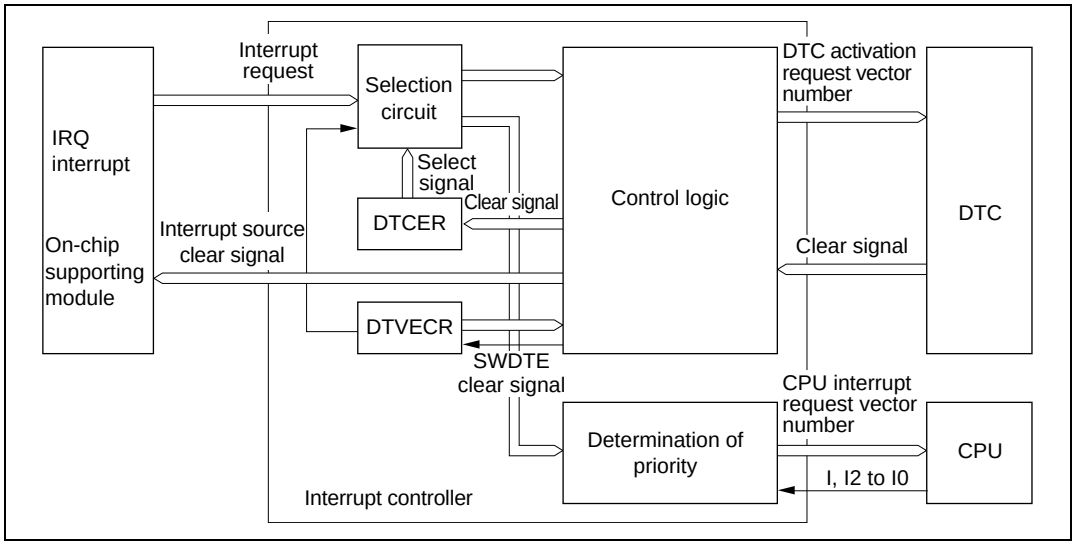


Figure 5.6 DTC and Interrupt Controller

(1) Selection of Interrupt Source: Interrupt factors are selected as DTC activation source or CPU interrupt source by the DTCE bit of DTCERA to DTCERF of DTC.

By specifying the DISEL bit of the DTC's MRB, it is possible to clear the DTCE bit to 0 after DTC data transfer, and request a CPU interrupt.

If DTC carries out the designate number of data transfers and the transfer counter reads 0, after DTC data transfer, the DTCE bit is also cleared to 0, and an interrupt is requested to the CPU.

(2) Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels.

(3) Operation Order: If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

Table 5.6 shows the interrupt factor clear control and selection of interrupt factors by specification of the DTCE bit of DTC's DTCERA to DTCERH, and the DISEL bit of DTC's MRB.

Table 5.6 Interrupt Source Selection and Clearing Control

Settings			
DTC		**Interrupt Sources Selection/Clearing Control**	
DTCE	**DISEL**	**DTC**	**CPU**
0	*	X	◎
1	0	◎	X
	1	○	◎
*	*	X	X

Legend:
◎: The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)
○: The relevant interrupt is used. The interrupt source is not cleared.
X : The relevant interrupt cannot be used.
* : Don't care

Note: The SCI or A/D converter interrupt source is cleared when the DTC reads or writes to the prescribed register, and is not dependent upon the DISEL bit.

5.7 Usage Notes

5.7.1 Contention between Interrupt Generation and Disabling

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

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, 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. Figure 5.7 shows an example in which the TCIEV bit in the TPU's TIER_0 register 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.

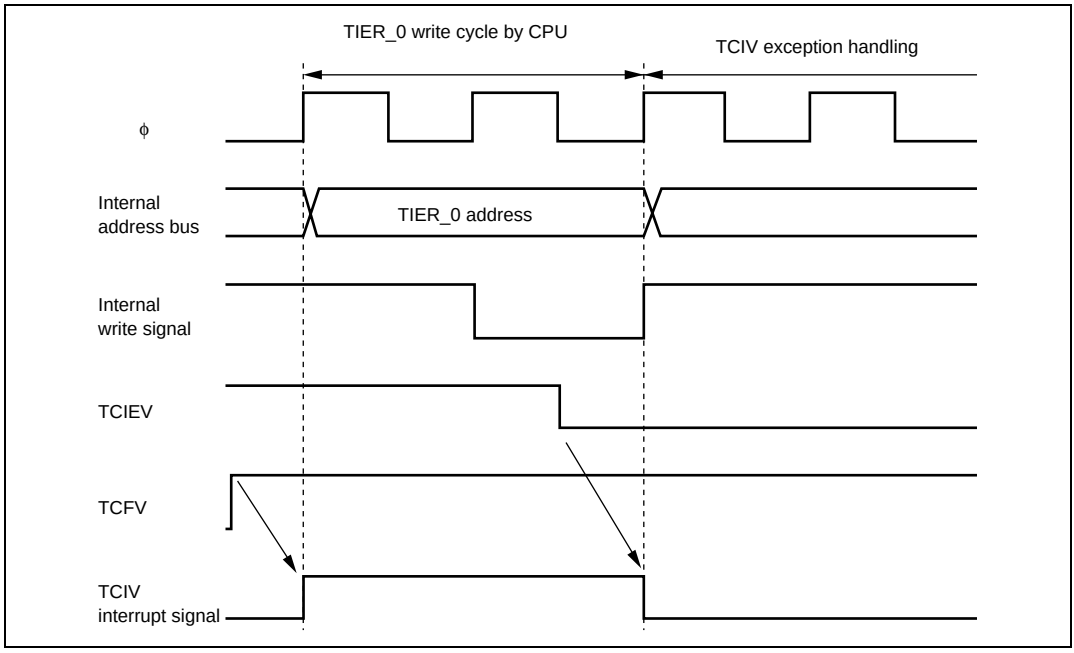


Figure 5.7 Contention between Interrupt Generation and Disabling

5.7.2 Instructions that Disable Interrupts

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is 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.

5.7.3 Times 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.

5.7.4 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
```

5.7.5 Change of IRQ Pin Select Register (ITSR) Setting

When the ITSR setting is changed, an edge occurs internally and the IRQnF bit (n = 0 to 7) of ISR may be set to 1 at the unintended timing if the selected pin level before the change is different from the selected pin level after the change. If the IRQn interrupt request (n = 0 to 7) is enabled, the interrupt exception handling is executed. To prevent the unintended interrupt, ITSR setting should be changed while the IRQn interrupt request is disabled, then the IRQnF bit should be cleared to 0.

5.7.6 Note on IRQ Status Register (ISR)

Since IRQnF flags may be set to 1 depending on the pin states after a reset, be sure to read from ISR after a reset and then write 0 to clear the IRQnF flags.

Section 6 Bus Controller (BSC)

This LSI has an on-chip bus controller (BSC) that manages the external address space divided into eight areas.

The bus controller also has a bus arbitration function, and controls the operation of the bus masterships—the CPU and data transfer controller (DTC).

6.1 Features

- Manages external address space in area units
  Manages the external address space divided into eight areas of 2 Mbytes
  Bus specifications can be set independently for each area
- Basic bus interface
  Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for areas 0 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
- Bus arbitration function
  Includes a bus arbiter that arbitrates bus mastershipship between the CPU and DTC

A block diagram of the bus controller is shown in figure 6.1.

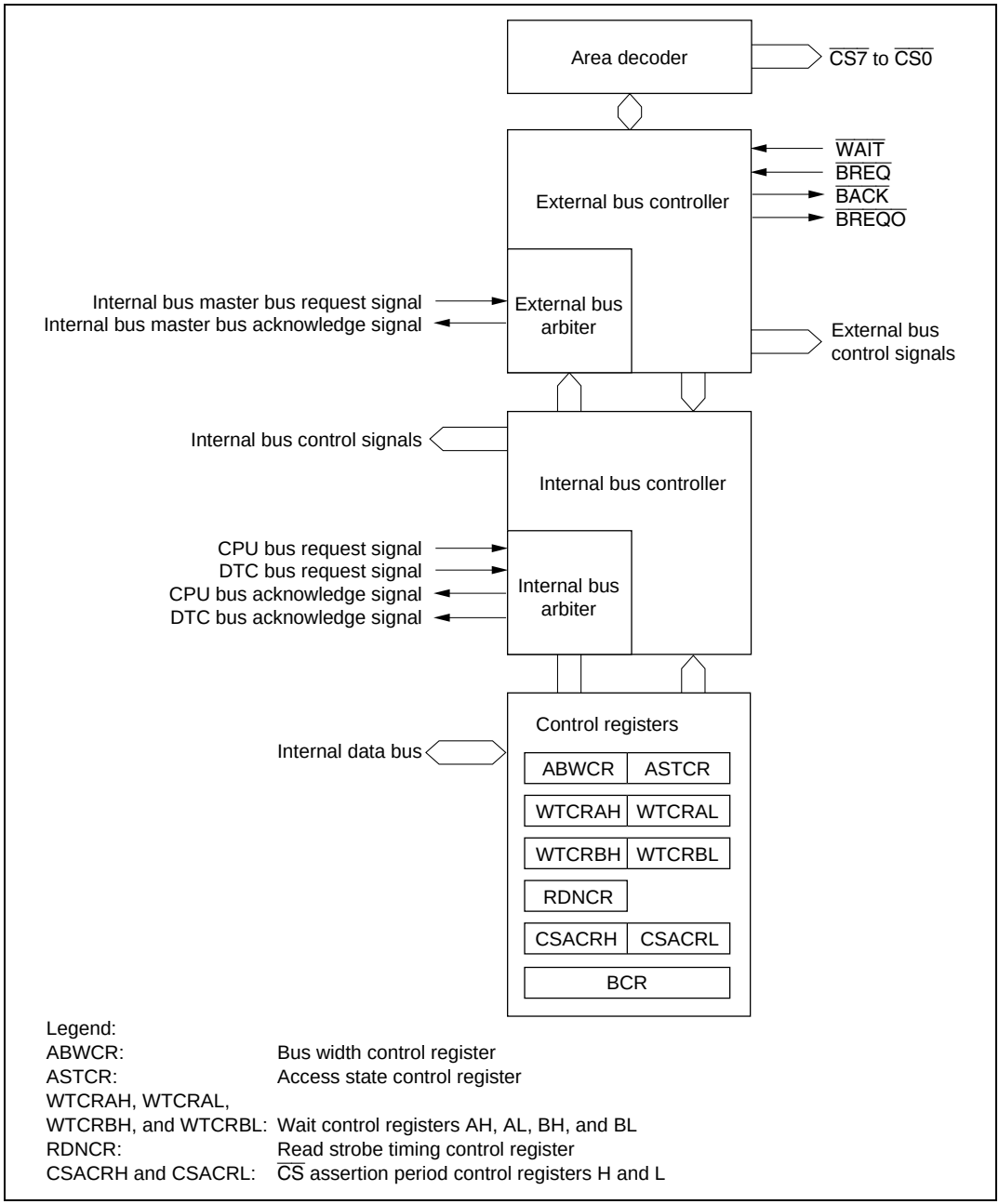


Figure 6.1 Block Diagram of Bus Controller

6.2 Input/Output Pins

Table 6.1 summarizes the pin configuration of the bus controller.

Table 6.1 Pin Configuration

Name	Symbol	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that an external address space is accessed and address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that an external address space is being read.
High write/write enable	$\overline{HWR}$	Output	Strobe signal indicating that an external address space is written to, and upper half (D15 to D8) of data bus is enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that an external address space is written to, and lower half (D7 to D0) of data bus is enabled.
Chip select 0	$\overline{CS0}$	Output	Strobe signal indicating that area 0 is selected.
Chip select 1	$\overline{CS1}$	Output	Strobe signal indicating that area 1 is selected
Chip select 2	$\overline{CS2}$	Output	Strobe signal indicating that area 2 is selected.
Chip select 3	$\overline{CS3}$	Output	Strobe signal indicating that area 3 is selected.
Chip select 4	$\overline{CS4}$	Output	Strobe signal indicating that area 4 is selected.
Chip select 5	$\overline{CS5}$	Output	Strobe signal indicating that area 5 is selected.
Chip select 6	$\overline{CS6}$	Output	Strobe signal indicating that area 6 is selected.
Chip select 7	$\overline{CS7}$	Output	Strobe signal indicating that area 7 is selected.
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external address space.
Bus request	$\overline{BREQ}$	Input	Request signal for release of bus to external bus master.
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has been released to external bus master.
Bus request output	$\overline{BREQO}$	Output	External bus request signal used when internal bus master accesses external address space when external bus is released.

6.3 Register Descriptions

The bus controller has the following registers.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register AH (WTCRAH)
- Wait control register AL (WTCRAL)
- Wait control register BH (WTCRBH)
- Wait control register BL (WTCRBL)
- Read strobe timing control register (RDNCR)
- $\overline{CS}$ assertion period control register H (CSACRH)
- $\overline{CS}$ assertion period control register L (CSACRL)
- Bus control register (BCR)

6.3.1 Bus Width Control Register (ABWCR)

ABWCR designates each area in the external address space as either 8-bit access space or 16-bit access space.

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

Note: * In modes 2, 4, and 6, ABWCR is initialized to 1. In modes 1, 5, and 7, ABWCR is initialized to 0.

6.3.2 Access State Control Register (ASTCR)

ASTCR designates each area in the external address space as either 2-state access space or 3-state access space.

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 2-state access space or 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	
2	AST2	1	R/W	0: Area n is designated as 2-state access space
1	AST1	1	R/W	Wait state insertion in area n access is disabled
0	AST0	1	R/W	1: Area n is designated as 3-state access space
				Wait state insertion in area n access is enabled
				(n = 7 to 0)

6.3.3 Wait Control Registers AH, AL, BH, and BL (WTCRAH, WTCRAL, WTCRBH, and WTCRBL)

WTCRA and WTCRB select the number of program wait states for each area in the external address space.

- WTCRAH

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
14 13 12	W72 W71 W70	1 1 1	R/W R/W R/W	Area 7 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 7 while AST7 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
11	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
10 9 8	W62 W61 W60	1 1 1	R/W R/W R/W	Area 6 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 6 while AST6 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted

- WT ARAL

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
6	W52	1	R/W	Area 5 Wait Control 2 to 0
5	W51	1	R/W	These bits select the number of program wait states when accessing area 5 while AST5 bit in ASTCR = 1.
4	W50	1	R/W	000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2	W42	1	R/W	Area 4 Wait Control 2 to 0
1	W41	1	R/W	These bits select the number of program wait states when accessing area 4 while AST4 bit in ASTCR = 1.
0	W40	1	R/W	000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted

- WTCRBH

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
14	W32	1	R/W	Area 3 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 3 while AST3 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
13	W31	1	R/W	
12	W30	1	R/W	
11	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
10	W22	1	R/W	Area 2 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 2 while AST2 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
9	W21	1	R/W	
8	W20	1	R/W	

- WTCRBL

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
6	W12	1	R/W	Area 1 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 1 while AST1 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
5	W11	1	R/W	
4	W10	1	R/W	
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2	W02	1	R/W	Area 0 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 0 while AST0 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
1	W01	1	R/W	
0	W00	1	R/W	

6.3.4 Read Strobe Timing Control Register (RDNCR)

RDNCR selects the read strobe signal ($\overline{RD}$) negation timing in an external address space read access.

Bit	Bit Name	Initial Value	R/W	Description
7	RDN7	0	R/W	Read Strobe Timing Control 7 to 0
6	RDN6	0	R/W	These bits set the negation timing of the read strobe in a corresponding area read access.
5	RDN5	0	R/W	
4	RDN4	0	R/W	As shown in figure 6.2, the read strobe for an area for which the RDNn bit is set to 1 is negated one half-state earlier than that for an area for which the RDNn bit is cleared to 0. The read data setup and hold time specifications are also one half-state earlier.
3	RDN3	0	R/W	
2	RDN2	0	R/W	
1	RDN1	0	R/W	
0	RDN0	0	R/W	0: In an area n read access, the $\overline{RD}$ is negated at the end of the read cycle 1: In an area n read access, the $\overline{RD}$ is negated one half-state before the end of the read cycle (n = 7 to 0)

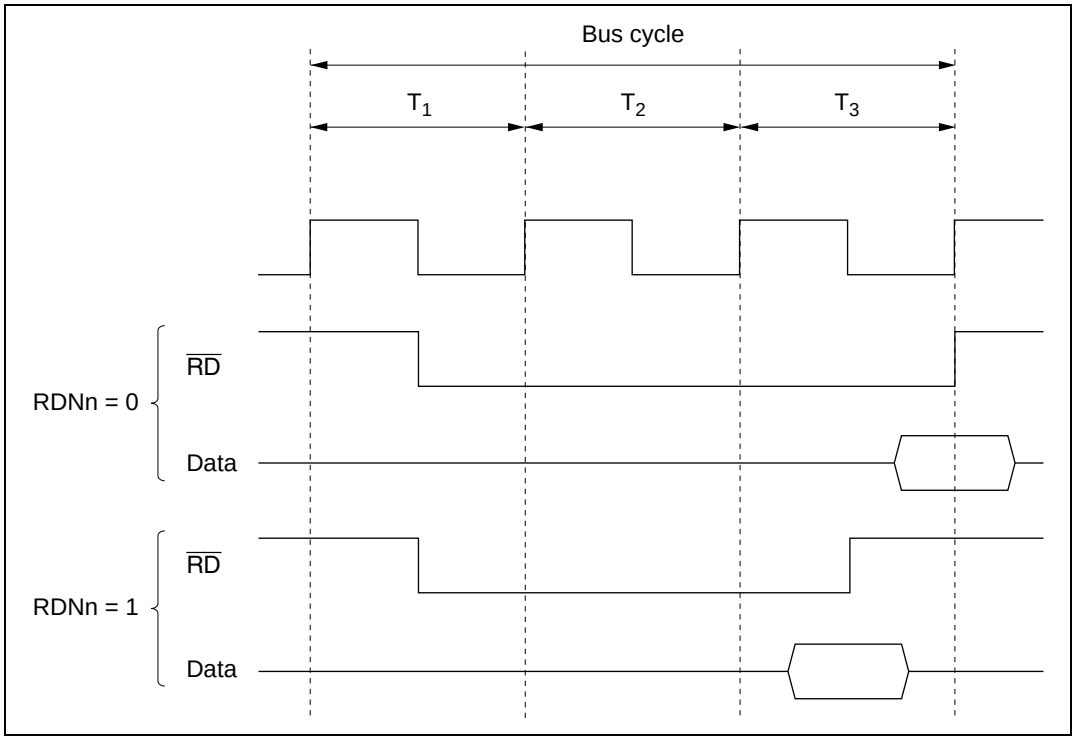


Figure 6.2 Read Strobe Negation Timing (Example of 3-State Access Space)

6.3.5 $\overline{CS}$ Assertion Period Control Registers H, L (CSACRH, CSACRL)

CSACRH and CSACRL select whether or not the assertion period of the basic bus interface chip select signals ($\overline{CSn}$) and address signals is to be extended. Extending the assertion period of the $\overline{CSn}$ and address signals allows flexible interfacing to external I/O devices.

- CSACRH

Bit	Bit Name	Initial Value	R/W	Description
7	CSXH7	0	R/W	$\overline{CS}$ and Address Signal Assertion Period Control 1 These bits specify whether or not the T_h cycle is to be inserted (see figure 6.3). When an area for which the CSXHn bit is set to 1 is accessed, a one-state T_h cycle, in which only the $\overline{CSn}$ and address signals are asserted, is inserted before the normal access cycle. 0: In area n basic bus interface access, the $\overline{CSn}$ and address assertion period (T_h) is not extended 1: In area n basic bus interface access, the $\overline{CSn}$ and address assertion period (T_h) is extended (n = 7 to 0)
6	CSXH6	0	R/W	
5	CSXH5	0	R/W	
4	CSXH4	0	R/W	
3	CSXH3	0	R/W	
2	CSXH2	0	R/W	
1	CSXH1	0	R/W	
0	CSXH0	0	R/W	

- CSACRL

Bit	Bit Name	Initial Value	R/W	Description
7	CSXT7	0	R/W	$\overline{CS}$ and Address Signal Assertion Period Control 2 These bits specify whether or not the T_t cycle shown in figure 6.3 is to be inserted. When an area for which the CSXTn bit is set to 1 is accessed, a one-state T_t cycle, in which only the $\overline{CSn}$ and address signals are asserted, is inserted before the normal access cycle. 0: In area n basic bus interface access, the $\overline{CSn}$ and address assertion period (T_t) is not extended 1: In area n basic bus interface access, the $\overline{CSn}$ and address assertion period (T_t) is extended (n = 7 to 0)
6	CSXT6	0	R/W	
5	CSXT5	0	R/W	
4	CSXT4	0	R/W	
3	CSXT3	0	R/W	
2	CSXT2	0	R/W	
1	CSXT1	0	R/W	
0	CSXT0	0	R/W	

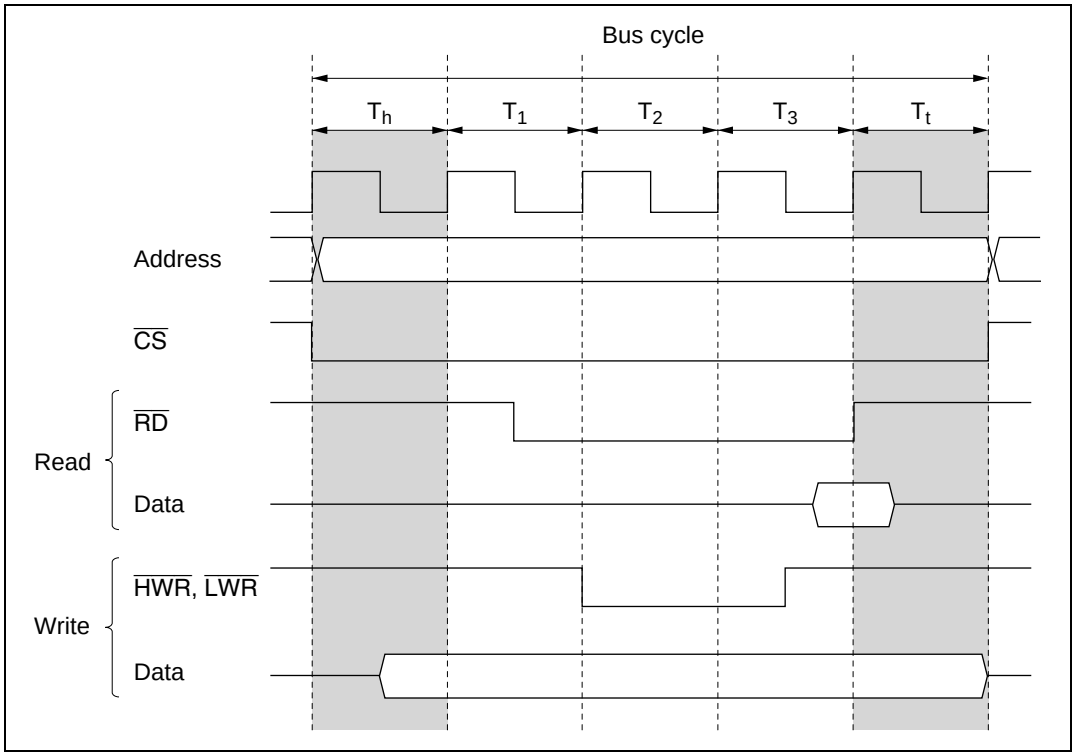


Figure 6.3 $\overline{CS}$ and Address Assertion Period Extension (Example of 3-State Access Space and RDNn = 0)

6.3.6 Bus Control Register (BCR)

BCR is used for idle cycle settings, selection of the external bus released state protocol, enabling or disabling of the write data buffer function, and enabling or disabling of $\overline{\text{WAIT}}$ pin input.

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	External Bus Release Enable Enables or disables external bus release. 0: External bus release disabled $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ pins can be used as I/O ports 1: External bus release enabled
14	BREQOE	0	R/W	$\overline{\text{BREQO}}$ Pin Enable Controls outputting the bus request signal (BREQO) to the external bus master in the external bus released state, when an internal bus master performs an external address space access. 0: $\overline{\text{BREQO}}$ output disabled $\overline{\text{BREQO}}$ pin can be used as I/O port 1: $\overline{\text{BREQO}}$ output enabled
13	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
12	IDLC	1	R/W	Idle Cycle State Number Select Specifies the number of states in the idle cycle set by ICIS2, ICIS1, and ICIS0. 0: Idle cycle comprises 1 state 1: Idle cycle comprises 2 states
11	ICIS1	1	R/W	Idle Cycle Insert 1 When consecutive external read cycles are performed in different areas, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted

Bit	Bit Name	Initial Value	R/W	Description
10	ICIS0	1	R/W	Idle Cycle Insert 0 When an external read cycle and external write cycle are performed consecutively, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted
9	WDBE	0	R/W	Write Data Buffer Enable The write data buffer function can be used for an external write cycle. 0: Write data buffer function not used 1: Write data buffer function used
8	WAITE	0	R/W	$\overline{\text{WAIT}}$ Pin Enable Selects enabling or disabling of wait input by the $\overline{\text{WAIT}}$ pin. 0: Wait input by $\overline{\text{WAIT}}$ pin disabled $\overline{\text{WAIT}}$ pin can be used as I/O port 1: Wait input by $\overline{\text{WAIT}}$ pin enabled
7 to 3	—	All 0	R/W	Reserved These are readable/writable bits, but the write value should always be 0.
2	ICIS2	0	R/W	Idle Cycle Insert 2 When an external write cycle and external read cycle are performed consecutively, an idle cycle can be inserted between the bus cycles. 0: Idle cycle not inserted 1: Idle cycle inserted
1,0	—	All 0	R/W	Reserved These bits can be read from or written to. However, the write value should always be 0.

6.3.7 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit readable/writable up-counter. RTCNT counts up using the internal clock selected by bits RTCK2 to RTCK0 in REFCR.

When RTCNT matches RTCOR (compare match), the CMF flag in REFCR is set to 1 and RTCNT is cleared to H'00. If the RFSHE bit in REFCR is set to 1 at this time, a refresh cycle is started. If the RFSHE bit is cleared to 0 and the CMIE bit in REFCR is set to 1, a compare match interrupt (CMI) is generated.

RTCNT is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

6.3.8 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit readable/writable register that sets the period for compare match operations with RTCNT.

The values of RTCOR and RTCNT are constantly compared, and if they match, the CMF flag in REFCR is set to 1 and RTCNT is cleared to H'00.

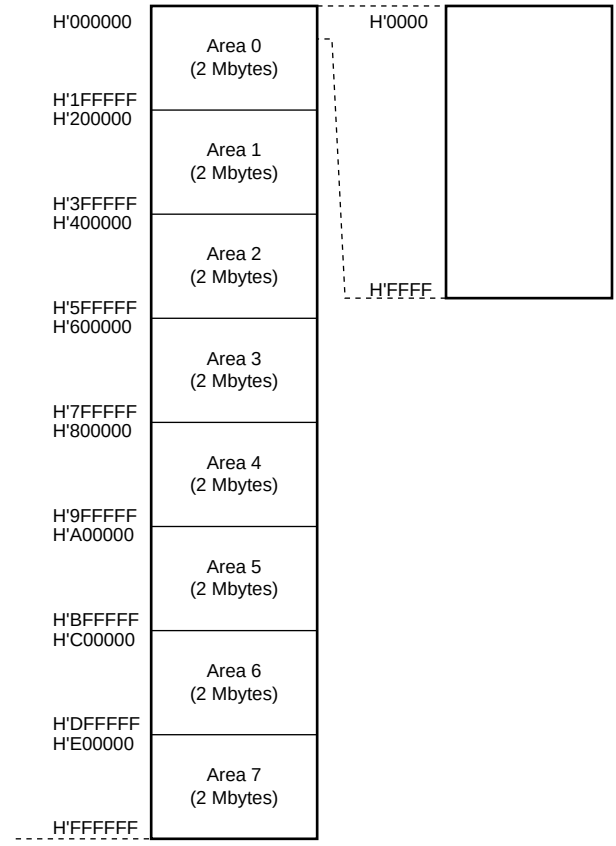
RTCOR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

6.4 Bus Control

6.4.1 Area Division

The bus controller divides the 16-Mbyte address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external address space in area units. Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area. In normal mode, a part of area 0, 64-kbyte address space, is controlled. Figure 6.4 shows an outline of the memory map.

Note: Normal mode is not available in this LSI.



Note: * Not available in this LSI

Figure 6.4 Area Divisions

6.4.2 Bus Specifications

The external address space bus specifications consist of five elements: bus width, number of access states, number of program wait states, read strobe timing, and chip select ($\overline{CS}$) assertion period extension 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 as 8-bit access space, 8-bit bus mode is set; if any area is designated as 16-bit access space, 16-bit bus mode is 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.

When 2-state access space is designated, wait insertion is disabled. When 3-state access space is designated, it is possible to insert program waits by means of the WTCRA and WTCRB, and external waits by means of the $\overline{WAIT}$ pin.

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 WTCRA and WTCRB. From 0 to 7 program wait states can be selected. Table 6.2 shows the bus specifications (bus width, and number of access states and program wait states) for each basic bus interface area.

Table 6.2 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WTCRA, WTCRB			Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn2	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	—	16	2	0
	1	0	0	0		3	0
				1			1
			1	0			2
				1			3
		1	0	0			4
				1			5
			1	0			6
				1			7
1	0	—	—	—	8	2	0
	1	0	0	0		3	0
				1			1
			1	0			2
				1			3
		1	0	0			4
				1			5
			1	0			6
				1			7

(n = 0 to 7)

Read Strobe Timing: RDNCR can be used to select either of two negation timings (at the end of the read cycle or one half-state before the end of the read cycle) for the read strobe ($\overline{RD}$) used in the basic bus interface space.

Chip Select ($\overline{CS}$) Assertion Period Extension States: Some external I/O devices require a setup time and hold time between address and $\overline{CS}$ signals and strobe signals such as $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$. CSACR can be used to insert states in which only the $\overline{CS}$, $\overline{AS}$, and address signals are asserted before and after a basic bus space access cycle.

6.4.3 Memory Interfaces

The memory interfaces in this LSI allows direct connection of ROM, SRAM, and so on.

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.

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

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

Area 1: In externally expanded mode, all of area 1 is external address space.

When area 1 external address space is accessed, the $\overline{CS1}$ signal can be output.

Areas 2 to 5: In externally expanded mode, areas 2 to 5 are all external address space.

When area 2 to 5 external space is accessed, signals $\overline{CS2}$ to $\overline{CS5}$ can be output.

Area 6: In externally expanded mode, all of area 6 is external space.

When area 6 external space is accessed, the $\overline{CS6}$ signal can be output.

Area 7: Area 7 includes the on-chip RAM and internal/O registers. In externally expanded mode, the space excluding the on-chip RAM and internal I/O registers is external address space. The on-chip RAM is enabled when the RAME bit is set to 1 in the system control register (SYSCR); when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding addresses are in external address space.

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

6.4.4 Chip Select Signals

This LSI can output chip select signals ($\overline{CS0}$ to $\overline{CS7}$) for areas 0 to 7. The signal outputs low when the corresponding external space area is accessed. Figure 6.5 shows an example of $\overline{CS0}$ to $\overline{CS7}$ signals output timing.

Enabling or disabling of $\overline{CS0}$ to $\overline{CS7}$ signals output is set by the data direction register (DDR) bit for the port corresponding to the $\overline{CS0}$ to $\overline{CS7}$ pins.

In expanded mode with on-chip ROM disabled, the $\overline{CS0}$ pin is placed in the output state after a reset. Pins $\overline{CS1}$ to $\overline{CS7}$ are placed in the input state after a reset and so the corresponding DDR bits should be set to 1 when outputting signals $\overline{CS1}$ to $\overline{CS7}$.

In expanded mode with on-chip ROM enabled, pins $\overline{CS0}$ to $\overline{CS7}$ are all placed in the input state after a reset and so the corresponding DDR bits should be set to 1 when outputting signals $\overline{CS0}$ to $\overline{CS7}$.

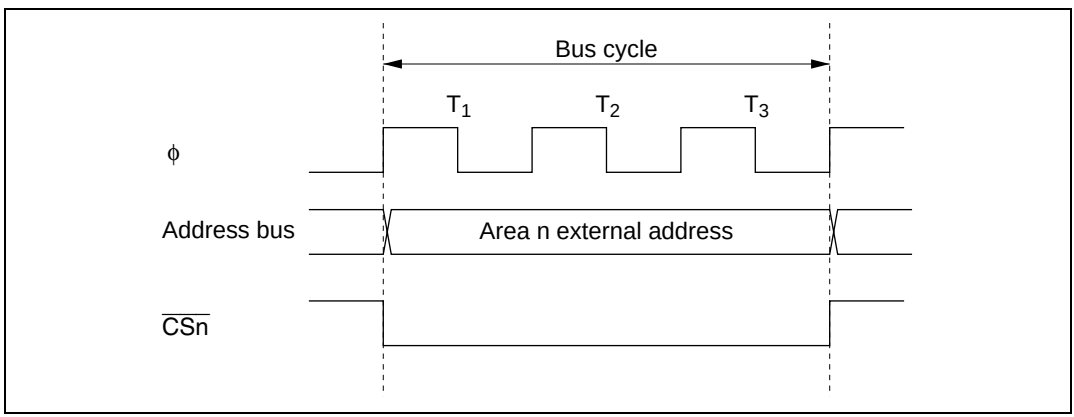


Figure 6.5 $\overline{CSn}$ Signal Output Timing (n = 0 to 7)

6.5 Basic Bus Interface

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

6.5.1 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, and when accessing external address space, 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 6.6 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 access is performed as two byte accesses, and a longword access, as four byte accesses.

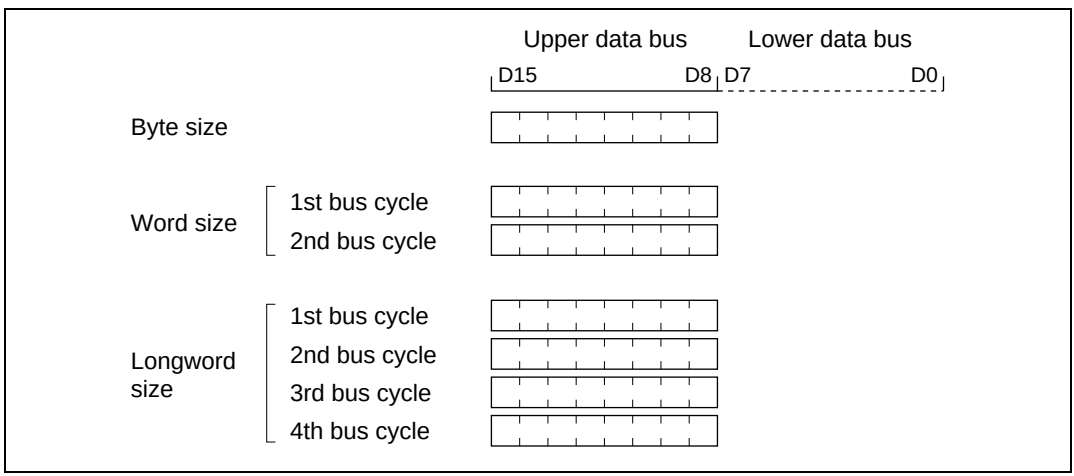


Figure 6.6 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 6.7 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 access is executed as two word accesses.

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

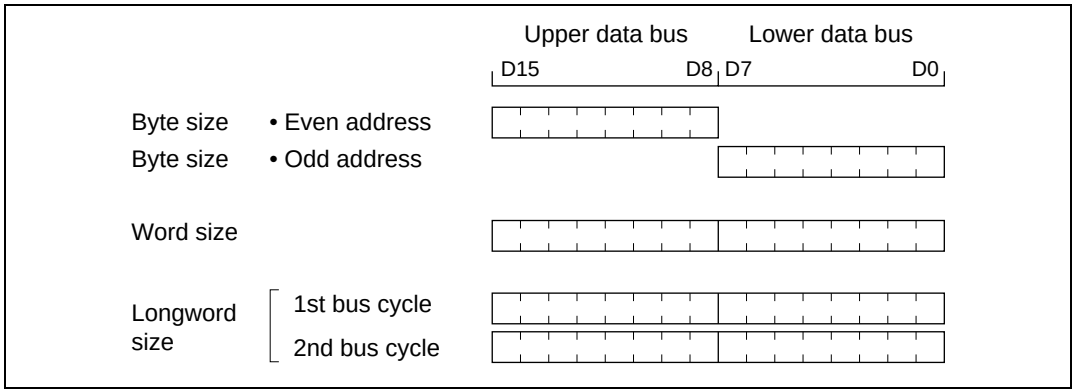


Figure 6.7 Access Sizes and Data Alignment Control (16-bit Access Space)

6.5.2 Valid Strobes

Table 6.3 shows the data buses used and valid strobes for the access spaces.

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

Table 6.3 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	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		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

Note: Hi-Z: High-impedance state
Invalid: Input state; input value is ignored.

6.5.3 Basic Timing

8-Bit, 2-State Access Space: Figure 6.8 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. The $\overline{\text{LWR}}$ pin is always fixed high. Wait states can be inserted.

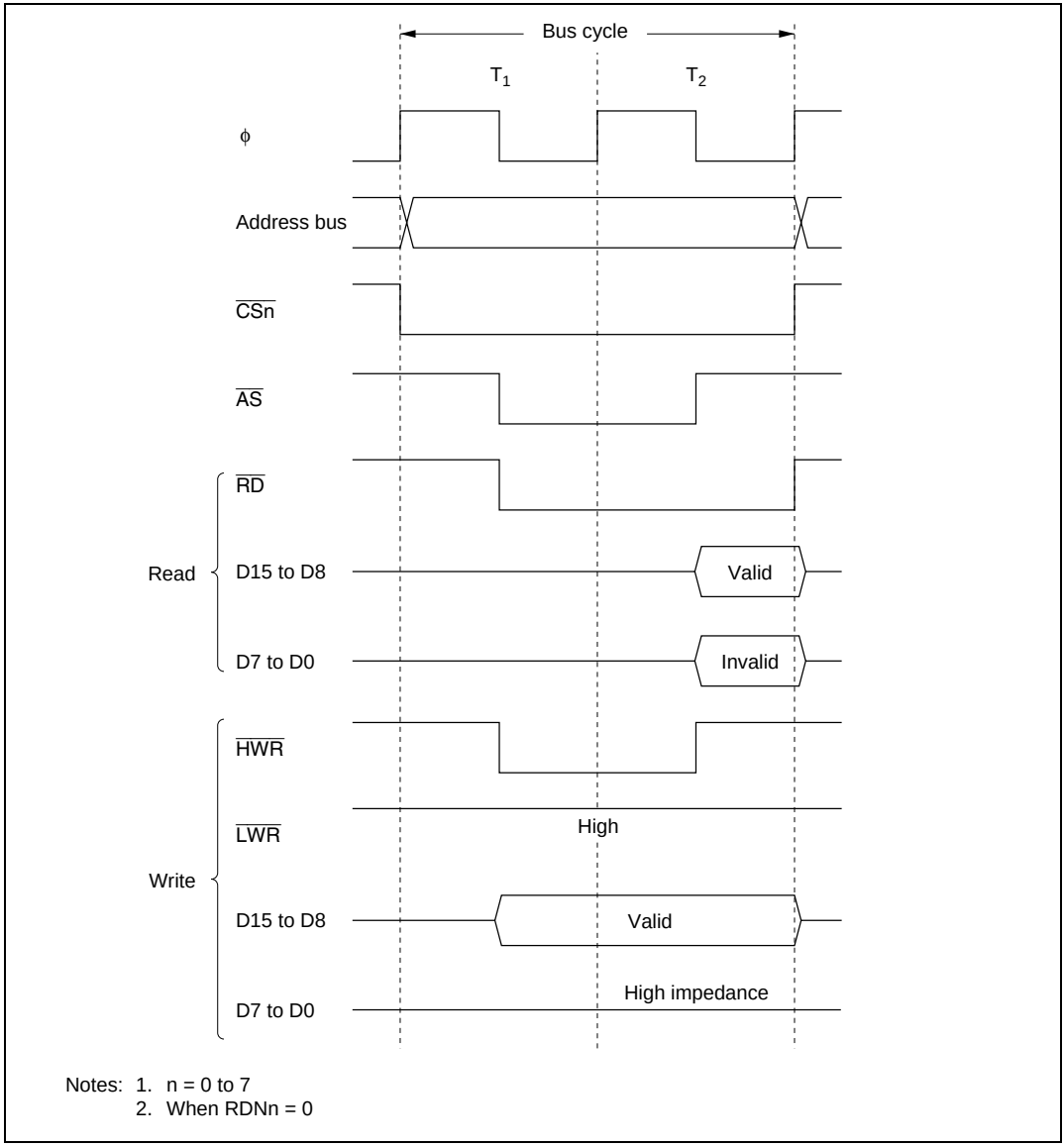


Figure 6.8 Bus Timing for 8-Bit, 2-State Access Space

8-Bit, 3-State Access Space: Figure 6.9 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. The $\overline{\text{LWR}}$ pin is always fixed high. Wait states can be inserted.

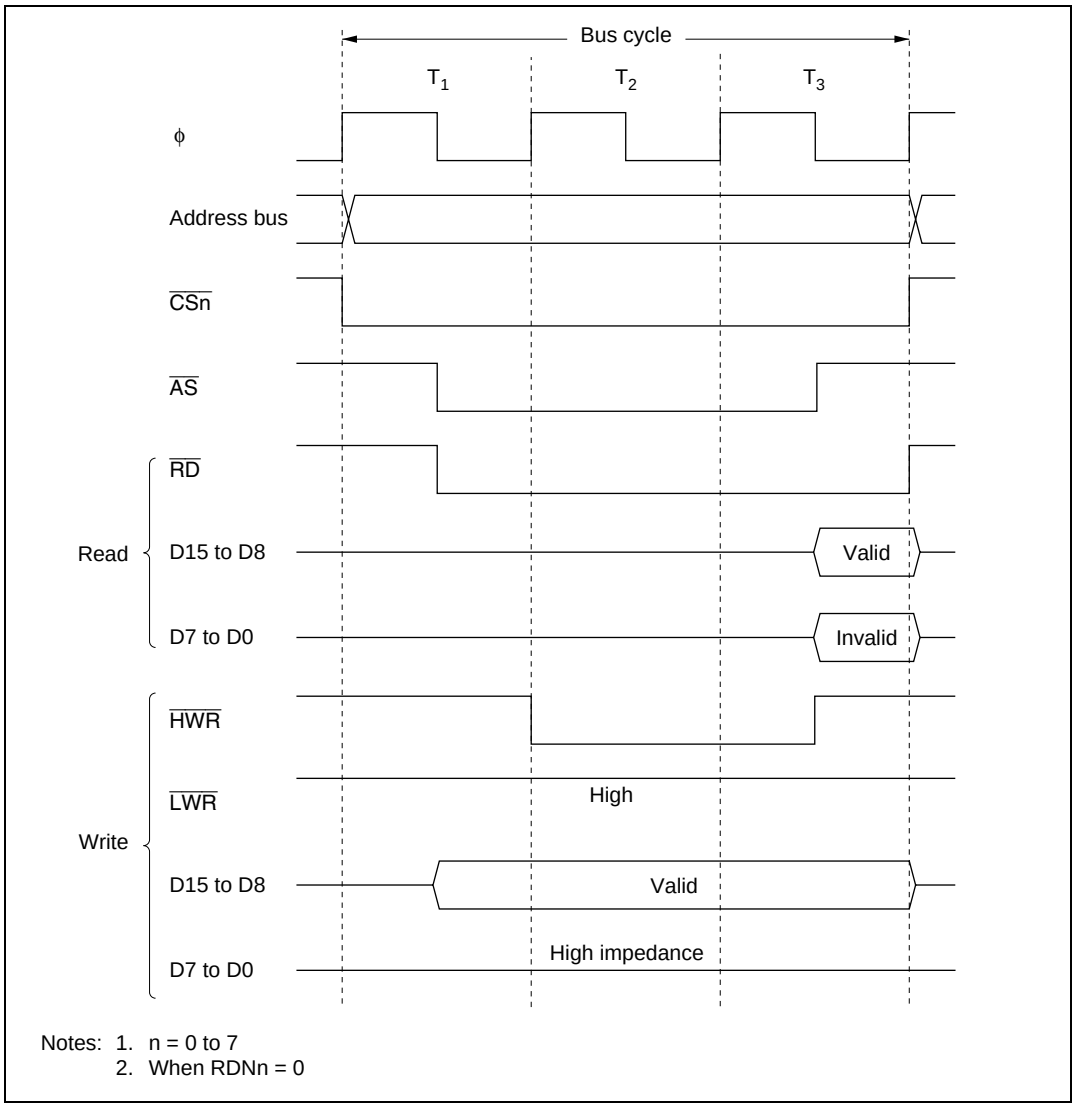
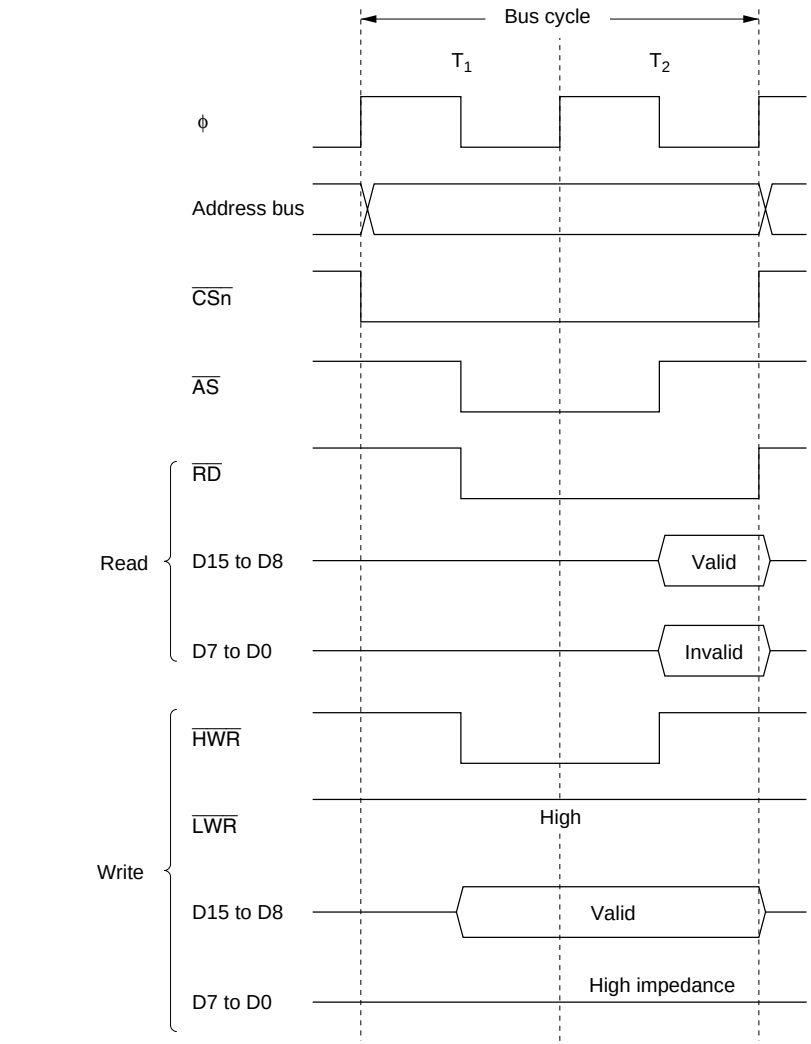


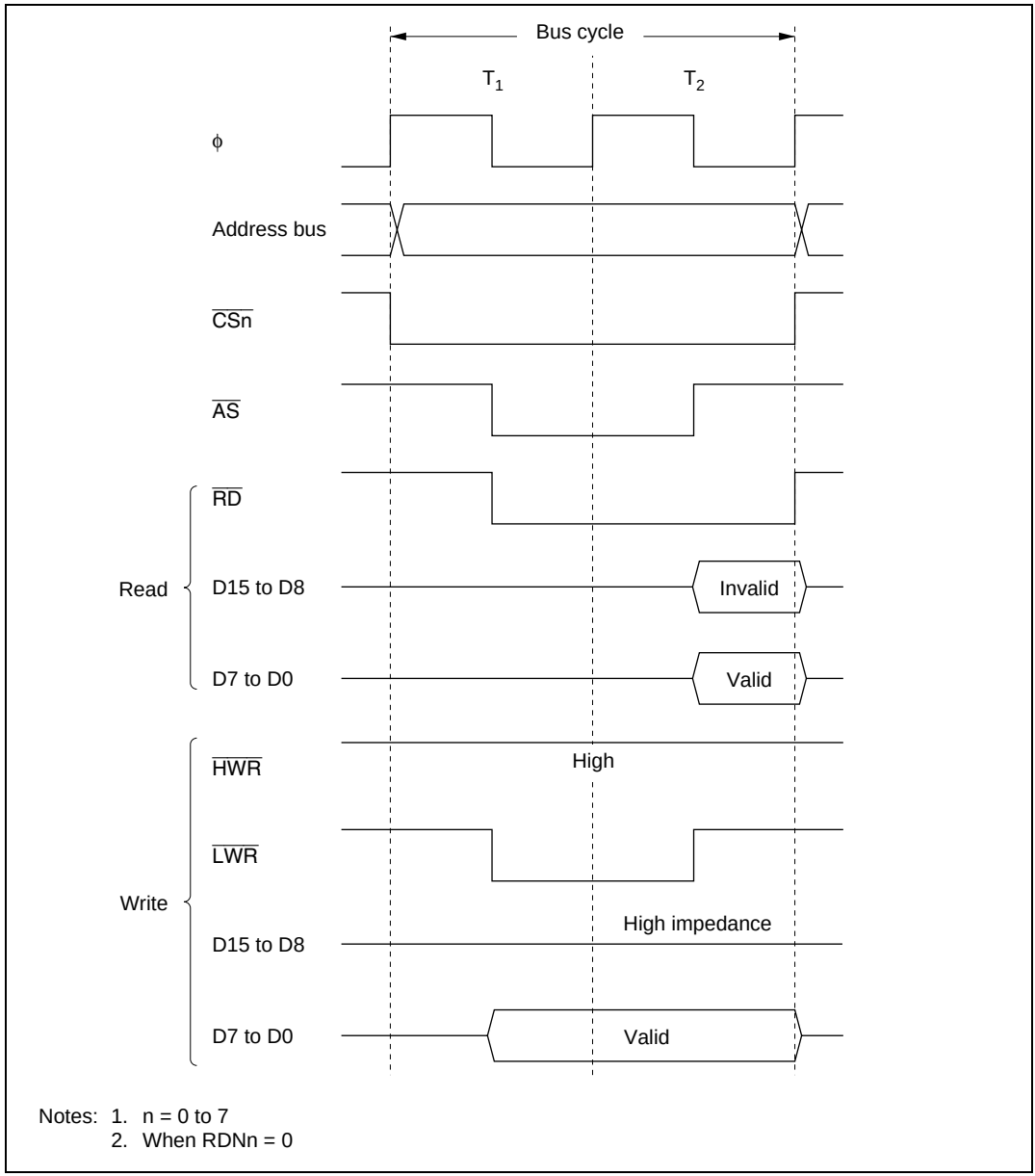
Figure 6.9 Bus Timing for 8-Bit, 3-State Access Space

16-Bit, 2-State Access Space: Figures 6.10 to 6.12 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 odd addresses, and the lower half (D7 to D0) for even addresses. Wait states cannot be inserted.



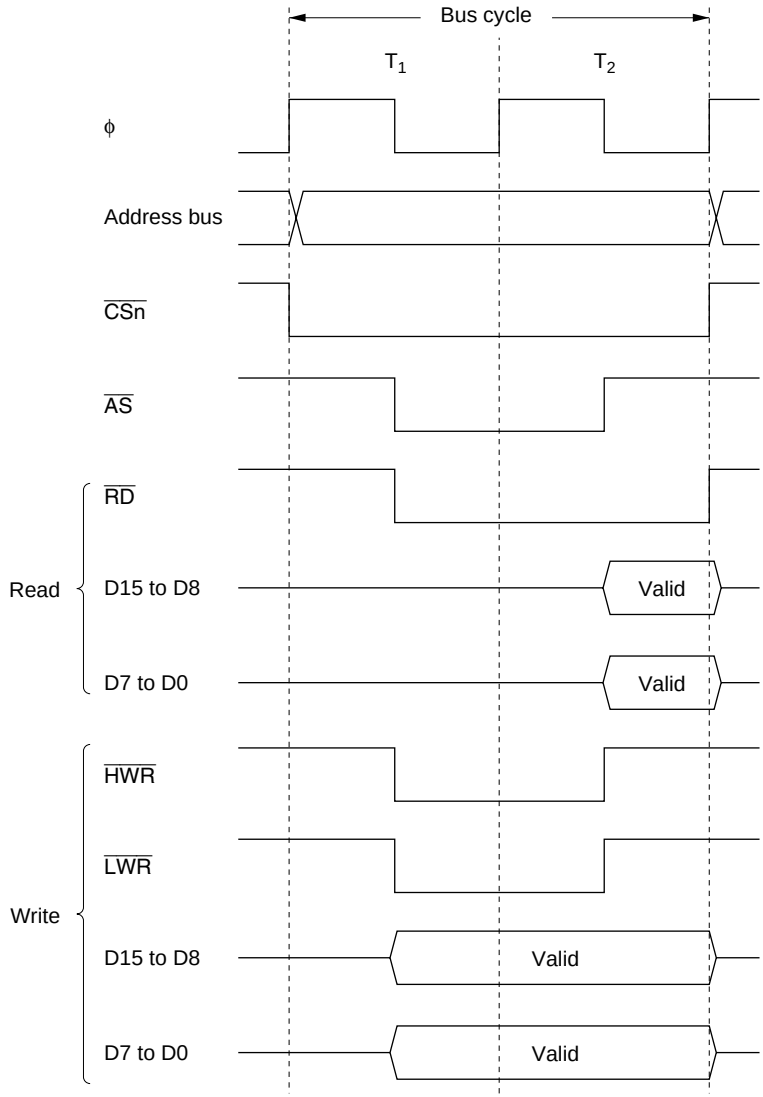
Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.10 Bus Timing for 16-Bit, 2-State Access Space (Even Address Byte Access)



Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.11 Bus Timing for 16-Bit, 2-State Access Space (Odd Address Byte Access)



Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.12 Bus Timing for 16-Bit, 2-State Access Space (Word Access)

16-Bit, 3-State Access Space: Figures 6.13 to 6.15 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 the even address, and the lower half (D7 to D0) for the odd address. Wait states can be inserted.

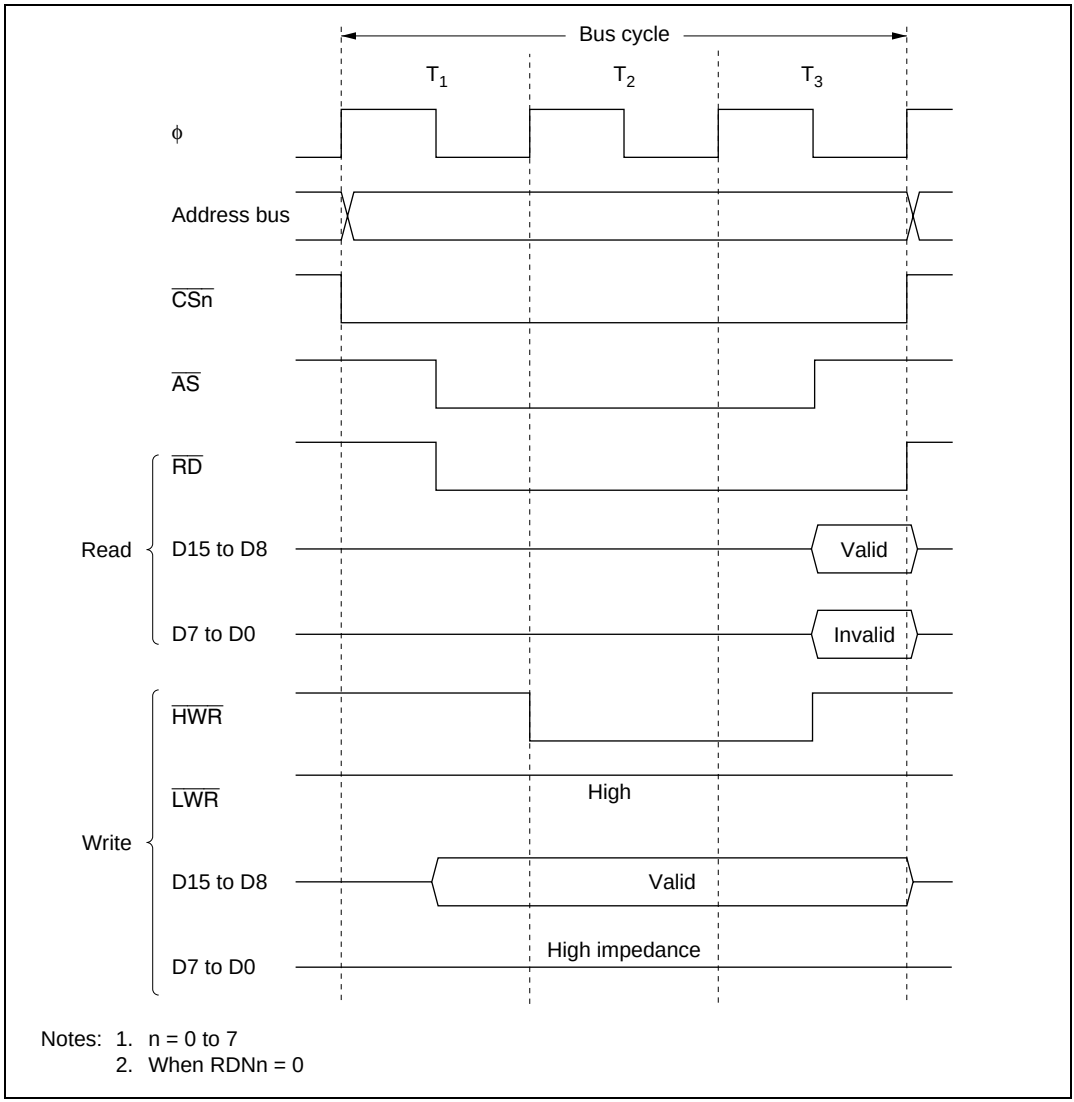


Figure 6.13 Bus Timing for 16-Bit, 3-State Access Space (Even Address Byte Access)

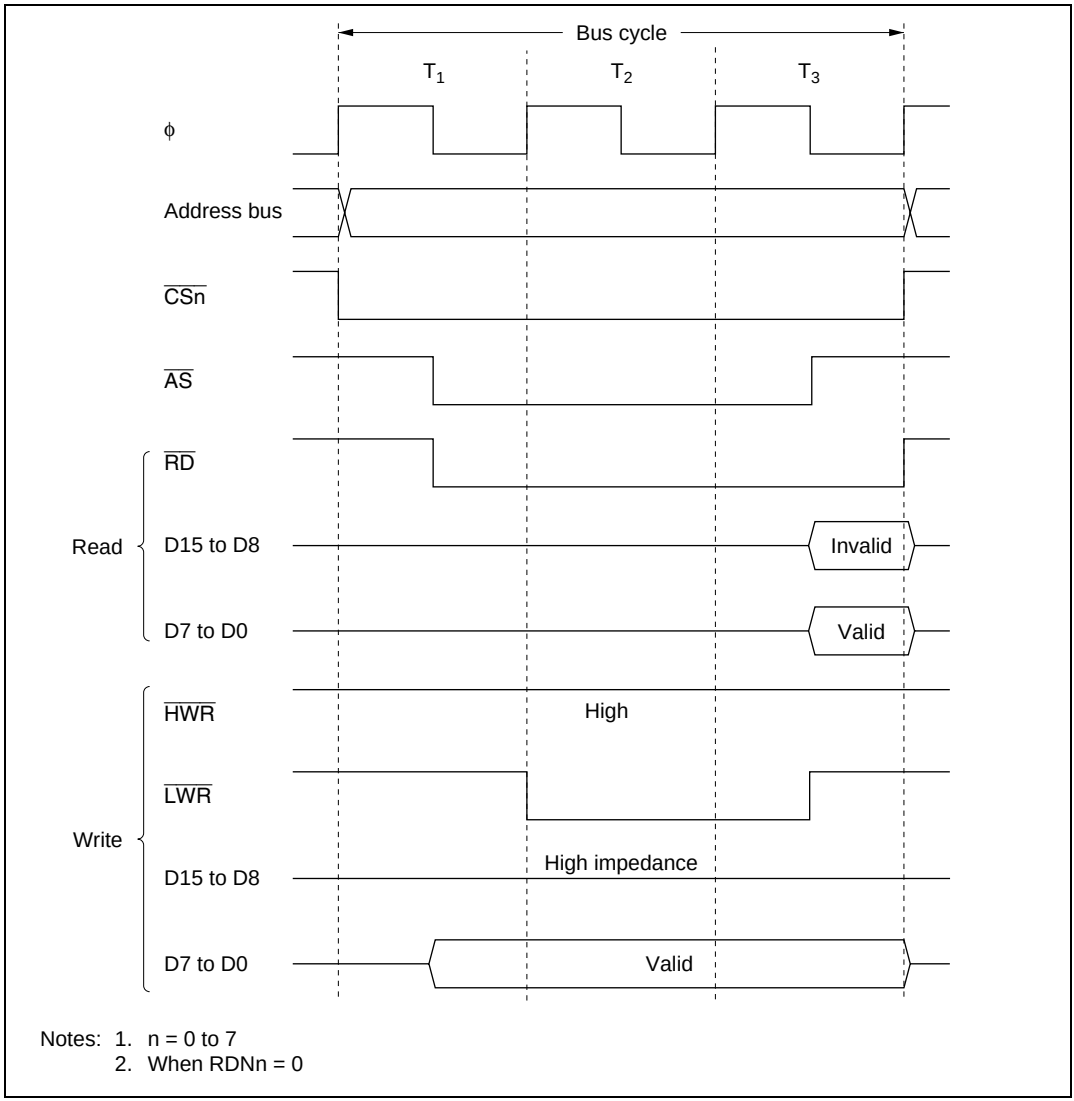
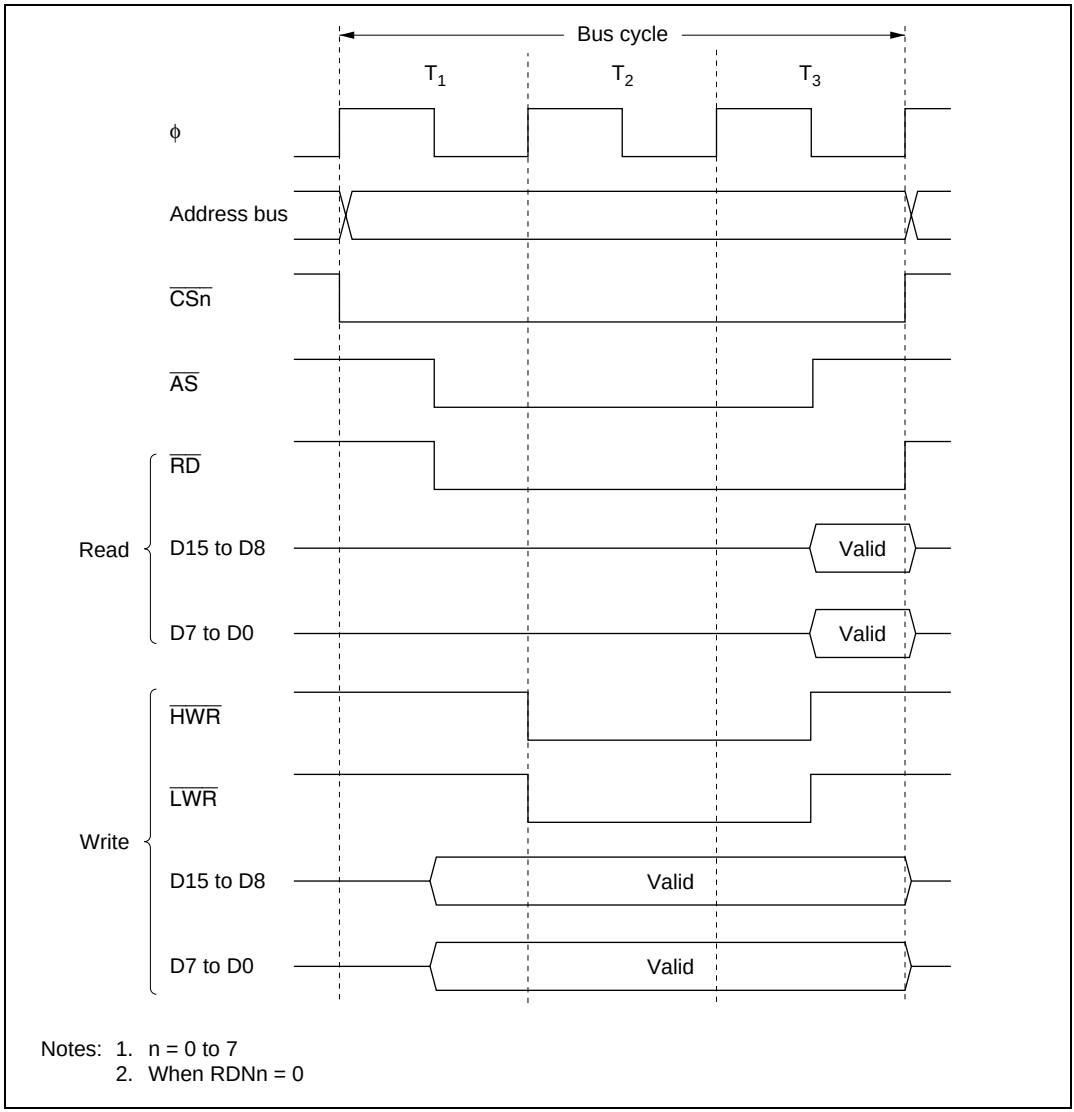


Figure 6.14 Bus Timing for 16-Bit, 3-State Access Space (Odd Address Byte Access)



Notes: 1. n = 0 to 7
2. When RDNn = 0

Figure 6.15 Bus Timing for 16-Bit, 3-State Access Space (Word Access)

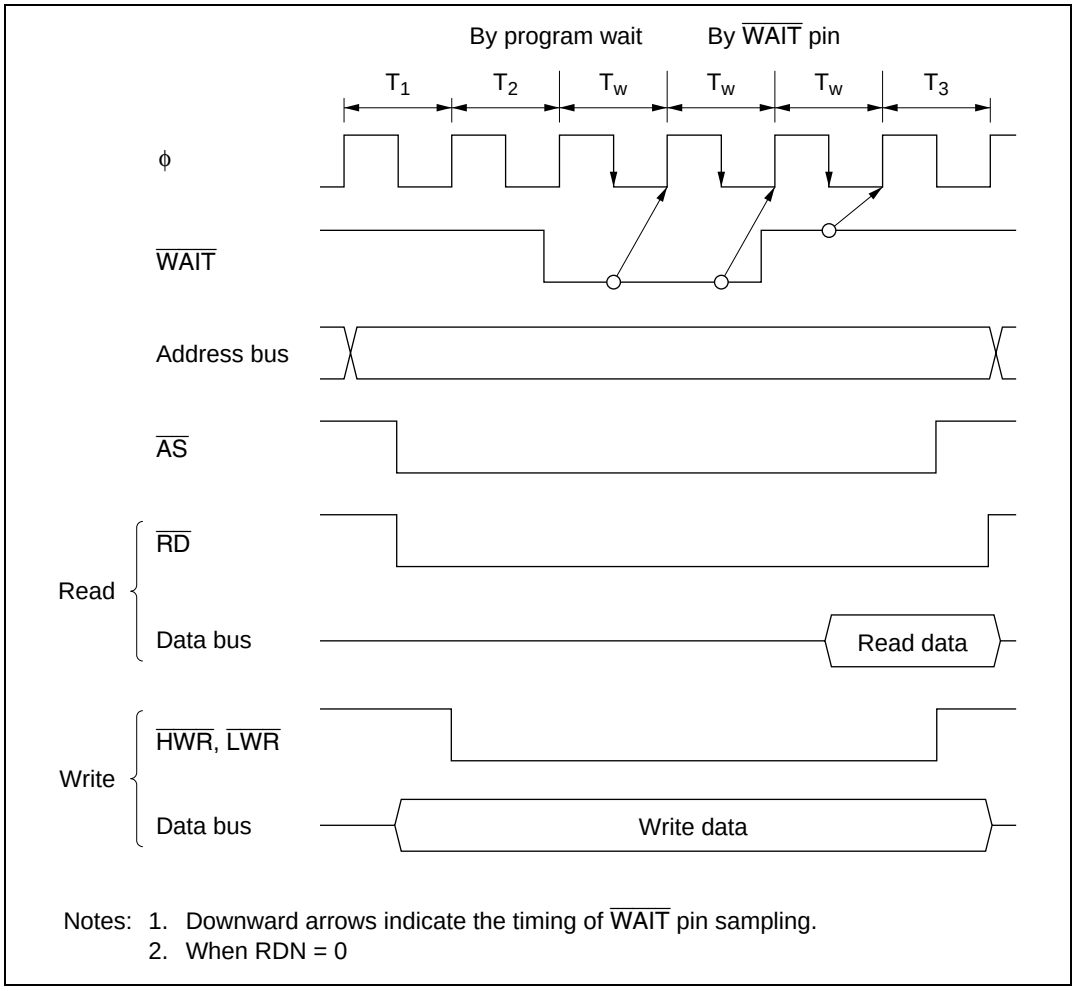
6.5.4 Wait Control

When accessing external space, this LSI can extend the bus cycle by inserting one or more wait states (T_w). There are two ways of inserting wait states: program wait insertion and pin wait insertion using the $\overline{\text{WAIT}}$ pin.

Program Wait Insertion: From 0 to 7 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 in WTCRA and WTCRB.

Pin Wait Insertion: Setting the WAITE bit to 1 in BCR enables wait input by means of the $\overline{\text{WAIT}}$ pin. When an external address space is accessed in this state, a program wait is first inserted in accordance with the settings in WTCRA and WTCRB. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high. This is useful when inserting seven or more T_w states, or when changing the number of T_w states to be inserted for different external devices. The WAITE bit setting applies to all areas. Figure 6.16 shows an example of wait state insertion timing.

The settings after a reset are: 3-state access, insertion of 7 program wait states, and $\overline{\text{WAIT}}$ input disabled.



Notes: 1. Downward arrows indicate the timing of $\overline{WAIT}$ pin sampling.
2. When RDN = 0

Figure 6.16 Example of Wait State Insertion Timing

6.5.5 Read Strobe ($\overline{RD}$) Timing

The read strobe ($\overline{RD}$) timing can be changed for individual areas by setting bits RDN7 to RDN0 to 1 in RDNCR. Figure 6.17 shows an example of the timing when the read strobe timing is changed in basic bus 3-state access space.

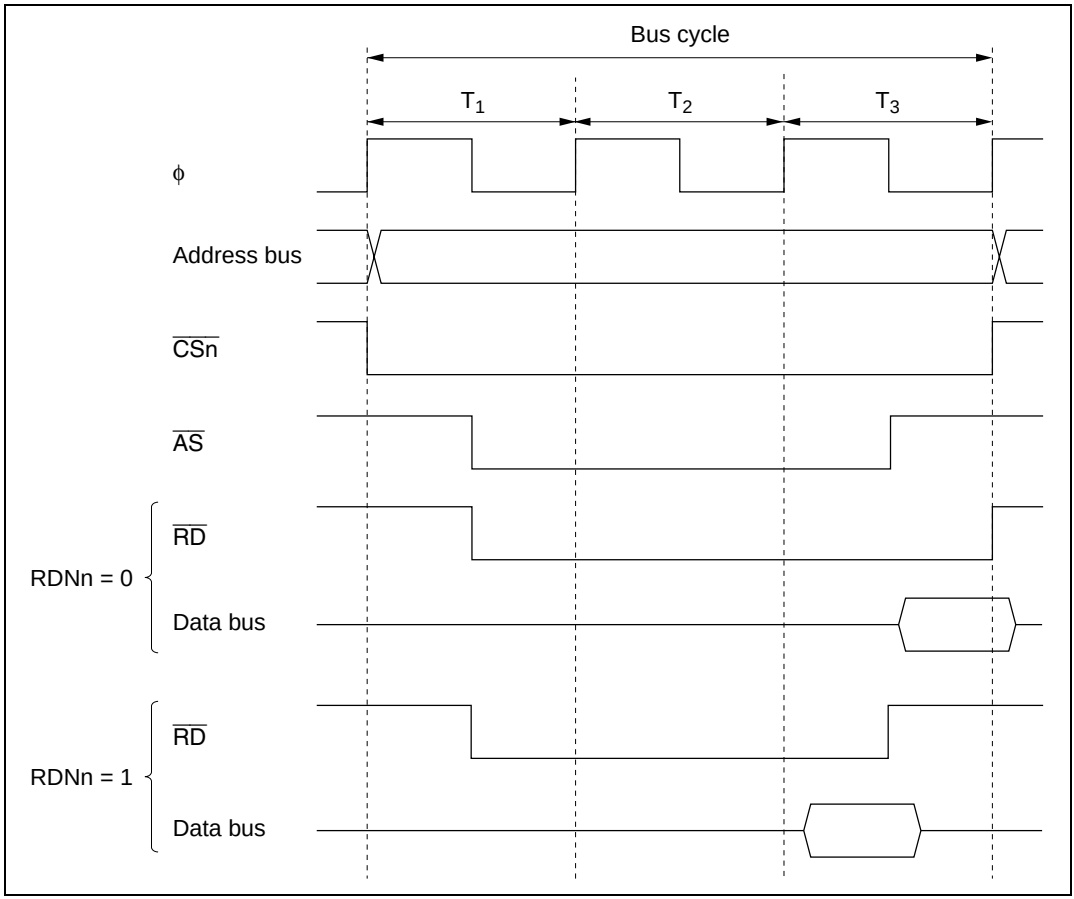


Figure 6.17 Example of Read Strobe Timing

6.5.6 Extension of Chip Select ($\overline{CS}$) Assertion Period

Some external I/O devices require a setup time and hold time between address and $\overline{CS}$ signals and strobe signals such as $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$. Settings can be made in the CSACR register to insert states in which only the $\overline{CS}$, $\overline{AS}$, and address signals are asserted before and after a basic bus space access cycle. Extension of the $\overline{CS}$ assertion period can be set for individual areas. With the $\overline{CS}$

assertion extension period in write access, the data setup and hold times are less stringent since the write data is output to the data bus.

Figure 6.18 shows an example of the timing when the $\overline{CS}$ assertion period is extended in basic bus 3-state access space.

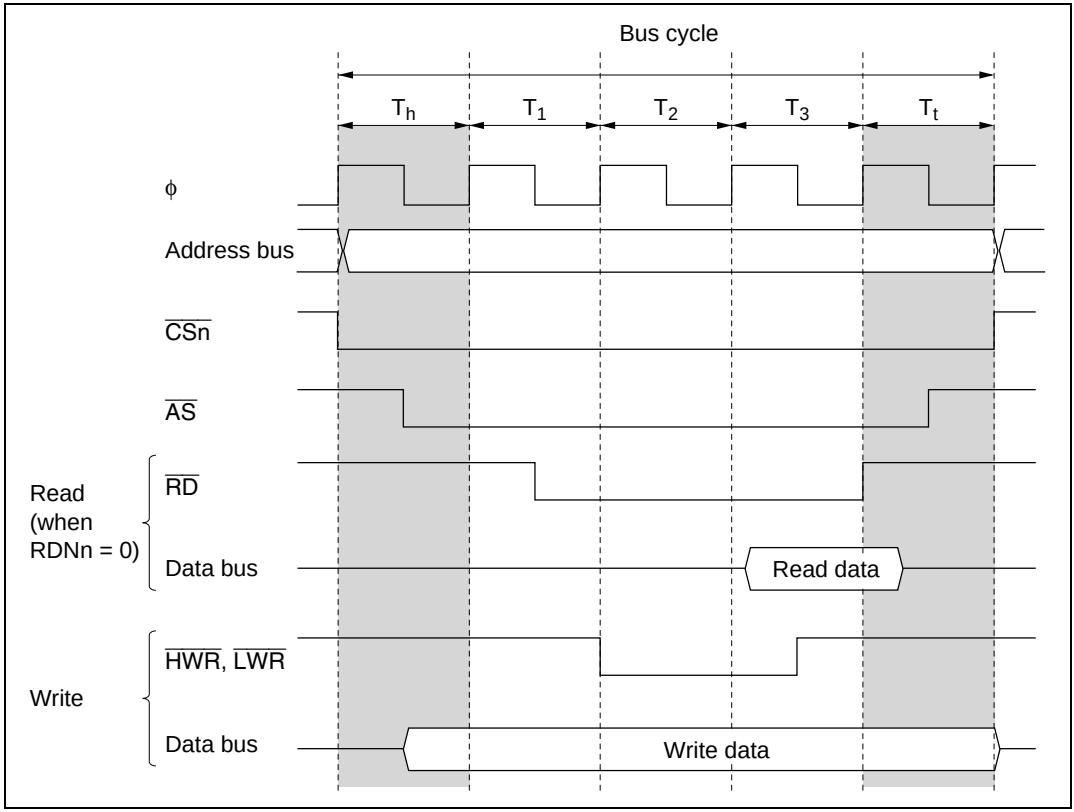


Figure 6.18 Example of Timing when Chip Select Assertion Period Is Extended

Both extension state T_h inserted before the basic bus cycle and extension state T_t inserted after the basic bus cycle, or only one of these, can be specified for individual areas. Insertion or non-insertion can be specified for the T_h state with the upper 8 bits (CSXH7 to CSXH0) in the CSACR register, and for the T_t state with the lower 8 bits (CSXT7 to CSXT0).

6.6 Idle Cycle

6.6.1 Operation

When this LSI accesses external address space, it can insert an idle cycle (T_i) between bus cycles in the following three cases: (1) when read accesses in different areas occur consecutively, (2) when a write cycle occurs immediately after a read cycle, and (3) when a read cycle occurs immediately after a write cycle. Insertion of a 1-state or 2-state idle cycle can be selected with the IDLC bit in BCR. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, etc., with a long output floating time, and high-speed memory, I/O interfaces, and so on.

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

Figure 6.19 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a read cycle for SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

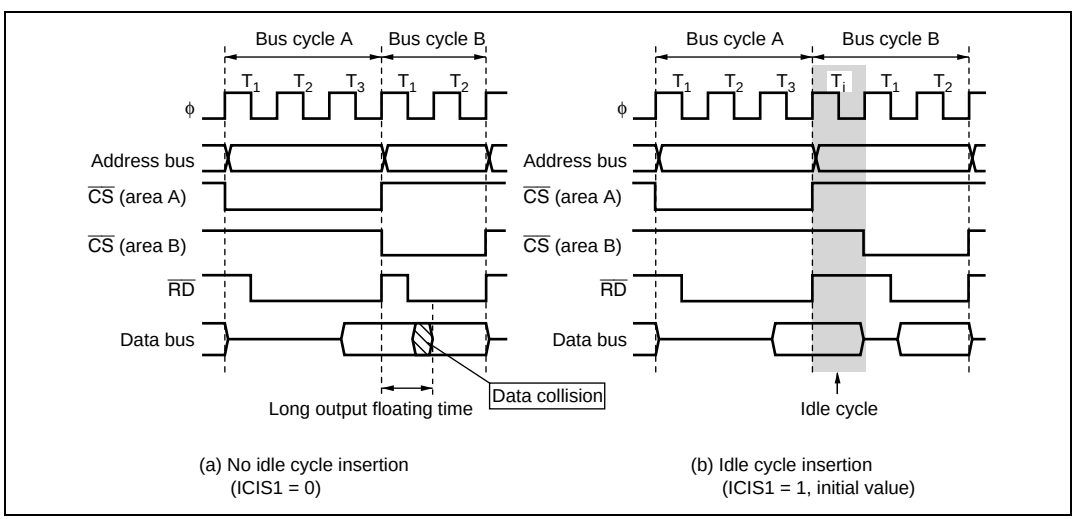


Figure 6.19 Example of Idle Cycle Operation (Consecutive Reads in Different Areas)

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

Figure 6.20 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

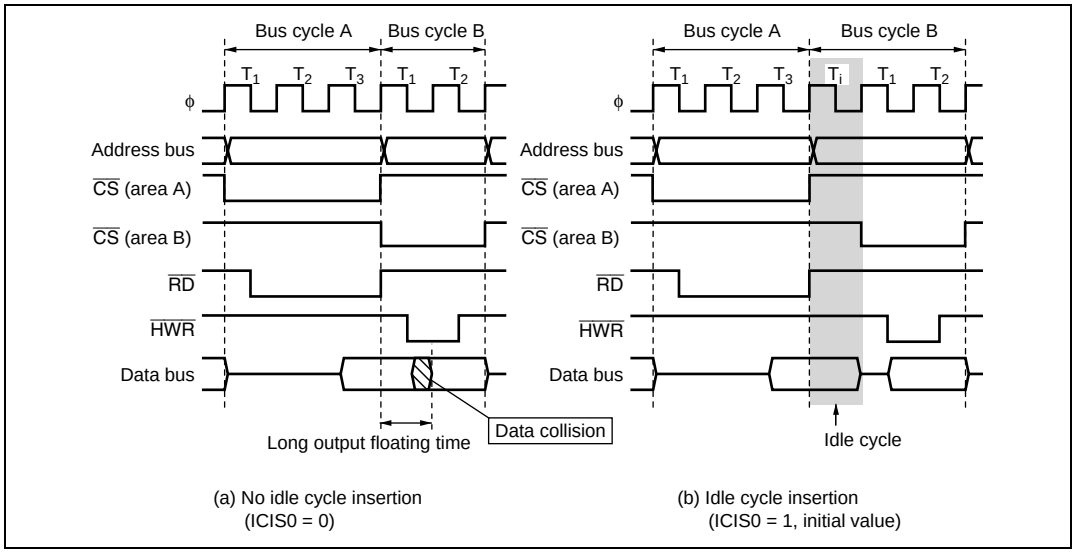


Figure 6.20 Example of Idle Cycle Operation (Write after Read)

Read after Write: If an external read occurs after an external write while the ICIS2 bit is set to 1 in BCR, an idle cycle is inserted at the start of the read cycle.

Figure 6.21 shows an example of the operation in this case. In this example, bus cycle A is a CPU write cycle and bus cycle B is a read cycle from an external device. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the CPU write data and read data from an external device. In (b), an idle cycle is inserted, and a data collision is prevented.

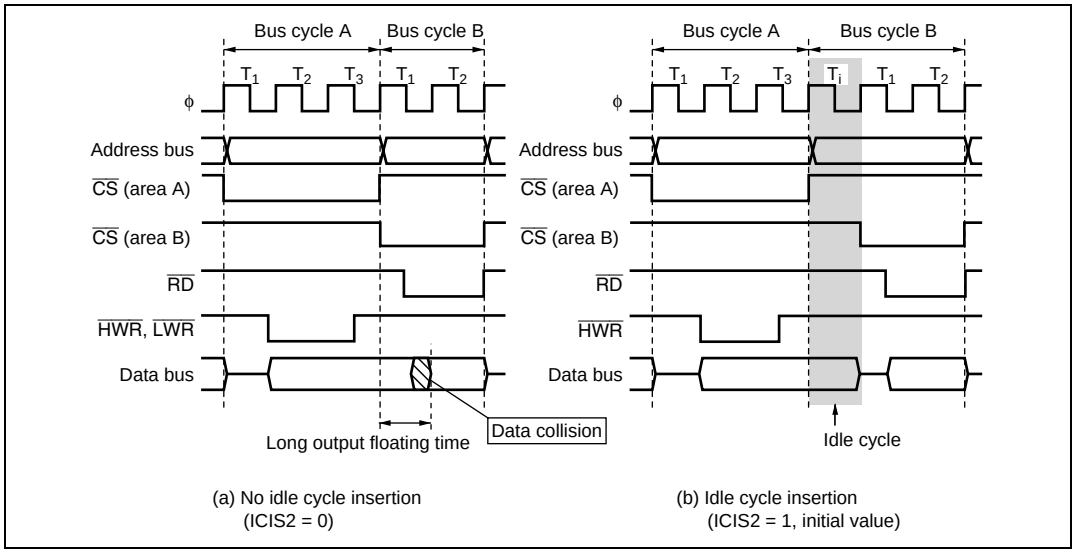


Figure 6.21 Example of Idle Cycle Operation (Read after Write)

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 6.22. 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.

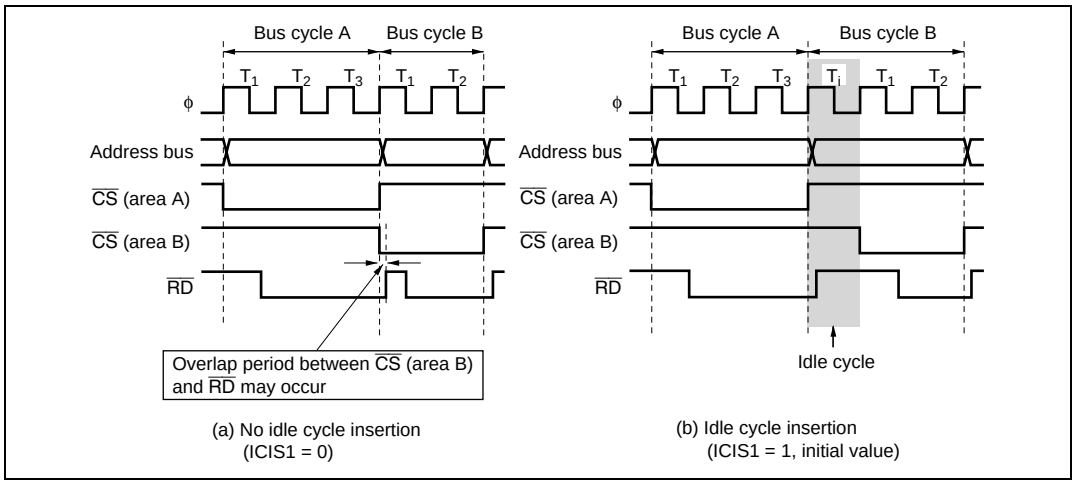


Figure 6.22 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

6.6.2 Pin States in Idle Cycle

Table 6.4 shows the pin states in an idle cycle.

Table 6.4 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of following bus cycle
D15 to D0	High impedance
$\overline{CSn}$ (n = 7 to 0)	High
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$, $\overline{LWR}$	High

6.7 Write Data Buffer Function

This LSI has a write data buffer function for the external data bus. Using the write data buffer function enables external writes to be executed in parallel with internal accesses. The write data buffer function is made available by setting the WDBE bit to 1 in BCR.

Figure 6.23 shows an example of the timing when the write data buffer function is used. When this function is used, if an external address space write continues for two states or longer, and there is an internal access next, an external write only is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external address space write rather than waiting until it ends.

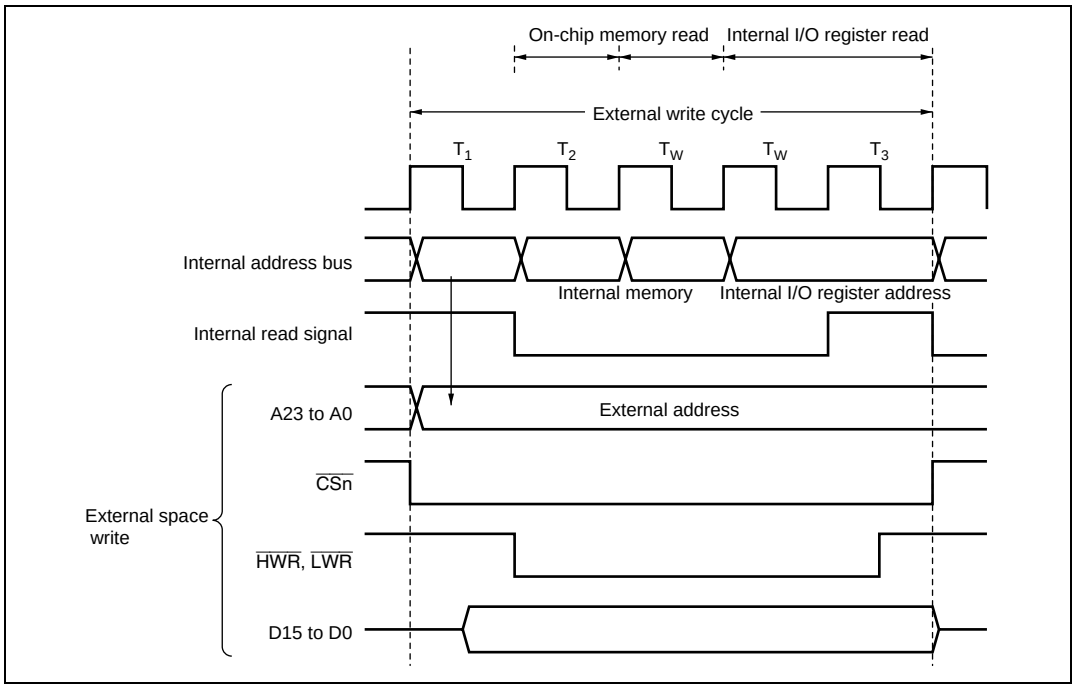


Figure 6.23 Example of Timing when Write Data Buffer Function Is Used

6.8 Bus Release

This LSI can release the external bus in response to a bus request from an external device. In the external bus released state, internal bus masters continue to operate as long as there is no external access. If any of the following requests are issued in the external bus released state, the $\overline{\text{BREQO}}$ signal can be driven low to output a bus request externally.

- When an internal bus master wants to perform an external access
- When a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode

6.8.1 Operation

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

In the external bus released state, internal bus masters can perform accesses using the internal bus. When an internal bus master wants to make an external access, it temporarily defers initiation of the bus cycle, and waits for the bus request from the external bus master to be canceled. If a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode, software standby and all-module-clocks-stopped control are deferred until the bus request from the external bus master is canceled.

If the BREQOE bit is set to 1 in BCR, the $\overline{\text{BREQO}}$ pin can be driven low when any of the following requests are issued, to request cancellation of the bus request externally.

- When an internal bus master wants to perform an external access
- When a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode

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 released state is terminated.

If an external bus release request and external access occur simultaneously, the order of priority is as follows:

(High) External bus release > External access by internal bus master (Low)

6.8.2 Pin States in External Bus Released State

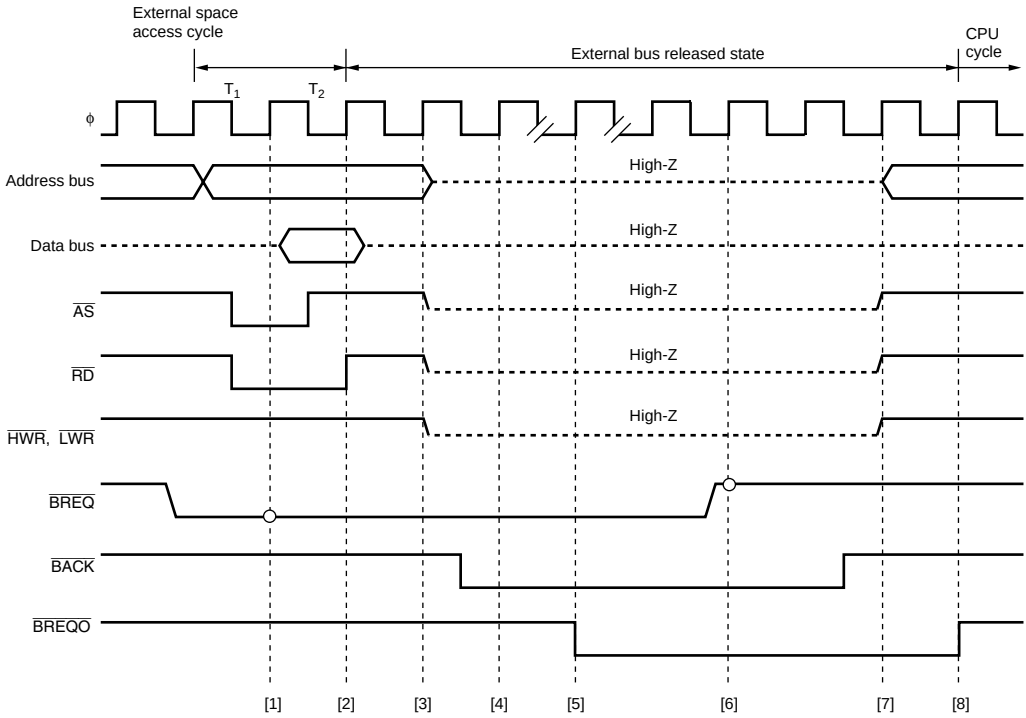
Table 6.5 shows pin states in the external bus released state.

Table 6.5 Pin States in Bus Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{CSn}}$ (n = 7 to 0)	High impedance
$\overline{\text{AS}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{HWR}}$, $\overline{\text{LWR}}$	High impedance

6.8.3 Transition Timing

Figure 6.24 shows the timing for transition to the bus released state.



[1] Low level of $\overline{\text{BREQ}}$ signal is sampled at rise of ø.

[2] Bus control signal returns to be high at end of external space access cycle. At least one state from sampling of $\overline{\text{BREQ}}$ signal.

[3] $\overline{\text{BACK}}$ signal is driven low, releasing bus to external bus master.

[4] $\overline{\text{BREQ}}$ signal state is also sampled in external bus released state.

[5] High level of $\overline{\text{BREQ}}$ signal is sampled.

[6] $\overline{\text{BACK}}$ signal is driven high, ending external bus release cycle.

[7] When there is external access of internal bus master during external bus release while BREQOE bit is set to 1, $\overline{\text{BREQO}}$ signal goes low.

[8] Normally $\overline{\text{BREQO}}$ signal goes high 1.5 states after rising edge of $\overline{\text{BACK}}$ signal.

Figure 6.24 Bus Released State Transition Timing

6.9 Bus Arbitration

This LSI has a bus arbiter that arbitrates bus mastership operations (bus arbitration).

There are two bus masters—the CPU and DTC—that perform read/write operations when they have possession of the bus. Each bus master requests the bus 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 request acknowledge signal. The selected bus master then takes possession of the bus and begins its operation.

6.9.1 Operation

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

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

(High) DTC > CPU (Low)

6.9.2 Bus Transfer Timing

Even if a bus 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 is not necessarily transferred immediately. There are specific timings at which each bus master can relinquish the bus.

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

- The bus 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 is not transferred between the component operations.
- With bit manipulation instructions such as BSET and BCLR, the sequence of operations is: data read (read), relevant bit manipulation operation (modify), write-back (write). The bus is not transferred during this read-modify-write cycle, which is executed as a series of bus cycles.
- If the CPU is in sleep mode, the bus is transferred immediately.

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

The DTC can release the bus 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 during a register information read (3 states), a single data transfer, or a register information write (3 states).

External Bus Release: When the $\overline{\text{BREQ}}$ pin goes low and an external bus release request is issued while the BRLE bit is set to 1 in BCR, a bus request is sent to the bus arbiter.

External bus release can be performed on completion of an external bus cycle.

6.10 Bus Controller Operation in Reset

In a reset, this LSI, including the bus controller, enters the reset state immediately, and any executing bus cycle is aborted.

6.11 Usage Notes

6.11.1 External Bus Release Function and All-Module-Clocks-Stopped Mode

In this LSI, if the ACSE bit is set to 1 in MSTPCR, and then a SLEEP instruction is executed with the setting for all peripheral module clocks to be stopped (MSTPCR = H'FFFF) or for operation of the 8-bit timer module alone (MSTPCR = H'FFFE), and a transition is made to the sleep state, the all-module-clocks-stopped mode is entered in which the clock is also stopped for the bus controller and I/O ports. In this state, the external bus release function is halted. To use the external bus release function in sleep mode, the ACSE bit in MSTPCR must be cleared to 0. Conversely, if a SLEEP instruction to place the chip in all-module-clocks-stopped mode is executed in the external bus released state, the transition to all-module-clocks-stopped mode is deferred and performed until after the bus is recovered.

6.11.2 External Bus Release Function and Software Standby

In this LSI, internal bus mastership operation does not stop even while the bus is released, as long as the program is running in on-chip ROM, etc., and no external access occurs. If a SLEEP instruction to place the chip in software standby mode is executed while the external bus is released, the transition to software standby mode is deferred and performed after the bus is recovered.

Also, since clock oscillation halts in software standby mode, if $\overline{\text{BREQ}}$ goes low in this mode, indicating an external bus release request, the request cannot be answered until the chip has recovered from the software standby state.

6.11.3 $\overline{\text{BREQO}}$ Output Timing

When the BREQOE bit is set to 1 and the $\overline{\text{BREQO}}$ signal is output, $\overline{\text{BREQO}}$ may go low before the $\overline{\text{BACK}}$ signal.

This will occur if the next external access request occurs while internal bus arbitration is in progress after the chip samples a low level of $\overline{\text{BREQ}}$.

Section 7 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 7.1 shows a block diagram of the DTC. The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

7.1 Features

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

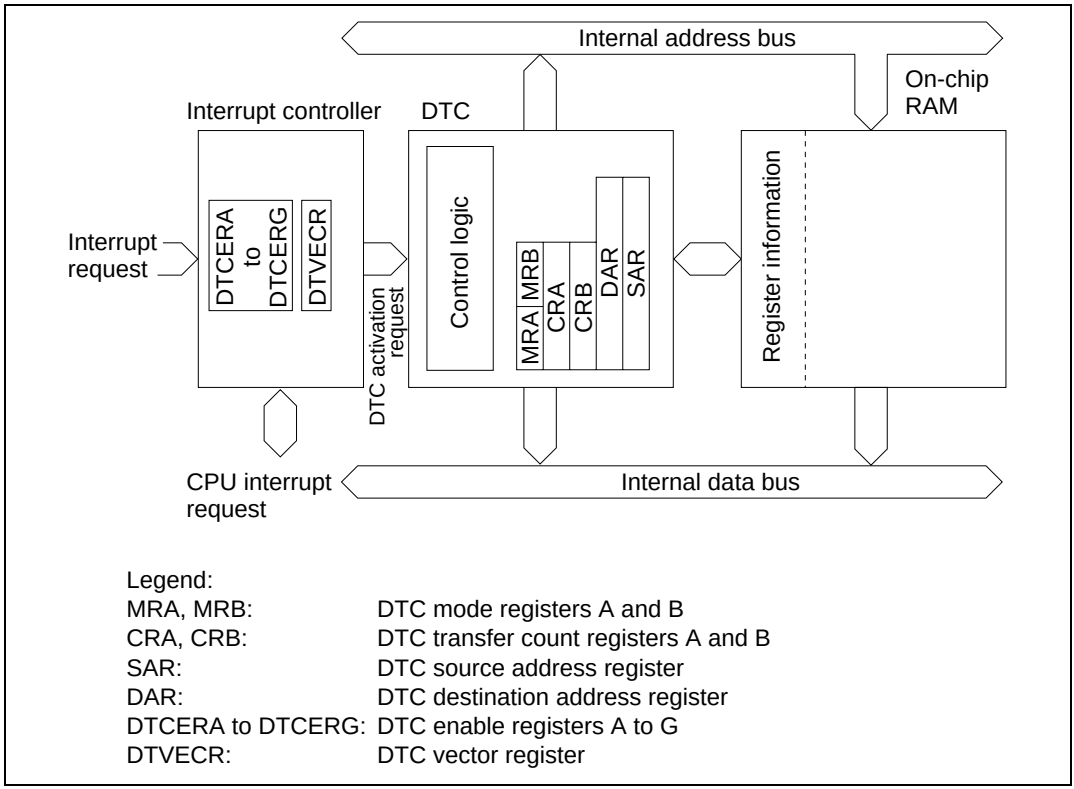


Figure 7.1 Block Diagram of DTC

7.2 Register Descriptions

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 an on-chip RAM to the corresponding DTC registers and transfers data. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable registers A to H (DTCERA to DTCERH)
- DTC vector register (DTVECR)

7.2.1 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
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
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

7.2.2 DTC Mode Register B (MRB)

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. For details, refer to section 7.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the activation source flag, and clearing of DTCER is not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated every time after a data transfer ends. When this bit is set to 0, a CPU interrupt request is generated at the time when the specified number of data transfer ends.
5	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. 0: Chain transfer every time 1: Chain transfer only when transfer counter = 0
4 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation, and should always be written with 0.

7.2.3 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.

7.2.4 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.

7.2.5 DTC Transfer Count Register A (CRA)

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

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65,536). 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.

7.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

7.2.7 DTC Enable Registers A to H (DTCERA to DTCERH)

DTCER which is comprised of seven registers, DTCERA to DTCERH, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 7.1. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCE7	0	R/W	DTC Activation Enable
6	DTCE6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source. [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

7.2.8 DTC Vector Register (DTVECR)

DTVECR 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 Setting this bit to 1 activates DTC. Only 1 can be written to this bit. [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU. When the DISEL bit is 1 and data transfer has ended or when the specified number of transfers have ended, this bit will not be cleared.
6	DTVEC6	0	R/W	DTC Software Activation Vectors 6 to 0 These bits specify a vector number for DTC software activation. 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. When the bit SWDTE is 0, these bits can be written.
5	DTVEC5	0	R/W	
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

7.3 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI_0.

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

Figure 7.2 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

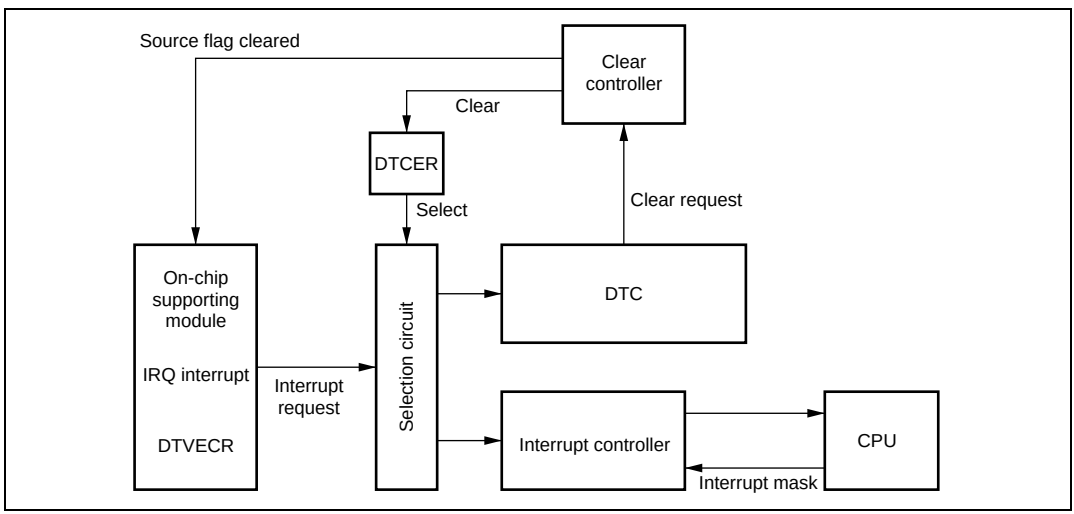


Figure 7.2 Block Diagram of DTC Activation Source Control

7.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFBC00 to H'FFBFFF). Register information should be located at the address that is multiple of four within the range. Locating the register information in address space is shown in figure 7.3. 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 7.3 and the register information start address should be located at the corresponding vector address to the activation source. 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: * Not available in this LSI.

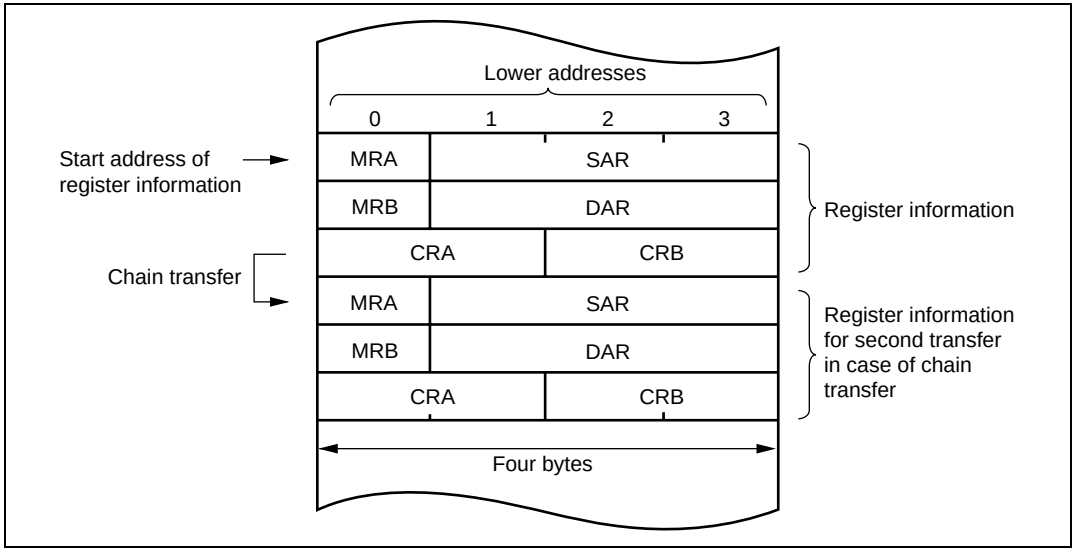


Figure 7.3 Correspondence between DTC Vector Address and Register Information

Table 7.1 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (DTVECR [6:0] × 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	
Reserved		24	H'0430	DTCEB7	
		25	H'0432	DTCEB6	
		26	H'0434	DTCEB5	
		17	H'0436	DTCEB4	
		18	H'0438	DTCEB3	
		19	H'043A	DTCEB2	
		30	H'043C	DTCEB1	
		31	H'043E	DTCEB0	
A/D	ADI	38	H'044C	DTCEC6	
TPU_0	TGI0A	40	H'0450	DTCEC5	
	TGI0B	41	H'0452	DTCEC4	
	TGI0C	42	H'0454	DTCEC3	
	TGI0D	43	H'0456	DTCEC2	
TPU_1	TGI1A	48	H'0460	DTCEC1	
	TGI1B	49	H'0462	DTCEC0	
TPU_2	TGI2A	52	H'0468	DTCED7	
	TGI2B	53	H'046A	DTCED6	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
TPU_3	TGI3A	56	H'0470	DTCED5	High
	TGI3B	57	H'0472	DTCED4	
	TGI3C	58	H'0474	DTCED3	
	TGI3D	59	H'0476	DTCED2	
TPU_4	TGI4A	64	H'0480	DTCED1	
	TGI4B	65	H'0482	DTCED0	
TPU_5	TGI5A	68	H'0488	DTCEE7	
	TGI5B	69	H'048A	DTCEE6	
TMR_0	CMIA0	72	H'0490	DTCEE3	
	CMIB0	73	H'0492	DTCEE2	
TMR_1	CMIA1	76	H'0498	DTCEE1	
	CMIB1	77	H'049A	DTCEE0	
Reserved		80	H'04A0	DTCEF7	
		81	H'04A2	DTCEF6	
		82	H'04A4	DTCEF5	
		83	H'04A6	DTCEF4	
SCI_0	RXI0	89	H'04B2	DTCEF3	
	TXI0	90	H'04B4	DTCEF2	
SCI_1	RXI1	93	H'04BA	DTCEF1	
	TXI1	94	H'04BC	DTCEF0	
SCI_2	RXI2	97	H'04C2	DTCEG7	
	TXI2	98	H'04C4	DTCEG6	Low

Note: * DTCE bits with no corresponding interrupt are reserved, and should be written with 0. When clearing the software standby state or all-module-clocks-stop mode with an interrupt, write 0 to the corresponding DTCE bit.

7.5 Operation

The DTC stores register information in the on-chip RAM. When activated, the DTC reads register information that is already stored in the on-chip RAM and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to the on-chip RAM. Pre-storage of register information in the on-chip RAM makes it possible to transfer data over any required number of channels. There are three transfer modes: normal mode, repeat mode, and block transfer mode. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation (chain transfer). A setting can also be made to have chain transfer performed only when the transfer counter value is 0. This enables DTC re-setting to be performed by the DTC itself.

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.

Figure 7.4 shows a flowchart of DTC operation, and table 7.2 summarizes the chain transfer conditions (combinations for performing the second and third transfers are omitted).

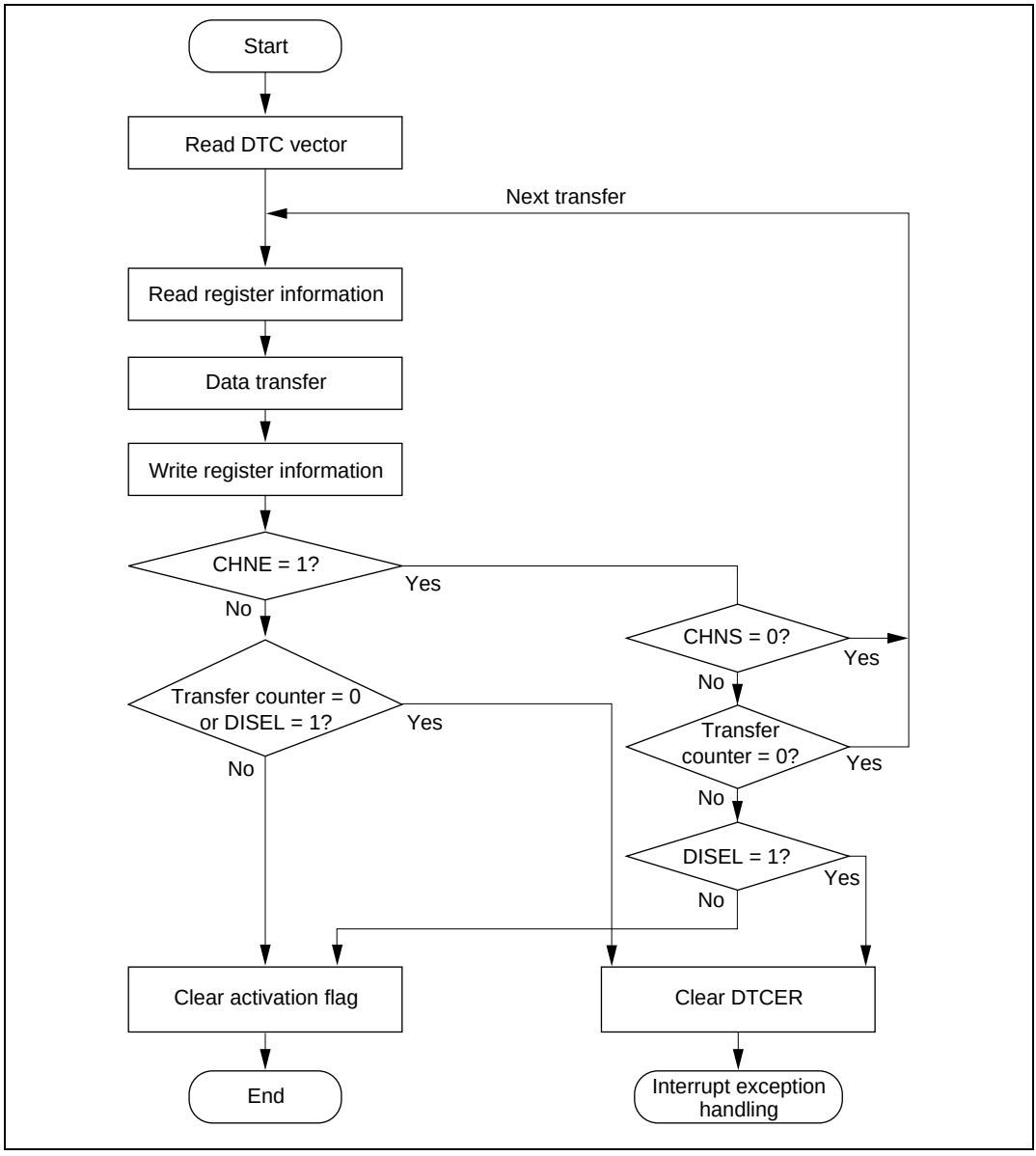


Figure 7.4 Flowchart of DTC Operation

Table 7.2 Chain Transfer Conditions

1st Transfer				2nd Transfer				
CHNE	**CHNS**	**DISEL**	**CR**	**CHNE**	**CHNS**	**DISEL**	**CR**	**DTC Transfer**
0	—	0	Not 0	—	—	—	—	Ends at 1st transfer
0	—	0	0	—	—	—	—	Ends at 1st transfer
0	—	1	—	—	—	—	—	Interrupt request to CPU
1	0	—	—	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	0	Not 0	—	—	—	—	Ends at 1st transfer
1	1	—	0	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	1	Not 0	—	—	—	—	Ends at 1st transfer Interrupt request to CPU

7.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data. Table 7.3 lists the register function in normal mode. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt can be requested.

Table 7.3 Register Function in Normal Mode

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

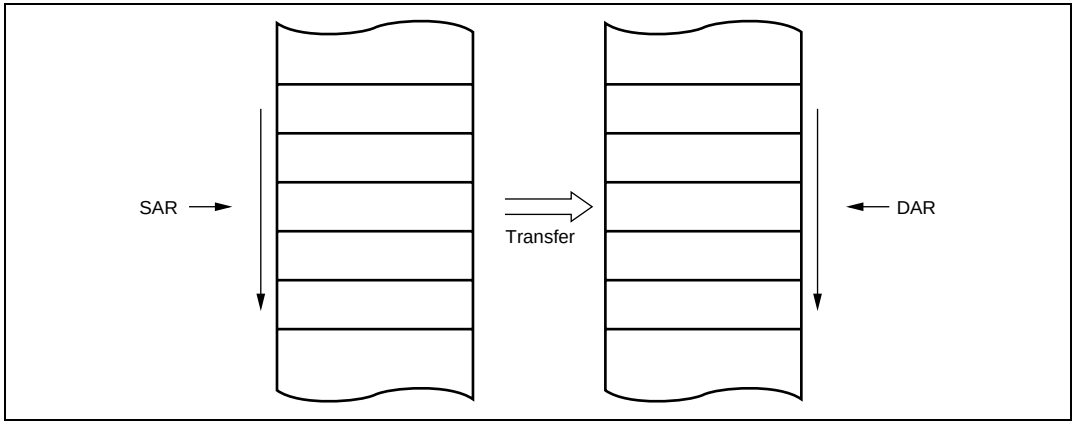


Figure 7.5 Memory Mapping in Normal Mode

7.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data. Table 7.4 lists the register function in repeat mode. From 1 to 256 transfers can be specified. Once the specified number of transfers has 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 7.4 Register Function in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates 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

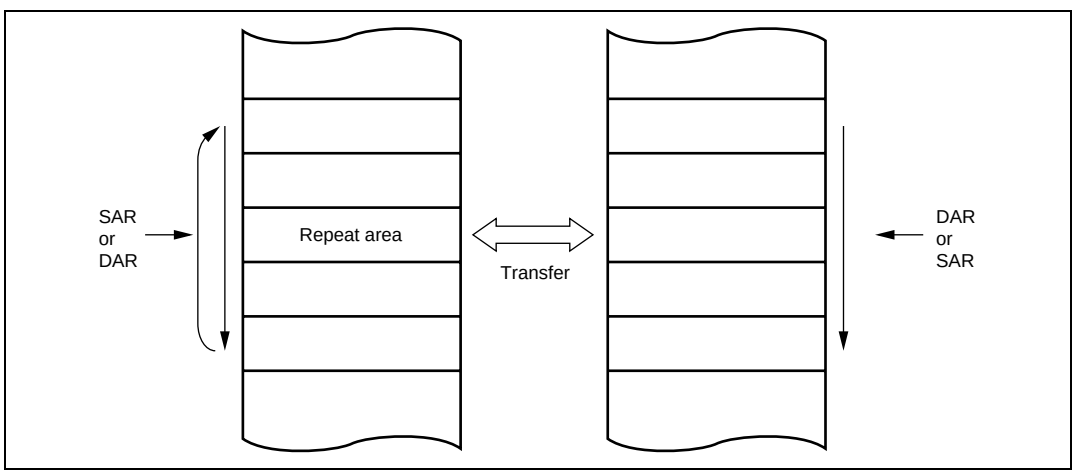


Figure 7.6 Memory Mapping in Repeat Mode

7.5.3 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. Table 7.5 lists the register function in block transfer mode.

The block size is 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. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt is requested.

Table 7.5 Register Function in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates 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

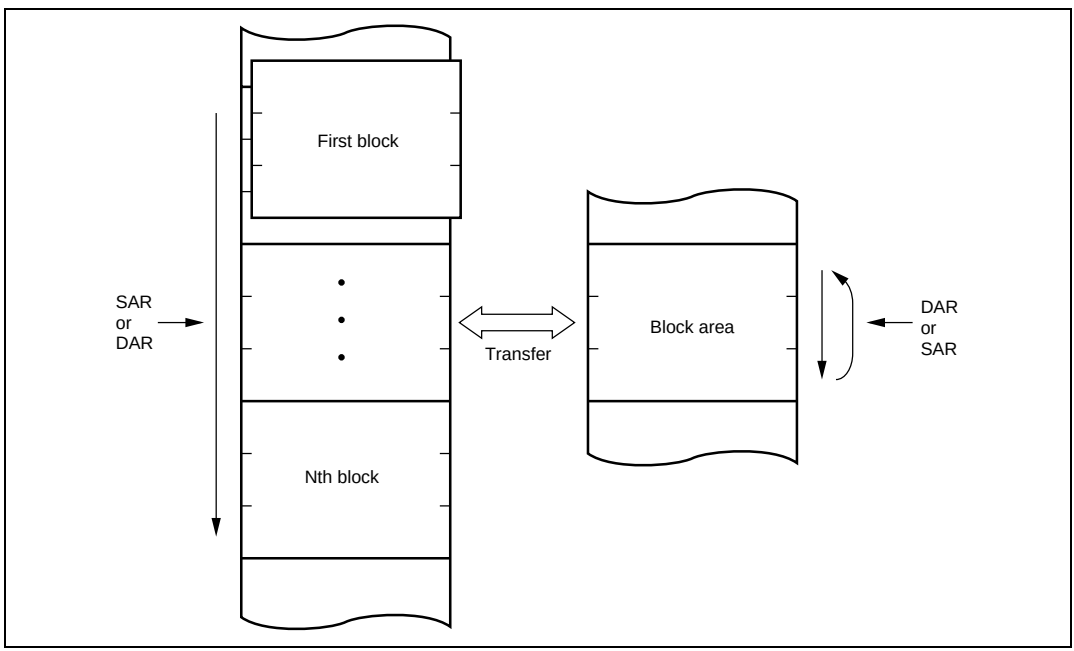


Figure 7.7 Memory Mapping in Block Transfer Mode

7.5.4 Chain Transfer

Setting the CHNE bit 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 7.8 shows the operation of chain transfer. When activated, the DTC reads the register information start address stored at the vector address, and then reads the first register information at that start address. The CHNE bit in MRB is checked after the end of data transfer, if the value is 1, the next register information, which is located consecutively, is read and transfer is performed. This operation is repeated until the end of data transfer of register information with CHNE = 0. It is also possible, by setting both the CHNE bit and CHNS bit to 1, to specify execution of chain transfer only when the transfer counter value is 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 source flag for the activation source is not affected.

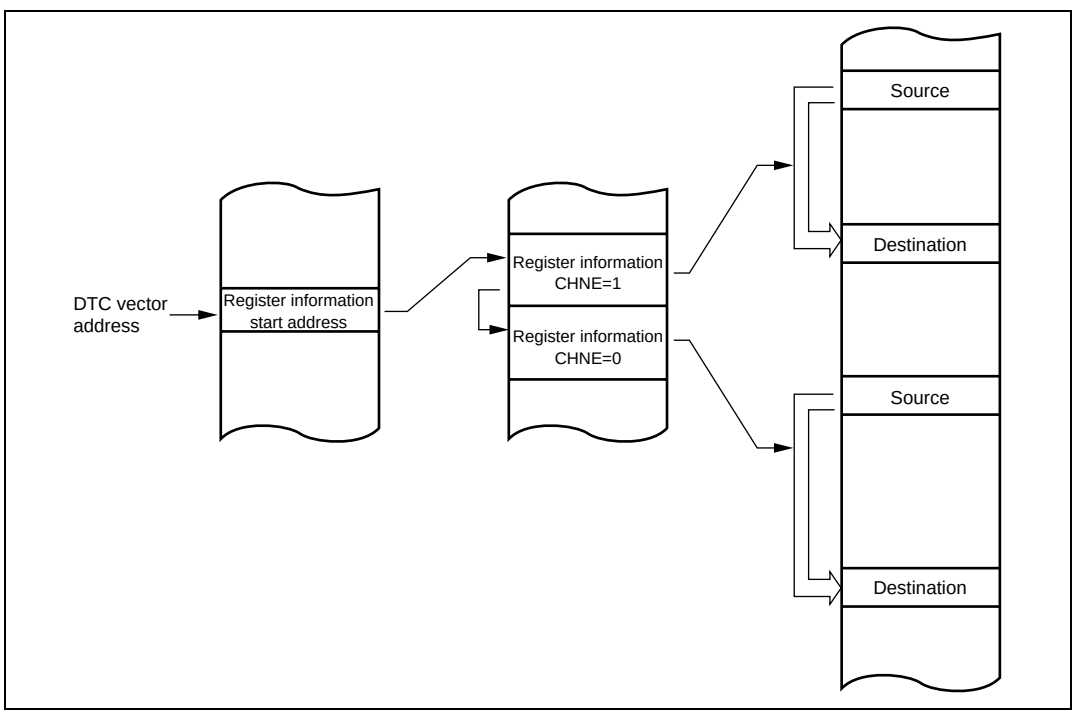


Figure 7.8 Operation of Chain Transfer

7.5.5 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes 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 activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers has ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should 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.

7.5.6 Operation Timing

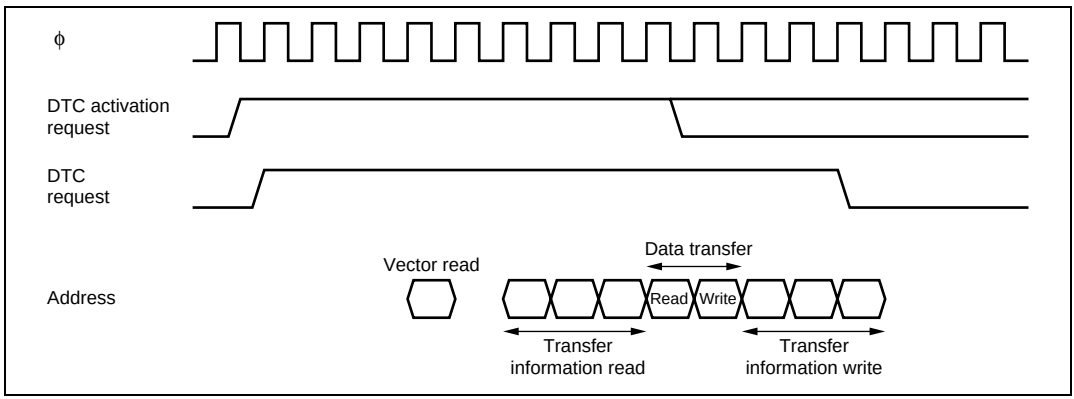


Figure 7.9 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

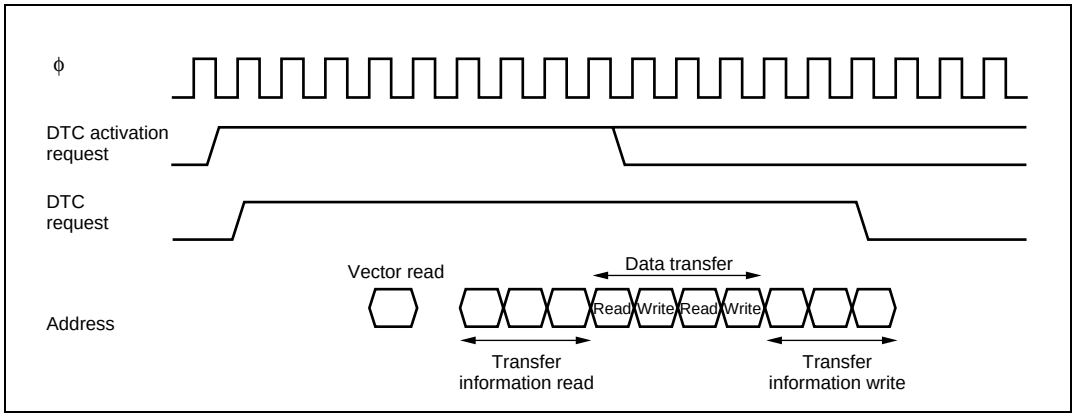


Figure 7.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

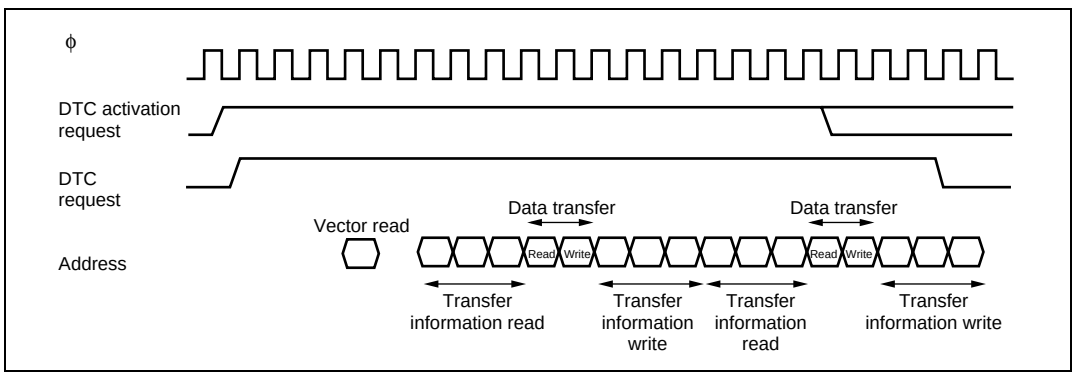


Figure 7.11 DTC Operation Timing (Example of Chain Transfer)

7.5.7 Number of DTC Execution States

Table 7.6 lists execution status for a single DTC data transfer, and table 7.7 shows the number of states required for each execution status.

Table 7.6 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 7.7 Number of States Required for Each Execution Status

Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width			32	16	8	16	8		16	
Access states			1	1	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	2	2	2	3+m	2	3+m
	Word data read	S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write	S_L	1	1	2	2	2	3+m	2	3+m
	Word data write	S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation	S_M	1							

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (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 table is located in on-chip ROM, normal mode is set, and data is transferred from the 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.

7.6 Procedures for Using DTC

7.6.1 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 the 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 the end of one data transfer, or after the specified number of data transfers have ended, 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.

7.6.2 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 the 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 the end of one data transfer, 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 have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.

7.7 Examples of Use of the DTC

7.7.1 Normal Mode

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

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at 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 reception of one byte of data ends 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 ended, 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 should perform wrap-up processing.

7.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

1. Perform settings for transfer to the PPG's NDR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.

2. Perform settings for transfer to the TPU's TGR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and the data table size in CRA. CRB can be set to any value.
3. Locate the TPU transfer register information consecutively after the NDR transfer register information.
4. Set the start address of the NDR transfer register information to the DTC vector address.
5. Set the bit corresponding to TGIA in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable the TGIA interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match to be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

7.7.3 Chain Transfer when Counter = 0

By executing a second data transfer, and performing re-setting of the first data transfer, only when the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 7.12 shows the chain transfer when the counter value is 0.

1. For the first transfer, set the normal mode for input data. Set fixed transfer source address (G/A, etc.), CRA = H'0000 (65,536 times), and CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for each of the 65,536 transfer start addresses for the first data transfer in a separate area (in ROM, etc.). For example, if the input buffer comprises H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper 8 bits of DAR in the first register information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.

4. Execute the first data transfer 65,536 times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
5. Next, execute the first data transfer the 65,536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, an interrupt request is not sent to the CPU.

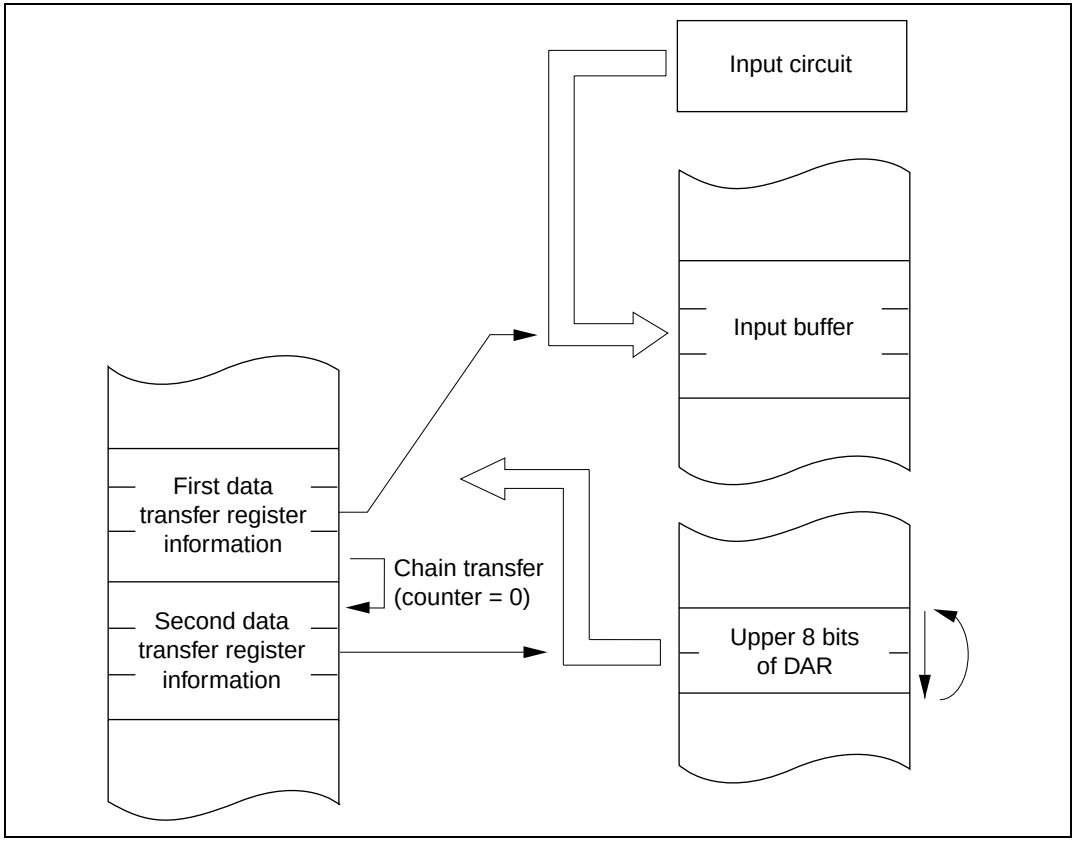


Figure 7.12 Chain Transfer when Counter = 0

7.7.4 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 destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
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 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.

7.8 Usage Notes

7.8.1 Module Stop Mode Setting

DTC operation can be disabled or enabled using the module stop control register. The initial setting is for DTC operation to be enabled. Register access is disabled by setting module stop mode. Module stop mode cannot be set while the DTC is activated. For details, refer to section 19, Power-Down Modes.

7.8.2 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 must not be cleared to 0.

7.8.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are disabled, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

- Chain Transfer

  When chain transfer is used, clearing of the activation source or DTCER is performed when the last of the chain of data transfers is executed. SCI and high-speed A/D converter interrupt/activation sources, on the other hand, are cleared when the DTC reads or writes to the prescribed register.

  Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.

Section 8 I/O Ports

Table 8.1 summarizes the port functions. The pins of each port also have other functions such as input/output or external 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 a DR or DDR register.

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

Ports 1 to 3, 5 (P50 to P53), and 6 to 8 can drive a single TTL load and 30 pF capacitive load. Ports A to H can drive a single TTL load and 50 pF capacitive load.

All of the I/O ports can drive a Darlington transistor when outputting data.

Ports 1 and 2 are Schmitt-triggered inputs. Ports 5,6, F (PF1, PF2), and H (PH2, PH3) are Schmitt-triggered inputs when used as the IRQ input.

Table 8.1 Port Functions

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 1	General I/O port also functioning as PPG outputs, and TPU I/Os	P17/PO15/TIOCB2/TCLKD P16/PO14/TIOCA2 P15/PO13/TIOCB1/TCLKC P14/PO12/TIOCA1 P13/PO11/TIOCD0/TCLKB P12/PO10/TIOCC0/TCLKA P11/PO9/TIOCB0 P10/PO8/TIOCA0					Schmitt-triggered input
Port 2	General I/O port also functioning as PPG outputs, TPU I/Os, and interrupt inputs	P27/PO7/TIOCB5 P26/PO6/TIOCA5 P25/PO5/TIOCB4 P24/PO4/TIOCA4 P23/PO3/TIOCD3 P22/PO2/TIOCC3 P21/PO1/TIOCB3 P20/PO0/TIOCA3					Schmitt-triggered input
Port 3	General I/O port also functioning as SCI I/Os	P35/SCK1 P34/SCK0 P33/RxD1 P32/RxD0/IrRxD P31/TxD1 P30/TxD0/IrTxD					Open-drain output enable
Port 4	General I/O port also functioning as A/D converter analog inputs and D/A converter analog outputs	P47/AN7/DA1 P46/AN6/DA0 P45/AN5 P44/AN4 P43/AN3 P42/AN2 P41/AN1 P40/AN0					

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port 5	General I/O port also functioning as interrupt inputs, A/D converter analog inputs, and D/A converter analog outputs	P57/AN15/DA3/$\overline{\text{IRQ7}}$ P56/AN14/DA2/$\overline{\text{IRQ6}}$ P55/AN13/$\overline{\text{IRQ5}}$ P54/AN12/$\overline{\text{IRQ4}}$					Schmitt-triggered input when used as IRQ input
	General I/O port also functioning as interrupt inputs, A/D converter analog inputs, and SCI I/Os	P53/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$ P52/SCK2/$\overline{\text{IRQ2}}$ P51/RxD2/$\overline{\text{IRQ1}}$ P50/TxD2/$\overline{\text{IRQ0}}$					Schmitt-triggered input when used as IRQ input
Port 6	General I/O port also functioning as interrupt inputs, and TMR I/Os	P65/TMO1 P64/TMO0 P63/TMCI1 P62/TMCI0 P61/TMRI1 P60/TMRI0					
Port 7	General I/O port	P75 P74 P73 P72 P71 P70			P75 P74 P73 P72 P71 P70	P75 P74 P73 P72 P71 P70	
Port 8	General I/O port as interrupt inputs	P85/$\overline{\text{IRQ5}}$ P84/$\overline{\text{IRQ4}}$ P83/$\overline{\text{IRQ3}}$ P82/$\overline{\text{IRQ2}}$ P81/$\overline{\text{IRQ1}}$ P80/$\overline{\text{IRQ0}}$			P85/$\overline{\text{IRQ5}}$ P84/$\overline{\text{IRQ4}}$ P83/$\overline{\text{IRQ3}}$ P82/$\overline{\text{IRQ2}}$ P81/$\overline{\text{IRQ1}}$ P80/$\overline{\text{IRQ0}}$	P85/$\overline{\text{IRQ5}}$ P84/$\overline{\text{IRQ4}}$ P83/$\overline{\text{IRQ3}}$ P82/$\overline{\text{IRQ2}}$ P81/$\overline{\text{IRQ1}}$ P80/$\overline{\text{IRQ0}}$	Schmitt-triggered input when used as IRQ input

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port A	General I/O port also functioning as address outputs	PA7/A23		PA7/A23	PA7/A23	PA7	Built-in input pull-up MOS Open-drain output enable
		PA6/A22		PA6/A22	PA6/A22	PA6	
		PA5/A21		PA5/A21	PA5/A21	PA5	
		A20		PA4/A20	PA4/A20	PA4	
		A19		PA3/A19	PA3/A19	PA3	
		A18		PA2/A18	PA2/A18	PA2	
		A17		PA1/A17	PA1/A17	PA1	
		A16		PA0/A16	PA0/A16	PA0	
Port B	General I/O port also functioning as address outputs	A15		PB7/A15	PB7/A15	PB7	Built-in input pull-up MOS
		A14		PB6/A14	PB6/A14	PB6	
		A13		PB5/A13	PB5/A13	PB5	
		A12		PB4/A12	PB4/A12	PB4	
		A11		PB3/A11	PB3/A11	PB3	
		A10		PB2/A10	PB2/A10	PB2	
		A9		PB1/A9	PB1/A9	PB1	
		A8		PB0/A8	PB0/A8	PB0	
Port C	General I/O port also functioning as address outputs	A7		PC7/A7	PC7/A7	PC7	Built-in input pull-up MOS
		A6		PC6/A6	PC6/A6	PC6	
		A5		PC5/A5	PC5/A5	PC5	
		A4		PC4/A4	PC4/A4	PC4	
		A3		PC3/A3	PC3/A3	PC3	
		A2		PC2/A2	PC2/A2	PC2	
		A1		PC1/A1	PC1/A1	PC1	
		A0		PC0/A0	PC0/A0	PC0	
Port D	General I/O port also functioning as data I/Os	D15			D15	PD7	Built-in input pull-up MOS
		D14			D14	PD6	
		D13			D13	PD5	
		D12			D12	PD4	
		D11			D11	PD3	
		D10			D10	PD2	
		D9			D9	PD1	
		D8			D8	PD0	

Port	Description	Modes 1 and 5	Modes 2 and 6	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port E	General I/O port also functioning as data I/Os	D7	PE7/D7	PE7/D7	PE7/D7	PE7	Built-in input pull-up MOS
		D6	PE6/D6	PE6/D6	PE6/D6	PE6	
		D5	PE5/D5	PE5/D5	PE5/D5	PE5	
		D4	PE4/D4	PE4/D4	PE4/D4	PE4	
		D3	PE3/D3	PE3/D3	PE3/D3	PE3	
		D2	PE2/D2	PE2/D2	PE2/D2	PE2	
		D1	PE1/D1	PE1/D1	PE1/D1	PE1	
		D0	PE0/D0	PE0/D0	PE0/D0	PE0	
Port F	General I/O port also functioning as interrupt inputs and bus control I/Os	PF7/ϕ			PF7/ϕ	PF7/ϕ	
		PF6/$\overline{\text{AS}}$			PF6/$\overline{\text{AS}}$	PF6	
		$\overline{\text{RD}}$			$\overline{\text{RD}}$	PF5	
		$\overline{\text{HWR}}$			$\overline{\text{HWR}}$	PF4	
		PF3/$\overline{\text{LWR}}$			PF3/$\overline{\text{LWR}}$	PF3	
		PF2			PF2	PF2	
		PF1			PF1	PF1	
		PF0/$\overline{\text{WAIT}}$			PF0/$\overline{\text{WAIT}}$	PF0	
Port G	General I/O port also functioning as bus control I/Os	PG6			PG6	PG6	
		PG5			PG5	PG5	
		PG4			PG4	PG4	
		PG3/$\overline{\text{CS3}}$			PG3/$\overline{\text{CS3}}$	PG3	
		PG2/$\overline{\text{CS2}}$			PG2/$\overline{\text{CS2}}$	PG2	
		PG1/$\overline{\text{CS1}}$			PG1/$\overline{\text{CS1}}$	PG1	
		PG0/$\overline{\text{CS0}}$			PG0/$\overline{\text{CS0}}$	PG0	
Port H	General I/O port also functioning as interrupt inputs and bus control I/Os	PH3/$\overline{\text{CS7}}$/($\overline{\text{IRQ7}}$)			PH3/$\overline{\text{CS7}}$/($\overline{\text{IRQ7}}$)	PH3/($\overline{\text{IRQ7}}$)	Schmitt-triggered input when used as IRQ input
		PH2/$\overline{\text{CS6}}$/($\overline{\text{IRQ6}}$)			PH2/$\overline{\text{CS6}}$/($\overline{\text{IRQ6}}$)	PH2/($\overline{\text{IRQ6}}$)	
		PH1/$\overline{\text{CS5}}$			PH1/$\overline{\text{CS5}}$	PH1	
		PH0/$\overline{\text{CS4}}$			PH0/$\overline{\text{CS4}}$	PH0	

8.1 Port 1

Port 1 is an 8-bit I/O port that also has other functions. The port 1 has the following registers.

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

8.1.1 Port 1 Data Direction Register (P1DDR)

The individual bits of P1DDR specify input or output for the pins of port 1.

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 function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while 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	

8.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the 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 function is specified to a general purpose I/O.
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	

8.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states.

PORT1 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	Undefined*	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	Undefined*	R	
5	P15	Undefined*	R	
4	P14	Undefined*	R	
3	P13	Undefined*	R	
2	P12	Undefined*	R	
1	P11	Undefined*	R	
0	P10	Undefined*	R	

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

8.1.4 Pin Functions

Port 1 pins also function as PPG outputs and TPU I/Os. The correspondence between the register specification and the pin functions is shown below.

P17/PO15/TIOCB2/TCLKD: The pin function is switched as shown below according to the combination of the TPU channel 2 settings (by bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bits TPSC2 to TPSC0 in TCR0 and TCR5, bit NDER15 in NDERH, and bit P17DDR.

TPU channel 2 settings	(1) in table below	(2) in table below		
P17DDR	—	0	1	1
NDER15	—	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output
		TIOCB2 input*1		
	TCLKD input*2			

Notes: 1. TIOCB2 input when MD3 to MD0 = B'0000, B000, and B'01xx and IOB3 = 1.
2. TCLKD input when the setting for either TCR0 or TCR5 is TPSC2 to TPSC0 = B'111. TCLKD input when channels 2 and 4 are set to phase counting mode.

TPU channel 2 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	PWM mode 1 output	PWM mode 2 output	—

x: Don't care

P16/PO14/TIOCA2: The pin function is switched as shown below according to the combination of the TPU channel 2 settings (by bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bit NDER14 in NDERH, and bit P16DDR.

TPU channel 2 settings	(1) in table below	(2) in table below		
P16DDR	—	0	1	1
NDER14	—	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output
		TIOCA2 input*1		

Note: 1. TIOCA2 input when MD3 to MD0 = B'0000, B'000, and B'01xx and IOB3 = 1.

TPU channel 2 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCB2 output disabled.

P15/PO13/TIOCB1/TCLKC: The pin function is switched as shown below according to the combination of the TPU channel 1 settings (by bits MD3 to MD0 in TMDR1, bits IOB3 to IOB0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bits TPSC2 to TPSC0 in TCR0, TCR2, TCR4, and TCR5, bit NDER13 in NDERH, and bit P15DDR.

TPU channel 1 settings	(1) in table below	(2) in table below		
P15DDR	—	0	1	1
NDER13	—	—	0	1
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output
		TIOCB1 input*1		
	TCLKC input*2			

Notes: 1. TIOCB1 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 to IOB0 = B'10xx.
2. TCLKC input when the setting for either TCR0 or TCR2 is TPSC2 to TPSC0 = B'110, or when the setting for either TCR4 or TCR5 is TPSC2 to TPSC0 = B'101.
TCLKC input when phase counting mode is set for channels 2 and 4.

TPU channel 1 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

P14/PO12/TIOCA1: The pin function is switched as shown below according to the combination of the TPU channel 1 settings (by bits MD3 to MD0 in TMDR1, bits IOA3 to IOA0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bit NDER12 in NDERH, and bit P14DDR.

TPU channel 1 settings	(1) in table below	(2) in table below		
P14DDR	—	0	1	1
NDER12	—	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output	PO12 output
		TIOCA1 input*1		

Note: 1. TIOCA1 input when MD3 to MD0 = B'0000, B'000, and B'01xx and IOA3 to IOA0 = B'10xx.

TPU channel 1 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCB1 output disabled.

P13/PO11/TIOCD0/TCLKB: The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0, bits IOD3 to IOD0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR2, bit NDER11 in NDERH, and bit P13DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P13DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output
		TIOCD0 input*1		
	TCLKB input*2			

Notes: 1. TIOCD0 input when MD3 to MD0 = B'0000 and IOD3 = B'10xx.
2. TCLKB input when the setting for any of TCR0 to TCR2 is TPSC2 to TPSC0 = B'101. TCLKB input when phase counting mode is set for channels 1 and 5.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

P12/PO10/TIOCC0/TCLKA: The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0, bits IOC3 to IOC0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR5, bit NDER10 in NDERH, and bit P12DDR.

<table>
<tr><td>TPU channel 0 settings</td><td>(1) in table below</td><td colspan="3">(2) in table below</td></tr>
<tr><td>P12DDR</td><td>—</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>NDER10</td><td>—</td><td>—</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCC0 output</td><td>P12 input</td><td>P12 output</td><td>PO10 output</td></tr>
<tr><td colspan="3">TIOCC0 input*1</td></tr>
<tr><td colspan="4">TCLKA input*2</td></tr>
</table>

Notes: 1. TIOCC0 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10xx.
2. TCLKA input when the setting for any of TCR0 to TCR5 is TPSC2 to TPSC0 = B'100.
TCLKA input when phase counting mode is set for channels 1 and 5.

<table>
<tr><td>TPU channel 0 settings</td><td>(2)</td><td>(1)</td><td>(2)</td><td>(1)</td><td>(1)</td><td>(2)</td></tr>
<tr><td>MD3 to MD0</td><td colspan="2">B'0000</td><td>B'001x</td><td>B'0010</td><td colspan="2">B'0011</td></tr>
<tr><td>IOC3 to IOC0</td><td>B'0000
B'0100
B'1xxx</td><td>B'0001 to B'0011
B'0101 to B'0111</td><td>B'xx00</td><td>Other than B'xx00</td><td colspan="2">Other than B'xx00</td></tr>
<tr><td>CCLR2, CCLR0</td><td>—</td><td>—</td><td>—</td><td>—</td><td>Other than B'101</td><td>B'101</td></tr>
<tr><td>Output function</td><td>—</td><td>Output compare output</td><td>—</td><td>PWM*3 mode 1 output</td><td>PWM mode 2 output</td><td>—</td></tr>
</table>

x: Don’t care

Note: 3. TIOCD0 output disabled.
Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR0.

P11/PO9/TIOCB0: The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0 and bits IOB3 to IOB0 in TIOR0H), bit NDER9 in NDERH, and bit P11DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P11DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input*		

Note: * TIOCB0 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10xx.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

P10/PO8/TIOCA0: The pin function is switched as shown below according to the combination of the TPU channel 0 settings (by bits MD3 to MD0 in TMDR0, bits IOA3 to IOA0 in TIOR0H, and bits CCLR2 to CCLR0 in TCR0), bit NDER8 in NDERH, and bit P10DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input*1		

Note: 1. TIOCA0 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10xx.

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCB0 output disabled.

8.2 Port 2

Port 2 is an 8-bit I/O port that also has other functions. The port 2 has the following registers.

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

8.2.1 Port 2 Data Direction Register (P2DDR)

The individual bits of P2DDR specify input or output for the pins of port 2.

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 function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while 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	

8.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for the 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 function is specified to a general purpose I/O.
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	

8.2.3 Port 2 Register (PORT2)

PORT2 shows the pin states.

PORT2 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P27	Undefined*	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	Undefined*	R	
5	P25	Undefined*	R	
4	P24	Undefined*	R	
3	P23	Undefined*	R	
2	P22	Undefined*	R	
1	P21	Undefined*	R	
0	P20	Undefined*	R	

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

8.2.4 Pin Functions

Port 2 pins also function as PPG outputs and TPU I/Os. The correspondence between the register specification and the pin functions is shown below.

P27/PO7/TIOCB5: The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR5, bits IOB3 to IOB0 in TIOR5, and bits CCLR1 and CCLR0 in TCR5), bit NDER7 in NDERL, and bit P27DDR.

TPU channel 5 settings	(1) in table below	(2) in table below		
P27DDR	—	0	1	1
NDER7	—	—	0	1
Pin function	TIOCB5 output	P27 input	P27 output	PO7 output
		TIOCB5 input*		

Note: * TIOCB5 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 = 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

P26/PO6/TIOCA5: The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR5, bits IOA3 to IOA0 in TIOR5, and bits CCLR1 and CCLR0 in TCR5), bit NDER6 in NDERL, and bit P26DDR.

TPU channel 5 settings	(1) in table below	(2) in table below		
P26DDR	—	0	1	1
NDER6	—	—	0	1
Pin function	TIOCA5 output	P26 input	P26 output	PO6 output
		TIOCA5 input*1		

Note: 1. TIOCA5 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 = 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCB5 output disabled.

P25/PO5/TIOCB4: The pin function is switched as shown below according to the combination of the TPU channel 4 settings (by bits MD3 to MD0 in TMDR4, bits IOB3 to IOB0 in TIOR4, and bits CCLR1 and CCLR0 in TCR4), bit NDER5 in NDERL, and bit P25DDR.

TPU channel 4 settings	(1) in table below	(2) in table below		
P25DDR	—	0	1	1
NDER5	—	—	0	1
Pin function	TIOCB4 output	P25 input	P25 output	PO5 output
		TIOCB4 input*		

Note: * TIOCB4 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 to IOB0 = B'10xx.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

P24/PO4/TIOCA4: The pin function is switched as shown below according to the combination of the TPU channel 4 settings (by bits MD3 to MD0 in TMDR4 and bits IOA3 to IOA0 in TIOR4), bit NDER4 in NDERL, and bit P24DDR.

TPU channel 4 settings	(1) in table below	(2) in table below		
P24DDR	—	0	1	1
NDER4	—	—	0	1
Pin function	TIOCA4 output	P24 input	P24 output	PO4 output
		TIOCA4 input*1		

Note: 1. TIOCA4 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 to IOA0 = B'10xx.

TPU channel 4 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCB4 output disabled.

P23/PO3/TIOCD3: The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOD3 to IOD0 in TIOR3L, and bits CCLR2 to CCLR0 in TCR3), bit NDER3 in NDERL, and bit P23DDR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P23DDR	—	0	1	1
NDER3	—	—	0	1
Pin function	TIOCD3 output	P23 input	P23 output	PO3 output
		TIOCD3 input*		

Note: * TIOCD3 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10xx.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

P22/PO2/TIOCC3: The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOC3 to IOC0 in TIOR3L, and bits CCLR2 to CCLR0 in TCR3), bit NDER2 in NDERL, and bit P22DDR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P22DDR	—	0	1	1
NDER2	—	—	0	1
Pin function	TIOCC3 output	P22 input	P22 output	PO2 output
		TIOCC3 input*1		

Note: 1. TIOCC3 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10xx.

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCD3 output disabled.
Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR3.

P21/PO1/TIOCB3: The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOB3 to IOB0 in TIOR3H, and bits CCLR2 to CCLR0 in TCR3), bit NDER1 in NDERL, and bit P21DDR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P21DDR	—	0	1	1
NDER1	—	—	0	1
Pin function	TIOCB3 output	P21 input	P21 output	PO1 output
		TIOCB3 input*		

Note: * TIOCB3 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10xx.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

P20/PO0/TIOCA3: The pin function is switched as shown below according to the combination of the TPU channel 3 settings (by bits MD3 to MD0 in TMDR3, bits IOA3 to IOA0 in TIOR3H, and bits CCLR2 to CCLR0 in TCR3), bit NDER0 in NDERL, and bit P20DDR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P20DDR	—	0	1	1
NDER0	—	—	0	1
Pin function	TIOCA3 output	P20 input	P20 output	PO0 output
		TIOCA3 input*1		

Note: 1. TIOCA3 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10xx.

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

x: Don't care

Note: 2. TIOCB3 output disabled.

8.3 Port 3

Port 3 is a 6-bit I/O port that also has other functions. The port 3 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)
- Port function control register 2 (PFCR2)

8.3.1 Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the pins of port 3.

P3DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0.
5	P35DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

8.3.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P35DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
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	

8.3.3 Port 3 Register (PORT3)

PORT3 shows the pin states.

PORT3 cannot be modified.

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

Note: * Determined by the states of pins P35 to P30.

8.3.4 Port 3 Open Drain Control Register (P3ODR)

P3ODR controls the output status for each port 3 pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P35ODR	0	R/W	Setting a P3ODR bit to 1 makes the corresponding port 3 pin an NMOS open-drain output pin, while clearing the bit to 0 makes the pin a CMOS output pin.
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	

8.3.5 Port Function Control Register 2 (PFCR2)

PFCR2 controls the I/O port.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
3	ASOE	1	R/W	$\overline{AS}$ Output Enable Selects to enable or disable the AS output pin. 0: PF6 is designated as I/O port 1: PF6 is designated as $\overline{AS}$ output pin
2	LWROE	1	R/W	$\overline{LWR}$ Output Enable Selects to enable or disable the $\overline{LWR}$ output pin. 0: PF3 is designated as I/O port 1: PF3 is designated as $\overline{LWR}$ output pin
1, 0	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

8.3.6 Pin Functions

Port 3 pins also function as SCI I/Os and a bus control signal output. The correspondence between the register specification and the pin functions is shown below.

P35/SCK1: The pin function is switched as shown below according to the combination of the C/$\overline{A}$ bit in SMR of SCI_1, bits CKE0 and CKE1, and bit P35DDR.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P35DDR	0	1	—	—	—
Pin function	P35 input	P35 output*	SCK1 output*	SCK1 output*	SCK1 input

Note: * NMOS open-drain output when P35ODR = 1.

P34/SCK0: The pin function is switched as shown below according to the combination of bit C/$\overline{A}$ in SMR of SCI_0, bits CKE0 and CKE1 in SCR, and bit P34DDR.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P34DDR	0	1	—	—	—
Pin function	P34 input	P34 output*	SCK0 output*	SCK0 output*	SCK0 input

Note: * NMOS open-drain output when P34ODR = 1.

P33/RxD1: The pin function is switched as shown below according to the combination of bit RE in SCR of SCI_1 and bit P33DDR.

RE	0		1
P33DDR	0	1	—
Pin function	P33 input	P33 output*	RxD1 input

Note: * NMOS open-drain output when P33ODR = 1.

P32/RxD0/IrRxD: The pin function is switched as shown below according to the combination of bit RE in SCR of SCI_0 and bit P32DDR.

RE	0		1
P32DDR	0	1	—
Pin function	P32 input	P32 output*	RxD0/IrRxD input

Note: * NMOS open-drain output when P32ODR = 1.

P31/TxD1: The pin function is switched as shown below according to the combination of bit TE in SCR of SCI_1 and bit P31DDR.

TE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	TxD1 output*

Note: * NMOS open-drain output when P31ODR = 1.

P30/TxD0/IrTxD: The pin function is switched as shown below according to the combination of bit TE in SCR of SCI_0 and bit P30DDR.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	RxD0/IrRxD output*

Note: * NMOS open-drain output when P30ODR = 1.

8.4 Port 4

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

- Port 4 register (PORT4)

8.4.1 Port 4 Register (PORT4)

PORT4 is an 8-bit read-only register that shows port 4 pin states.

PORT4 cannot be modified.

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

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

8.5 Port 5

Port 5 comprises a 4-bit I/O port (P53 to P50) and a 4-bit input-only port (P57 to P54). The 4-bit input-only port does not have the data direction register and data register. The port 5 has the following registers.

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

8.5.1 Port 5 Data Direction Register (P5DDR)

The individual bits of P5DDR specify input or output for the pins of port 5.

P5DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved When these bits are read, undefined value is returned.
3	P53DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

8.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	P53DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
2	P52DR	0	R/W	
1	P51DR	0	R/W	
0	P50DR	0	R/W	

8.5.3 Port 5 Register (PORT5)

PORT5 shows the pin states.

PORT5 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P57	Undefined*	R	When bits P57 to P54 are read, the pin states are always read from bits 7 to 4.
6	P56	Undefined*	R	
5	P55	Undefined*	R	
4	P54	Undefined*	R	
3	P53	Undefined*	R	If bits P53 to P50 are read 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.
2	P52	Undefined*	R	
1	P51	Undefined*	R	
0	P50	Undefined*	R	

Note: * Determined by the states of pins P57 to P50.

8.5.4 Pin Functions

Port 5 pins also function as SCI I/Os, A/D converter inputs, A/D converter analog inputs, D/A converter analog outputs, and interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

P57/AN15/DA3/$\overline{\text{IRQ7}}$: The pin function is switched as shown below according to bit ITS7 in ITSR.

Pin function	
	$\overline{\text{IRQ7}}$ interrupt input pin*
	AN15 input
	DA3 output

Note: * $\overline{\text{IRQ7}}$ input when ITS7 = 0.

P56/AN14/DA2/$\overline{\text{IRQ6}}$: The pin function is switched as shown below according to bit ITS6 in ITSR.

Pin function	$\overline{\text{IRQ6}}$ interrupt input pin*
	AN14 input
	DA2 output

Note: * $\overline{\text{IRQ6}}$ input when ITS6 = 0.

P55/AN13/$\overline{\text{IRQ5}}$: The pin function is switched as shown below according to bit ITS5 in ITSR.

Pin function	$\overline{\text{IRQ5}}$ interrupt input*
	AN13 input

Note: * $\overline{\text{IRQ5}}$ input when ITS5 = 0.

P54/AN12/$\overline{\text{IRQ4}}$: The pin function is switched as shown below according to bit ITS4 in ITSR.

Pin function	$\overline{\text{IRQ4}}$ interrupt input*
	AN12 input

Note: * $\overline{\text{IRQ4}}$ input when ITS4 = 0.

P53/$\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$: The pin function is switched as shown below according to the combination of bits TRGS1 and TRGS0 in the A/D control register (ADCR), bit ITS3 in ITSR, and bit P53DDR.

P53DDR	0	1
Pin function	P53 input	P53 output
	$\overline{\text{ADTRG}}$ input*1	
	$\overline{\text{IRQ3}}$ interrupt input*2	

Notes: 1. $\overline{\text{ADTRG}}$ input when TRGS1 = TRGS0 = 0.
2. $\overline{\text{IRQ3}}$ input when ITS3 = 0.

P52/SCK2/$\overline{\text{IRQ2}}$: The pin function is switched as shown below according to the combination of bit C/A in SMR of SCI_2, bits CKE0 and CKE1 in SCR, bit ITS2 in ITSR, and bit P52DDR.

CKE1	0				1
C/$\overline{\text{A}}$	0			1	—
CKE0	0		1	—	—
P52DDR	0	1	—	—	—
Pin function	P52 input	P52 output	SCK2 output	SCK2 output	SCK2 input
	$\overline{\text{IRQ2}}$ interrupt input*				

Note: * $\overline{\text{IRQ2}}$ input when ITS2 = 0.

P51/RxD2/$\overline{\text{IRQ1}}$: The pin function is switched as shown below according to the combination of bit RE in SCR of SCI_2, bit ITS1 in ITSR, and bit P51DDR.

RE	0		1
P51DDR	0	1	—
Pin function	P51 input	P51 output	RxD2 input
	$\overline{\text{IRQ1}}$ interrupt input*		

Note: * $\overline{\text{IRQ1}}$ input when ITS1 = 0.

P50/TxD2/$\overline{\text{IRQ0}}$: The pin function is switched as shown below according to the combination of bit TE in SCR of SCI_2, bit ITS0 in ITSR, and bit P50DDR.

TE	0		1
P50DDR	0	1	—
Pin function	P50 input	P50 output	TxD2 input
	$\overline{\text{IRQ0}}$ interrupt input*		

Note: * $\overline{\text{IRQ0}}$ input when ITS0 = 0.

8.6 Port 6

Port 6 is a 6-bit I/O port that also has other functions. The port 6 has the following registers.

- Port 6 data direction register (P6DDR)
- Port 6 data register (P6DR)
- Port 6 register (PORT6)

8.6.1 Port 6 Data Direction Register (P6DDR)

The individual bits of P6DDR specify input or output for the pins of port 6.

P6DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0.
5	P65DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P64DDR	0	W	
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

8.6.2 Port 6 Data Register (P6DR)

P6DR stores output data for the port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P65DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
4	P64DR	0	R/W	
3	P63DR	0	R/W	
2	P62DR	0	R/W	
1	P61DR	0	R/W	
0	P60DR	0	R/W	

8.6.3 Port 6 Register (PORT6)

PORT6 shows the pin states.

PORT6 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are reserved, if read they will return an undefined value.
5	P65	Undefined*	R	If a port 6 read is performed while P6DDR bits are set to 1, the P6DR values are read. If a port 6 read is performed while P6DDR bits are cleared to 0, the pin states are read.
4	P64	Undefined*	R	
3	P63	Undefined*	R	
2	P62	Undefined*	R	
1	P61	Undefined*	R	
0	P60	Undefined*	R	

Note: * Determined by the states of pins P65 to P60.

8.6.4 Pin Functions

Port 6 pins function as 8-bit timer I/Os. The correspondence between the register specification and the pin functions is shown below.

P65/TMO1: The pin function is switched as shown below according to the combination of bits OS3 to OS0 in TCSR1 of the 8-bit timer, and bit P65DDR.

OS3 to OS0	All 0		Not all 0
P65DDR	0	1	—
Pin function	P65 input	P65 output	TMO1 output

P64/TMO0: The pin function is switched as shown below according to the combination of bits OS3 to OS0 in TCSR1 of the 8-bit timer, and bit P64DDR.

OS3 to OS0	All 0		Not all 0
P64DDR	0	1	—
Pin function	P64 input	P64 output	TMO0 output

P63/TMCI1: The pin function is switched as shown below according to the bit P63DDR.

P63DDR	0	1
Pin function	P63 input	P63 output
	TMCI1 input*	

Note: * When used as the external clock input pin of TMR, the external clock is selected by the CKS2 to CKS0 bits of TCR_1.

P62/TMCI0: The pin function is switched as shown below according to the bit P62DDR.

P62DDR	0	1
Pin function	P62 input	P62 output
	TMCI0 input*	

Note: * When used as the external clock input pin of TMR, the external clock is selected by the CKS2 to CKS0 bits of TCR_0.

P61/TMRI1: The pin function is switched as shown below according to the combination of bit P61DDR.

P61DDR	0	1
Pin function	P61 input	P61 output
	TMRI1 input*	

Note: * When used as the counter reset of TMR, the CCLR1 and CCLR0 bits of TCR_1 are both set to 1.

P60/TMRI0: The pin function is switched as shown below according to the bit P60DDR.

P60DDR	0	1
Pin function	P60 input	P60 output
	TMRI0 input*	

Note: * When used as the counter reset of TMR, the CCLR1 and CCLR0 bits of TCR_0 are respectively set to 1.

8.7 Port 7

Port 7 is a 6-bit I/O port that also has other functions. The port 7 has the following registers.

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

8.7.1 Port 7 Data Direction Register (P7DDR)

The individual bits of P7DDR specify input or output for the pins of port 7.

P7DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0.
5	P75DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P74DDR	0	W	
3	P73DDR	0	W	
2	P72DDR	0	W	
1	P71DDR	0	W	
0	P70DDR	0	W	

8.7.2 Port 7 Data Register (P7DR)

P7DR stores output data for the port 7 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P75DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
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	

8.7.3 Port 7 Register (PORT7)

PORT7 shows the pin states.

PORT7 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are reserved, if read they will return an undefined value.
5	P75	Undefined*	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.
4	P74	Undefined*	R	
3	P73	Undefined*	R	
2	P72	Undefined*	R	
1	P71	Undefined*	R	
0	P70	Undefined*	R	

Note: * Determined by the states of pins P75 to P70.

8.8 Port 8

Port 8 is a 6-bit I/O port that also has other functions. The port 8 has the following registers.

- Port 8 data direction register (P8DDR)
- Port 8 data register (P8DR)
- Port 8 register (PORT8)

8.8.1 Port 8 Data Direction Register (P8DDR)

The individual bits of P8DDR specify input or output for the pins of port 8.

P8DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0.
5	P85DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

8.8.2 Port 8 Data Register (P8DR)

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0 and cannot be modified.
5	P85DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
4	P84DR	0	R/W	
3	P83DR	0	R/W	
2	P82DR	0	R/W	
1	P81DR	0	R/W	
0	P80DR	0	R/W	

8.8.3 Port 8 Register (PORT8)

PORT8 shows the pin states.

PORT8 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved
6	—	Undefined	—	These bits are reserved, if read they will return an undefined value.
5	P85	Undefined*	R	If a port 8 read is performed while P8DDR bits are set to 1, the P8DR values are read. If a port 8 read is performed while P8DDR bits are cleared to 0, the pin states are read.
4	P84	Undefined*	R	
3	P83	Undefined*	R	
2	P82	Undefined*	R	
1	P81	Undefined*	R	
0	P80	Undefined*	R	

Note: * Determined by the states of pins P85 to P80.

8.8.4 Pin Functions

Port 8 pins also function as interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

P85/$\overline{\text{IRQ5}}$: The pin function is switched as shown below according to the combination of bit P85DDR, and bit ITS5 in ITSR.

P85DDR	0	1
Pin function	P85 input	P85 output
	$\overline{\text{IRQ5}}$ interrupt input*	

Note: * $\overline{\text{IRQ5}}$ input when ITS5 = 1.

P84/$\overline{\text{IRQ4}}$: The pin function is switched as shown below according to the combination of bit P84DDR, and bit ITS4 in ITSR.

P84DDR	0	1
Pin function	P84 input	P84 output
	$\overline{\text{IRQ4}}$ interrupt input*	

Note: * $\overline{\text{IRQ4}}$ input when ITS4 = 1.

P83/$\overline{\text{IRQ3}}$: The pin function is switched as shown below according to the combination of bit P83DDR, and bit ITS3 in ITSR.

P83DDR	0	1
Pin function	P83 input	P83 output
	$\overline{\text{IRQ3}}$ interrupt input*	

Note: * $\overline{\text{IRQ3}}$ input when ITS3 = 1.

P82/$\overline{\text{IRQ2}}$: The pin function is switched as shown below according to the combination of bit P82DDR, and bit ITS2 in ITSR.

P82DDR	0	1
Pin function	P82 input	P82 output
	$\overline{\text{IRQ2}}$ interrupt input*	

Note: * $\overline{\text{IRQ2}}$ input when ITS2 = 1.

P81/$\overline{\text{IRQ1}}$: The pin function is switched as shown below according to the combination of bit P81DDR and bit ITS1 in ITSR.

P81DDR	0	1
Pin function	P81 input	P81 output
	$\overline{\text{IRQ1}}$ interrupt input*	

Note: * $\overline{\text{IRQ1}}$ input when ITS1 = 1.

P80/$\overline{\text{IRQ0}}$: The pin function is switched as shown below according to the combination of bit P80DDR and bit ITS0 in ITSR.

P80DDR	0	1
Pin function	P80 input	P80 output
	$\overline{\text{IRQ0}}$ interrupt input*	

Note: * $\overline{\text{IRQ0}}$ input when ITS0 = 1.

8.9 Port A

Port A is an 8-bit I/O port that also has other functions. The port A 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)
- Port function control register 1 (PFCR1)

8.9.1 Port A Data Direction Register (PADDR)

The individual bits of PADDR specify input or output for the pins of port A. 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	• Modes 1, 2, 5, and 6 Pins PA4 to PA0 are address outputs regardless of the PADDR settings. For pins PA7 to PA5, when the corresponding bit of A23E to A21E is set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing one of bits A23E to A21E to 0 makes the corresponding port A pin an I/O port, and its function can be switched with PADDR. • Mode 4 When the corresponding bit of A23E to A16E is set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing one of bits A23E to A16E to 0 makes the corresponding port A pin an I/O port, and its function can be switched with PADDR. • Mode 7 (when EXPE = 1) When the corresponding bit of A23E to A16E is set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing one of bits A23E to A16E to 0 makes the corresponding port A pin an I/O port; setting the corresponding PADDR bit to 1 makes the pin an output port, while clearing the bit to 0 makes the pin an input port. • Mode 7 (when EXPE = 0) Port A is an I/O port, and its pin functions can be switched with PADDR.
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	

8.9.2 Port A Data Register (PADR)

PADR stores output data for the 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 function is specified to a general purpose I/O.
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	

8.9.3 Port A Register (PORTA)

PORTA shows port A pin states.

PORTA cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7	Undefined*	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	Undefined*	R	
5	PA5	Undefined*	R	
4	PA4	Undefined*	R	
3	PA3	Undefined*	R	
2	PA2	Undefined*	R	
1	PA1	Undefined*	R	
0	PA0	Undefined*	R	

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

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

PAPCR controls the input pull-up MOS function. Bits 7 to 5 are valid in modes 1, 2, 5, and 6, and all the bits are valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PCR	0	R/W	When a pin function is specified to 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	

8.9.5 Port A Open Drain Control Register (PAODR)

PAODR specifies an output type of port A.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	Setting the corresponding bit to 1 specifies a pin output type to NMOS open-drain output, while clearing this bit to 0 specifies that to CMOS output.
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	

8.9.6 Port Function Control Register 1 (PFCR1)

PFCR1 performs I/O port control. Bits 7 to 5 are valid in modes 1, 2, 5, and 6, and all the bits are valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	A23E	1	R/W	Address 23 Enable Enables or disables output for address output 23 (A23). 0: DR output when PA7DDR = 1 1: A23 output when PA7DDR = 1
6	A22E	1	R/W	Address 22 Enable Enables or disables output for address output 22 (A22). 0: DR output when PA6DDR = 1 1: A22 output when PA6DDR = 1
5	A21E	1	R/W	Address 21 Enable Enables or disables output for address output 21 (A21). 0: DR output when PA5DDR = 1 1: A21 output when PA5DDR = 1
4	A20E	1	R/W	Address 20 Enable Enables or disables output for address output 20 (A20). 0: DR output when PA4DDR = 1 1: A20 output when PA4DDR = 1
3	A19E	1	R/W	Address 19 Enable Enables or disables output for address output 19 (A19). 0: DR output when PA3DDR = 1 1: A19 output when PA3DDR = 1
2	A18E	1	R/W	Address 18 Enable Enables or disables output for address output 18 (A18). 0: DR output when PA2DDR = 1 1: A18 output when PA2DDR = 1
1	A17E	1	R/W	Address 17 Enable Enables or disables output for address output 17 (A17). 0: DR output when PA1DDR = 1 1: A17 output when PA1DDR = 1
0	A16E	1	R/W	Address 16 Enable Enables or disables output for address output 16 (A16). 0: DR output when PA0DDR = 1 1: A16 output when PA0DDR = 1

8.9.7 Pin Functions

Port A pins also function as address outputs. The correspondence between the register specification and the pin functions is shown below.

PA7/A23, PA6/A22, PA5/A21: The pin function is switched as shown below according to the operating mode, bit EXPE, bits A23E to A21E, and bit PADDR.

Operating mode	1, 2, 4, 5, 6				7					
EXPE	—				0		1			
AxxE	0		1		—		0		1	
PADDR	0	1	0	1	0	1	0	1	0	1
Pin function	PA input	PA output	PA input	Address output	PA input	PA output	PA input	PA output	PA input	Address output

PA4/A20, PA3/A19, PA2/A18, PA1/A17, PA20/A16: The pin function is switched as shown below according to the operating mode, bit EXPE, bits A20E to A16E, and bit PADDR.

Operating mode	1, 2, 5, 6	4				7					
EXPE	—	—				0		1			
AxxE	—	0		1		—		0		1	
PADDR	—	0	1	0	1	0	1	0	1	0	1
Pin function	Address output	PA input	PA output	PA input	Address output	PA input	PA output	PA input	PA output	PA input	Address output

8.9.8 Port A Input Pull-Up MOS States

Port A has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used by pins PA7 to PA5 in modes 1, 2, 5, and 6, and by all pins in modes 4 and 7. Input pull-up MOS can be specified as on or off on a bit-by-bit basis.

Table 8.2 summarizes the input pull-up MOS states.

Table 8.2 Input Pull-Up MOS States (Port A)

Mode		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
4, 7	PA7 to PA0	Off	Off	On/Off	On/Off
1, 2, 5, 6	PA7 to PA5			On/Off	On/Off
	PA4 to PA0			Off	Off

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

8.10 Port B

Port B is an 8-bit I/O port that also has other functions. The port B 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)

8.10.1 Port B Data Direction Register (PBDDR)

The individual bits of PBDDR specify input or output for the pins of port B.

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	• Modes 1, 2, 5, and 6 Port B pins are address outputs regardless of the PBDDR settings. • Modes 4 and 7 (when EXPE = 1) Setting a PBDDR bit to 1 makes the corresponding port B pin an address output, while clearing the bit to 0 makes the pin an input port. • Mode 7 (when EXPE = 0) Port B is an I/O port, and its pin functions can be switched with PBDDR.
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	

8.10.2 Port B Data Register (PBDR)

PBDR is stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
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	

8.10.3 Port B Register (PORTB)

PORTB shows port B pin states.

PORTB cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	Undefined*	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	Undefined*	R	
5	PB5	Undefined*	R	
4	PB4	Undefined*	R	
3	PB3	Undefined*	R	
2	PB2	Undefined*	R	
1	PB1	Undefined*	R	
0	PB0	Undefined*	R	

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

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

PBPCR controls the on/off state of input pull-up MOS of port B. PBPCR is valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When a pin function is specified to 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	

8.10.5 Pin Functions

Port B pins also function as address outputs. The correspondence between the register specification and the pin functions is shown below.

PB7/A15, PB6/A14, PB5/A13, PB4/A12, PB3/A11, PB2/A10, PB1/A9, PB0/A8: The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PBDDR.

Operating mode	1, 2, 5, 6	4		7			
EXPE	—	—		0		1	
PBDDR	—	0	1	0	1	0	1
Pin function	Address output	PB input	Address output	PB input	PB output	PB input	Address output

8.10.6 Port B Input Pull-Up MOS States

Port B has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in modes 4 and 7. Input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In modes 3, 4 and 7, when a PBDDR bit is cleared to 0, setting the corresponding PBPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 8.3 summarizes the input pull-up MOS states.

Table 8.3 Input Pull-Up MOS States (Port B)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 5, 6	Off	Off	Off	Off
4, 7			On/Off	On/Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: On when PBDDR = 0 and PBPCR = 1; otherwise off.

8.11 Port C

Port C is an 8-bit I/O port that also has other functions. The port C 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)

8.11.1 Port C Data Direction Register (PCDDR)

The individual bits of PCDDR specify input or output for the pins of port C.

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	• Modes 1, 2, 5, and 6 Port C pins are address outputs regardless of the PCDDR settings. • Modes 4 and 7 (when EXPE = 1) Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, while clearing the bit to 0 makes the pin an input port. • Mode 7 (when EXPE = 0) Port C is an I/O port, and its pin functions can be switched with PCDDR.
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	

8.11.2 Port C Data Register (PCDR)

PCDR stores output data for the 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 function is specified to a general purpose I/O.
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	

8.11.3 Port C Register (PORTC)

PORTC is shows port C pin states.

PORTC cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	Undefined*	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	Undefined*	R	
5	PC5	Undefined*	R	
4	PC4	Undefined*	R	
3	PC3	Undefined*	R	
2	PC2	Undefined*	R	
1	PC1	Undefined*	R	
0	PC0	Undefined*	R	

Note: * Determined by the states of pins PC7 to PC0.

8.11.4 Port C Pull-Up MOS Control Register (PCPCR)

PCPCR controls the on/off state of input pull-up MOS of port C. PCPCR is valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When a pin function is specified to 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	

8.11.5 Pin Functions

Port C pins also function as address outputs. The correspondence between the register specification and the pin functions is 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 operating mode, bit EXPE, and bit PCDDR.

Operating mode	1, 2, 5, 6	4		7			
EXPE	—	—		0		1	
PCDDR	—	0	1	0	1	0	1
Pin function	Address output	PC input	Address output	PC input	PC output	PC input	Address output

8.11.6 Port C Input Pull-Up MOS States

Port C has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in modes 4 and 7. Input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In modes 4 and 7, when a PCDDR bit is cleared to 0, setting the corresponding PCPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 8.4 summarizes the input pull-up MOS states.

Table 8.4 Input Pull-Up MOS States (Port C)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 5, 6	Off	Off	Off	Off
4, 7			On/Off	On/Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: On when PCDDR = 0 and PCPCR = 1; otherwise off.

8.12 Port D

Port D is an 8-bit I/O port that also has other functions. The port D 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)

8.12.1 Port D Data Direction Register (PDDDR)

The individual bits of PDDDR specify input or output for the pins of port D.

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	• Modes 1, 2, 4, 5, 6, and 7 (when EXPE = 1) Port D is automatically designated for data input/output. • Mode 7 (when EXPE = 0) Port D is an I/O port, and its pin functions can be switched with PDDDR.
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	

8.12.2 Port D Data Register (PDDR)

PDDR stores output data for the 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 function is specified to a general purpose I/O.
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	

8.12.3 Port D Register (PORTD)

PORTD shows port D pin states.

PORTD cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	Undefined*	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	Undefined*	R	
5	PD5	Undefined*	R	
4	PD4	Undefined*	R	
3	PD3	Undefined*	R	
2	PD2	Undefined*	R	
1	PD1	Undefined*	R	
0	PD0	Undefined*	R	

Note: * Determined by the states of pins PD7 to PD0.

8.12.4 Port D Pull-up Control Register (PDPCR)

PDPCR controls on/off states of the input pull-up MOS of port D. PDPCR is valid in mode 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When the pin is in its input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
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	

8.12.5 Pin Functions

Port D pins also function as data I/Os. The correspondence between the register specification and the pin functions is 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 operating mode, bit EXPE, and bit PDDDR.

Operating mode	1, 2, 4, 5, 6	7		
EXPE	—	0		1
PDDDR	—	0	1	—
Pin function	Data I/O	PD input	PD output	Data I/O

8.12.6 Port D Input Pull-Up MOS States

Port D has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in mode 7. Input pull-up MOS can be specified as on or off on a bit-by-bit basis.

In mode 7, when a PDDDR bit is cleared to 0, setting the corresponding PDPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 8.5 summarizes the input pull-up MOS states.

Table 8.5 Input Pull-Up MOS States (Port D)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 4, 5, 6	Off	Off	Off	Off
7			On/Off	On/Off

Legend:

OFF: Input pull-up MOS is always off.

On/Off: On when PDDDR = 0 and PDPCR = 1; otherwise off.

8.13 Port E

Port E is an 8-bit I/O port that also has other functions. The port E 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)

8.13.1 Port E Data Direction Register (PEDDR)

The individual bits of PEDDR specify input or output for the pins of port E.

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	• Modes 1, 2, 4, 5, and 6 When 8-bit bus mode is selected, port E functions as an I/O port. The pin states can be changed with PEDDR. When 16-bit bus mode is selected, port E is designated for data input/output. For details on 8-bit and 16-bit bus modes, see section 6, Bus Controller. • Mode 7 (when EXPE = 1) When 8-bit bus mode is selected, port E functions as an I/O port. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port. When 16-bit bus mode is selected, port E is designated for data input/output. • Mode 7 (when EXPE = 0) Port E is an I/O port, and its pin functions can be switched with PEDDR.
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	

8.13.2 Port E Data Register (PEDR)

PEDR stores output data for the 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 function is specified to a general purpose I/O.
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	

8.13.3 Port E Register (PORTE)

PORTE shows port E pin states.

PORTE cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	Undefined*	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	Undefined*	R	
5	PE5	Undefined*	R	
4	PE4	Undefined*	R	
3	PE3	Undefined*	R	
2	PE2	Undefined*	R	
1	PE1	Undefined*	R	
0	PE0	Undefined*	R	

Note: * Determined by the states of pins PE7 to PE0.

8.13.4 Port E Pull-up Control Register (PEPCR)

PEPCR controls on/off states of the input pull-up MOS of port E. PEPCR is valid in 8-bit bus mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PCR	0	R/W	When the pin is in its input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
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	

8.13.5 Pin Functions

Port E pins also function as data I/Os. The correspondence between the register specification and the pin functions is 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 operating mode, bus mode, bit EXPE, and bit PEDDR.

Operating mode	1, 2, 4, 5, 6			7				
Bus mode	All areas 8-bit space		At least one area 16-bit space	—		All areas 8-bit space		At least one area 16-bit space
EXPE	—		—	0		1		1
PEDDR	0	1	—	0	1	0	1	—
Pin function	PE input	PE output	Data I/O	PE input	PE output	PE input	PE output	Data I/O

8.13.6 Port E Input Pull-Up MOS States

Port E has a built-in input pull-up MOS function that can be controlled by software. This input pull-up MOS function can be used in 8-bit bus mode. Input pull-up MOS can be specified as on or off on a bit-by-bit basis. In 8-bit bus mode, when a PEDDR bit is cleared to 0, setting the corresponding PEPCR bit to 1 turns on the input pull-up MOS for that pin.

Table 8.6 summarizes the input pull-up MOS states.

Table 8.6 Input Pull-Up MOS States (Port E)

Mode		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 4 to 7	8-bit bus	Off	Off	On/Off	On/Off
	16-bit bus			Off	Off

Legend:
Off: Input pull-up MOS is always off.
On/Off: On when PEDDR = 0 and PEPCR = 1; otherwise off.

8.14 Port F

Port F is an 8-bit I/O port that also has other functions. The port F has the following registers. For details on the port function control register 2, refer to section 8.3.5, Port Function Control Register 2 (PFCR2).

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)
- Port Function Control Register 2 (PFCR2)

8.14.1 Port F Data Direction Register (PFDDR)

The individual bits of PFDDR specify input or output for the pins of port F.
PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	1/0*	W	• Modes 1, 2, 4, 5, and 6 Pin PF7 functions as the φ output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. Pin PF6 functions as the $\overline{AS}$ output pin when ASOE is set to 1. When ASOE is cleared to 0, pin PF6 is an I/O port and its function can be switched with PF6DDR. Pins PF5 and PF4 are automatically designated as bus control outputs ($\overline{RD}$ and $\overline{HWR}$). Pin PF3 functions as the $\overline{LWR}$ output pin when LWROE is set to 1. When LWROE is cleared to 0, pin PF3 is an I/O port and its function can be switched with PF3DDR. Pins PF2 and PF1 are input/output port and their functions are switched with PFDDR. Pin PF0 functions as bus control input/output pin ($\overline{WAIT}$) when the appropriate bus controller settings are made. Otherwise, this pin is output port when the corresponding PFDDR bit is set to 1, and input port when the bit is cleared to 0. • Mode 7 (when EXPE = 1) Pin PF7 to PF1 function in the same way as in modes 1, 2, 4, 5, and 6. Pin PF0 functions as bus control input/output pin ($\overline{WAIT}$) when the appropriate PFCR2 settings are made. Otherwise, this pin is I/O port, and this function can be switched with PFDDR. • Mode 7 (when EXPE = 0) Pin PF7 functions as the φ output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. Pins PF6 to PF0 are I/O ports, and their functions can be switched with PFDDR.
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 modes 1, 2, 4, 5, and 6, and to 0 in mode 7.

8.14.2 Port F Data Register (PFDR)

PFDR stores output data for the 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 function is specified to a general purpose I/O.
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	

8.14.3 Port F Register (PORTF)

PORTF shows port F pin states.

PORTF cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	Undefined*	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	Undefined*	R	
5	PF5	Undefined*	R	
4	PF4	Undefined*	R	
3	PF3	Undefined*	R	
2	PF2	Undefined*	R	
1	PF1	Undefined*	R	
0	PF0	Undefined*	R	

Note: * Determined by the states of pins PF7 to PF0.

8.14.4 Pin Functions

Port F pins also function as external interrupt inputs, bus control signal I/Os, and system clock outputs (φ). The correspondence between the register specification and the pin functions is shown below.

PF7/φ: The pin function is switched as shown below according to bit PF7DDR.

Operating mode	1, 2, 4 to 7	
PFDDR	0	1
Pin function	PF7 input	φ output

PF6/$\overline{\text{AS}}$: The pin function is switched as shown below according to the operating mode, bit ASOE, bit EXPE, and bit PF6DDR.

Operating mode	1, 2, 4, 5, 6			7				
EXPE	—			0		1		
ASOE	1	0		—		1	0	
PF6DDR	—	0	1	0	1	—	0	1
Pin function	$\overline{\text{AS}}$ output	PF6 input	PF6 output	PF6 input	PF6 output	$\overline{\text{AS}}$ output	PF6 input	PF6 output

PF5/$\overline{\text{RD}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PF5DDR.

Operating mode	1, 2, 4, 5, 6	7		
EXPE	—	0		1
PF5DDR	—	0	1	—
Pin function	$\overline{\text{RD}}$ output	PF5 input	PF5 output	$\overline{\text{RD}}$ output

PF4/$\overline{\text{HWR}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PF4DDR.

Operating mode	1, 2, 4, 5, 6	7		
EXPE	—	0		1
PF4DDR	—	0	1	—
Pin function	$\overline{\text{HWR}}$ output	PF4 input	PF4 output	$\overline{\text{HWR}}$ output

PF3/$\overline{\text{LWR}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit PF3DDR, and bit LWROE.

Operating mode	1, 2, 4, 5, 6			7				
EXPE	—			0		1		
LWROD	1	0		—		1	0	
PF3DDR	—	0	1	0	1	—	0	1
Pin function	$\overline{\text{LWR}}$ output	PF3 input	PF3 output	PF3 input	PF3 output	$\overline{\text{LWR}}$ output	PF3 input	PF3 output

PF2: The pin function is switched as shown below according to the bit PF2DDR.

PF2DDR	0	1
Pin function	PF2 input	PF2 output

PF1: The pin function is switched as shown below according to the bit PF1DDR.

PF1DDR	0	1
Pin function	PF1 input	PF1 output

PF0/$\overline{\text{WAIT}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit WAITE of BCR, and bit PF0DDR.

Operating mode	1, 2, 4, 5, 6			7				
EXPE	—			0		1		
WAITE	0		1	—		0		1
PF0DDR	0	1	—	0	1	0	1	—
Pin function	PF0 input	PF0 output	$\overline{\text{WAIT}}$ input	PF0 input	PF0 output	PF0 input	PF0 output	$\overline{\text{WAIT}}$ input

8.15 Port G

Port G is a 7-bit I/O port that also has other functions. The port G has the following registers.

- Port G data direction register (PGDDR)
- Port G data register (PGDR)
- Port G register (PORTG)
- Port Function Control Register 0 (PFCR0)

8.15.1 Port G Data Direction Register (PGDDR)

The individual bits of PGDDR specify input or output for the pins of port G.

PGDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved If read, it returns an undefined value.
6	PG6DDR	0	W	• Modes 1, 2, 4, 5, and 6 Pins PG6 to PG4 function as bus control input/output pins ($\overline{\text{BREQO}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQ}}$) when the appropriate bus controller settings are made. Otherwise, these pins are I/O ports, and their functions can be switched with PGDDR. When the $\overline{\text{CS}}$ output enable bits (CS3E to CS0E) are set to 1, pins PG3 to PG0 function as $\overline{\text{CS}}$ output pins when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When CS3E to CS0E are cleared to 0, pins PG3 to PG0 are I/O ports, and their functions can be switched with PGDDR. • Mode 7 (when EXPE = 1) Pins PG6 to PG4 function as bus control input/output pins ($\overline{\text{BREQO}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQ}}$) when the appropriate bus controller settings are made. Otherwise, these pins are output ports when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When the $\overline{\text{CS}}$ output enable bits (CS3E to CS0E) are set to 1, pins PG3 to PG0 function as $\overline{\text{CS}}$ output pins when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When CS3E to CS0E are cleared to 0, pins PG3 to PG0 are I/O ports, and their functions can be switched with PGDDR. • Mode 7 (when EXPE = 0) Pins PG6 to PG0 are I/O ports, and their functions can be switched with PGDDR.
5	PG5DDR	0	W	
4	PG4DDR	0	W	
3	PG3DDR	0	W	
2	PG2DDR	0	W	
1	PG1DDR	0	W	
0	PG0DDR	1/0*	W	

Note: * PG0DDR is initialized to 1 in modes 1, 2, 5, and 6, and to 0 in modes 4 and 7.

8.15.2 Port G Data Register (PGDR)

PGDR stores output data for the port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0, and cannot be modified.
6	PG6DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
5	PG5DR	0	R/W	
4	PG4DR	0	R/W	
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

8.15.3 Port G Register (PORTG)

PORTG shows port G pin states.

PORTG cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved If this bit is read, it will return an undefined value.
6	PG6	Undefined*	R	If a port G read is performed while PGDDR bits are set to 1, the PGDR values are read. If a port G read is performed while PGDDR bits are cleared to 0, the pin states are read.
5	PG5	Undefined*	R	
4	PG4	Undefined*	R	
3	PG3	Undefined*	R	
2	PG2	Undefined*	R	
1	PG1	Undefined*	R	
0	PG0	Undefined*	R	

Note: * Determined by the states of pins PG6 to PG0.

8.15.4 Port Function Control Register 0 (PFCR0)

PFCR0 performs I/O port control.

Bit	Bit Name	Initial Value	R/W	Description
7	CS7E	1	R/W	CS7 to CS0 Enable
6	CS6E	1	R/W	These bits enable or disable the corresponding $\overline{\text{CSn}}$ output.
5	CS5E	1	R/W	
4	CS4E	1	R/W	0: Pin is designated as I/O port
3	CS3E	1	R/W	1: Pin is designated as $\overline{\text{CSn}}$ output pin
2	CS2E	1	R/W	(n = 7 to 0)
1	CS1E	1	R/W	
0	CS0E	1	R/W	

8.15.5 Pin Functions

Port G pins also function as bus control signal I/Os. The correspondence between the register specification and the pin functions is shown below.

PG6/$\overline{\text{BREQ}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, and bit PG6DDR.

Operating mode	1, 2, 4, 5, 6			7				
EXPE	—			0		1		
BRLE	0		1	—		0		1
PG6DDR	0	1	—	0	1	0	1	—
Pin function	PG6 input	PG6 output	$\overline{\text{BREQ}}$ input	PG6 input	PG6 output	PG6 input	PG6 output	$\overline{\text{BREQ}}$ input

PG5/$\overline{\text{BACK}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, and bit PG5DDR.

Operating mode	1, 2, 4, 5, 6			7				
EXPE	—			0		1		
BRLE	0		1	—		0		1
PG5DDR	0	1	—	0	1	0	1	—
Pin function	PG5 input	PG5 output	$\overline{\text{BACK}}$ output	PG5 input	PG5 output	PG5 input	PG5 output	$\overline{\text{BACK}}$ output

PG4/$\overline{\text{BREQO}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, bit BREQO, and bit PG4DDR.

Operating mode	1, 2, 4, 5, 6					7						
EXPE	—					0		1				
BRLE	0		1			—		0		1		
BREQO	—		0		1	—		—		0		1
PG4DDR	0	1	0	1	—	0	1	0	1	0	1	—
Pin function	PG4 input	PG4 output	PG4 input	PG4 output	$\overline{\text{BREQO}}$ output	PG4 input	PG4 output	PG4 input	PG4 output	PG4 input	PG4 output	$\overline{\text{BREQO}}$ output

PG3/$\overline{\text{CS3}}$/, PG2/$\overline{\text{CS2}}$, PG1/$\overline{\text{CS1}}$, PG0/$\overline{\text{CS0}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit CsnE, and bit PGnDDR.

Operating mode	1, 2, 4, 5, 6				7					
EXPE	—				0		1			
CSnE	0		1		—		0		1	
PGnDDR	0	1	0	1	0	1	0	1	0	1
Pin function	PGn input	PGn output	PGn input	$\overline{\text{CSn}}$ output	PGn input	PGn output	PGn input	PGn output	PGn input	$\overline{\text{CSn}}$ output

(n = 0 to 3)

8.16 Port H

Port H is a 4-bit I/O port that also has other functions. The port H has the following registers. For details on the port function control register 0, refer to section 8.15.4, Port Function Control Register 0 (PFCR0).

- Port H data direction register (PHDDR)
- Port H data register (PHDR)
- Port H register (PORTH)
- Port Function Control Register 0 (PFCR0)

8.16.1 Port H Data Direction Register (PHDDR)

The individual bits of PHDDR specify input or output for the pins of port H.

PHDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved If these bits are read, they will return an undefined value.
3	PH3DDR	0	W	• Modes 1, 2, 4, 5, 6, and 7 (when EXPE = 1) When the $\overline{OE}$ output enable bit (OEE) and $\overline{OE}$ output select bit (OES) are set to 1, pin PH3 functions as the $\overline{OE}$ output pin. Otherwise, when bit CS7E is set to 1, pin PH3 functions as a $\overline{CS}$ output pin when the corresponding PHDDR bit is set to 1, and as an input port when the bit is cleared to 0. When bit CS7E is cleared to 0, pin PH3 is an I/O port, and its function can be switched with PHDDR. When the $\overline{CS}$ output enable bits (CS7E to CS4E) are set to 1, pins PH2 to PH0 function as $\overline{CS}$ output pins when the corresponding PHDDR bit is set to 1, and as I/O ports when the bit is cleared to 0. When CS6E to CS4E are cleared to 0, pins PH2 to PH0 are I/O ports, and their functions can be switched with PHDDR. • Mode 7 (when EXPE = 0) Pins PH3 to PH0 are I/O ports, and their functions can be switched with PHDDR.
2	PH2DDR	0	W	
1	PH1DDR	0	W	
0	PH0DDR	0	W	

8.16.2 Port H Data Register (PHDR)

PHDR stores output data for the port H pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are reserved; they are always read as 0 and cannot be modified.
3	PH3DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

8.16.3 Port H Register (PORTH)

PORTH shows port H pin states.

PORTH cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, they will return an undefined value.
3	PH3	Undefined*	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.
2	PH2	Undefined*	R	
1	PH1	Undefined*	R	
0	PH0	Undefined*	R	

Note: * Determined by the states of pins PH3 to PH0.

8.16.4 Pin Functions

Port H pins also function as bus control signal I/Os and external interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

PH3/$\overline{\text{CS7}}$/$\overline{\text{OE}}$/($\overline{\text{IRQ7}}$): The pin function is switched as shown below according to the operating mode, bit EXPE, bit CS7E, and bit PH3DDR.

Operating mode	1, 2, 4, 5, 6				7					
EXPE	—				0		1			
CS7E	0		1		—		0		1	
PH3DDR	0	1	0	1	0	1	0	1	0	1
Pin function	PH3 input	PH3 output	PH3 input	$\overline{\text{CS7}}$ output	PH3 input	PH3 output	PH3 input	PH3 output	PH3 input	$\overline{\text{CS7}}$ output
	$\overline{\text{IRQ7}}$ input*									

Note: * $\overline{\text{IRQ7}}$ interrupt input pin when bit ITS7 is set to 1 in ITSR

PH2/$\overline{\text{CS6}}$/($\overline{\text{IRQ6}}$): The pin function is switched as shown below according to the operating mode, bit EXPE, bit CS6E, and bit PH2DDR.

Operating mode	1, 2, 4, 5, 6				7					
EXPE	—				0		1			
CS6E	0		1		—		0		1	
PH2DDR	0	1	0	1	0	1	0	1	0	1
Pin function	PH2 input	PH2 output	PH2 input	$\overline{\text{CS6}}$ output	PH2 input	PH2 output	PH2 input	PH2 output	PH2 input	$\overline{\text{CS6}}$ output
	$\overline{\text{IRQ6}}$ interrupt input*									

Note: * $\overline{\text{IRQ6}}$ interrupt input pin when bit ITS6 is set to 1 in ITSR.

PH1/$\overline{\text{CS5}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit CS5E, and bit PH1DDR.

Operating mode	1, 2, 4, 5, 6				7					
EXPE	—				0		1			
CS5E	0		1		—		0		1	
PH1DDR	0	1	0	1	0	1	0	1	0	1
Pin function	PH1 input	PH1 output	PH1 input	$\overline{\text{CS5}}$ output	PH1 input	PH1 output	PH1 input	PH1 output	PH1 input	$\overline{\text{CS5}}$ output

PH0/$\overline{\text{CS4}}$: The pin function is switched as shown below according to the operating mode, bit EXPE, bit CS4E, and bit PH0DDR.

Operating mode	1, 2, 4, 5, 6				3, 7					
EXPE	—				0		1			
CS4E	0		1		—		0		1	
PH0DDR	0	1	0	1	0	1	0	1	0	1
Pin function	PH0 input	PH0 output	PH0 input	$\overline{\text{CS4}}$ output	PH0 input	PH0 output	PH0 input	PH0 output	PH0 input	$\overline{\text{CS4}}$ output

Section 9 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels. The function list of the 16-bit timer unit and its block diagram are shown in table 9.1 and figure 9.1, respectively.

9.1 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 operations:
    Multiple timer counters (TCNT) can be written to simultaneously
    Simultaneous clearing by compare match and input capture possible
    Register simultaneous input/output possible by counter synchronous operation
  Maximum of 15-phase PWM output possible by 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
- Programmable pulse generator (PPG) output trigger can be generated
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

Table 9.1 TPU Functions

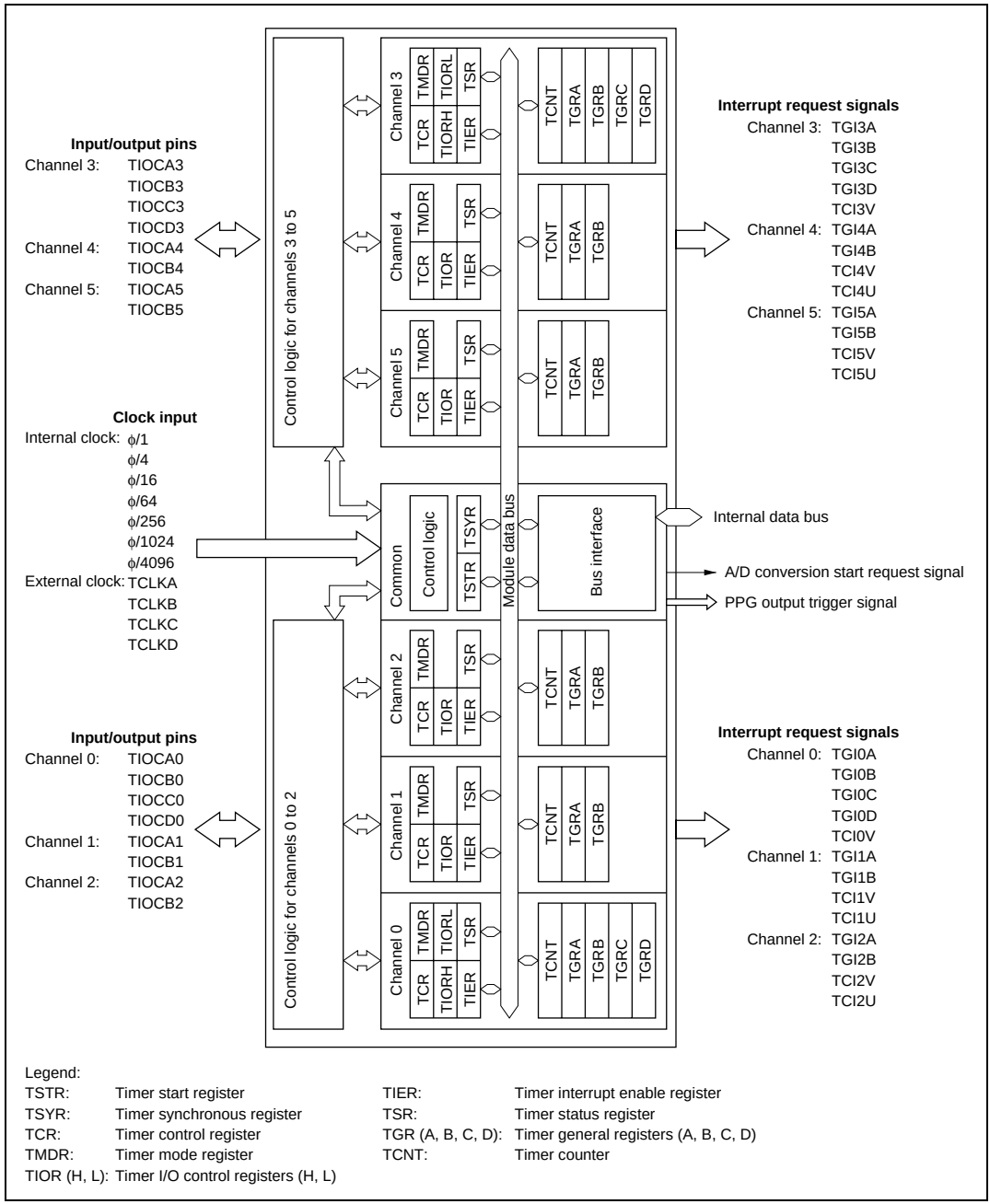
Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		φ/1 φ/4 φ/16 φ/64 TCLKA TCLKB TCLKC TCLKD	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKB	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKB TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 φ/1024 φ/4096 TCLKA	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 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 compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
PPG trigger	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	TGRA/ TGRB 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



Legend:

TSTR:	Timer start register	TIER:	Timer interrupt enable register
TSYR:	Timer synchronous register	TSR:	Timer status register
TCR:	Timer control register	TGR (A, B, C, D):	Timer general registers (A, B, C, D)
TMDR:	Timer mode register	TCNT:	Timer counter
TIOR (H, L):	Timer I/O control registers (H, L)		

Figure 9.1 Block Diagram of TPU

9.2 Input/Output Pins

Table 9.2 Pin Configuration

Channel	Symbol	I/O	Function
All	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

9.3 Register Descriptions

The TPU has the following registers in each channel.

- 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)
- 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)
- 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)
- 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)
- 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)
- 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 synchronous register (TSYR)

9.3.1 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. TCR register settings should be made 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 9.3 and 9.4 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. ϕ/4 both edges = ϕ/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 ϕ/4 or slower. This setting is ignored if the input clock is ϕ/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 9.5 to 9.10 for details.
0	TPSC0	0	R/W	

Table 9.3 CCLR2 to CCLR0 (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 setting is performed 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 9.4 CCLR2 to CCLR0 (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 setting is performed 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 9.5 TPSC2 to TPSC0 (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 9.6 TPSC2 to TPSC0 (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 9.7 TPSC2 to TPSC0 (Channel 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 9.8 TPSC2 to TPSC0 (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 9.9 TPSC2 to TPSC0 (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 9.10 TPSC2 to TPSC0 (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.

9.3.2 Timer Mode Register (TMDR)

TMDR registers are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. TMDR register settings should be made 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 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 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 9.11 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 9.11 MD3 to MD0

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.

9.3.3 Timer I/O Control Register (TIOR)

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 since 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. For details, see tables 9.12, 9.14, 9.15, 9.16, 9.18, and 9.19.
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. For details, see tables 9.20, 9.22, 9.23, 9.24, 9.26, and 9.27.
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. For details, see tables 9.13, and 9.17.
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. For details, see tables 9.21, and 9.25
1	IOC1	0	R/W	
0	IOC0	0	R/W	

Table 9.12 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 bits TPSC2 to TPSC0 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 9.13 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 bits TPSC2 to TPSC0 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 9.14 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.15 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.16 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 bits TPSC2 to TPSC0 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 9.17 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 bits TPSC2 to TPSC0 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 9.18 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.19 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.20 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.21 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.22 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.23 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.24 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.25 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.26 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output 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 9.27 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 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	x	0	0	Input capture register	Input capture source is TIOCA5 pin Input capture at rising edge
			1		Input capture source is TIOCA5 pin Input capture at falling edge
		1	x		Input capture source is TIOCA5 pin Input capture at both edges

Legend: x: Don’t care

9.3.4 Timer Interrupt Enable Register (TIER)

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

9.3.5 Timer Status Register (TSR)

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. 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. [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. 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 while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register [Clearing conditions] • When DTC is activated by TGID interrupt while 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. 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 while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIC interrupt while 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. [Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIB interrupt while 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. [Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1

Note: * Only 0 can be written, for flag clearing.

9.3.6 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, or in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

9.3.7 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers with a dual function as output compare and 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.

9.3.8 Timer Start Register (TSTR)

TSTR selects operation/stoppage 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 These bits should always be written with 0.
5	CST5	0	R/W	Counter Start 5 to 0 These bits select operation or stoppage for 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_5 to TCNT_0 count operation is stopped 1: TCNT_5 to TCNT_0 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	

9.3.9 Timer Synchronous Register (TSYR)

TSYR selects independent operation or synchronous operation for the TCNT counters of 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 These bits should always be written with 0.
5	SYNC5	0	R/W	Timer Synchronization 5 to 0 These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through 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 bits CCLR2 to CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_5 to TCNT_0 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	

9.4 Operation

9.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Counter Operation: When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

1. Example of count operation setting procedure
   Figure 9.2 shows an example of the count operation setting procedure.

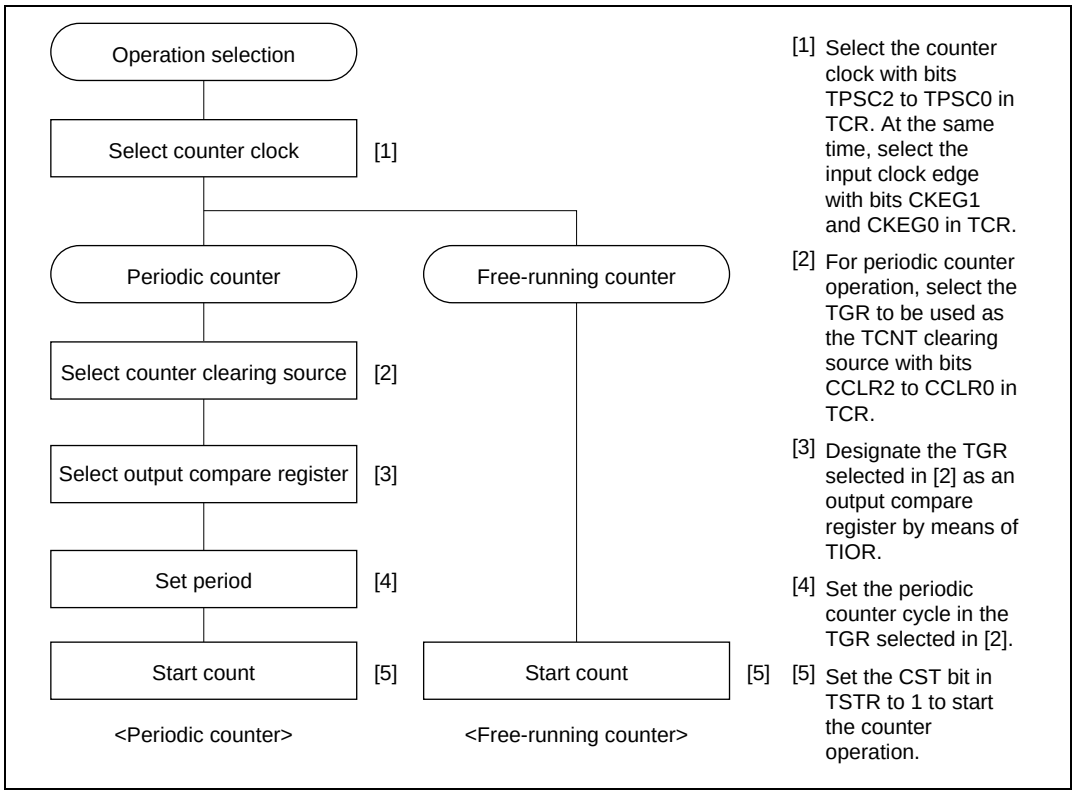


Figure 9.2 Example of Counter Operation Setting Procedure

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 (changes 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 9.3 illustrates free-running counter operation.

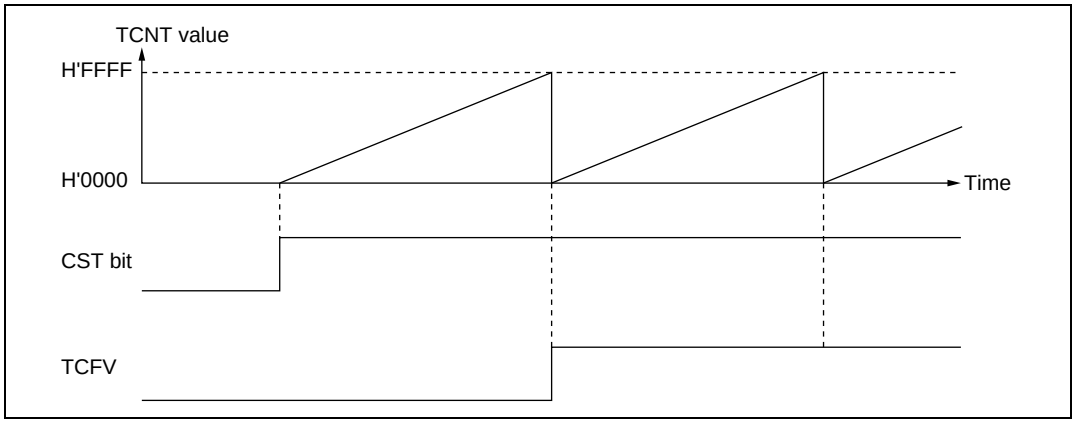


Figure 9.3 Free-Running Counter Operation

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 bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up 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 9.4 illustrates periodic counter operation.

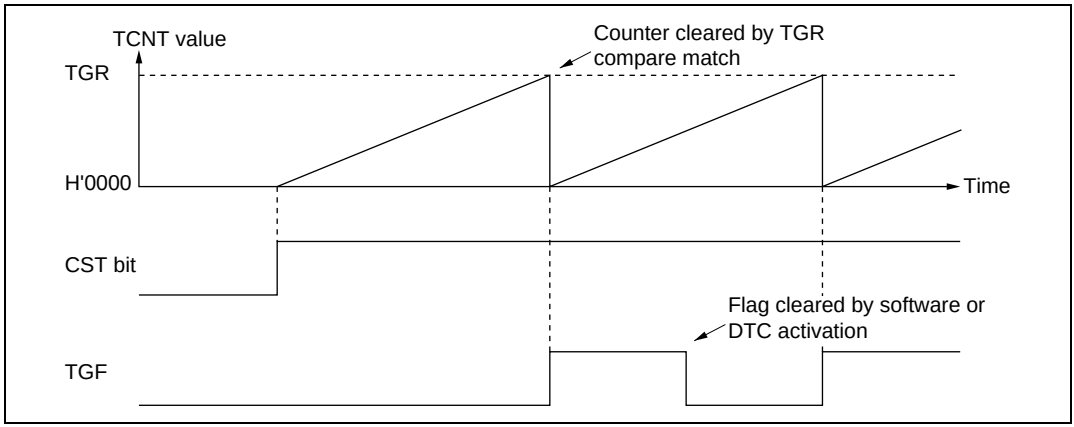


Figure 9.4 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using a compare match.

1. Example of setting procedure for waveform output by compare match

   Figure 9.5 shows an example of the setting procedure for waveform output by a compare match.

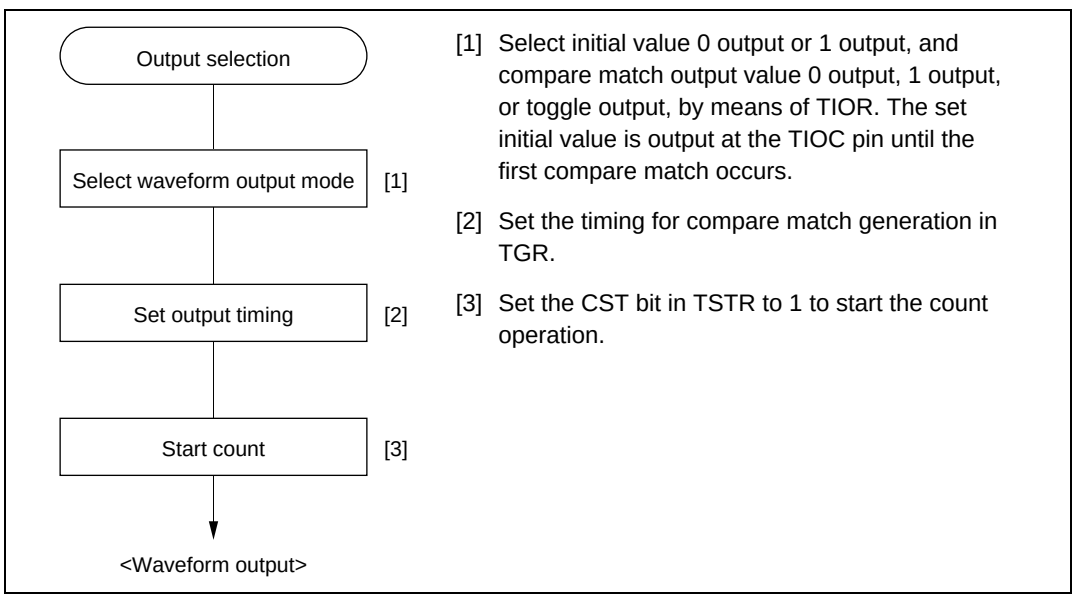


Figure 9.5 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of waveform output operation

Figure 9.6 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 so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level match, the pin level does not change.

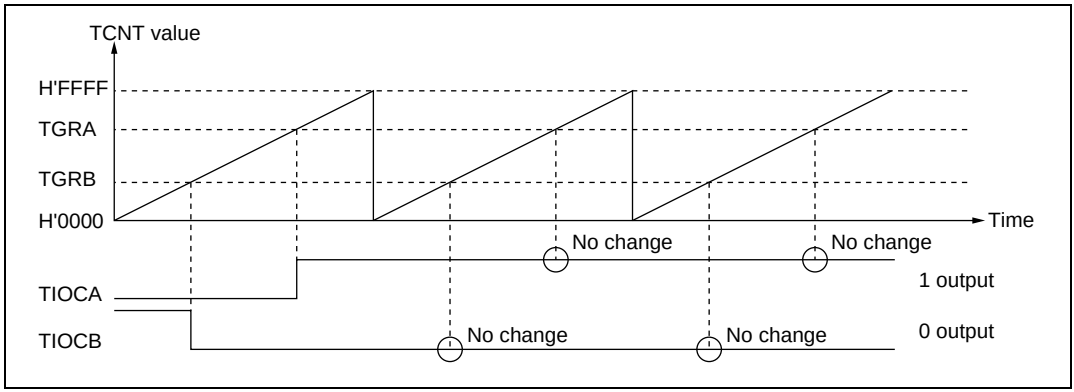


Figure 9.6 Example of 0 Output/1 Output Operation

Figure 9.7 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

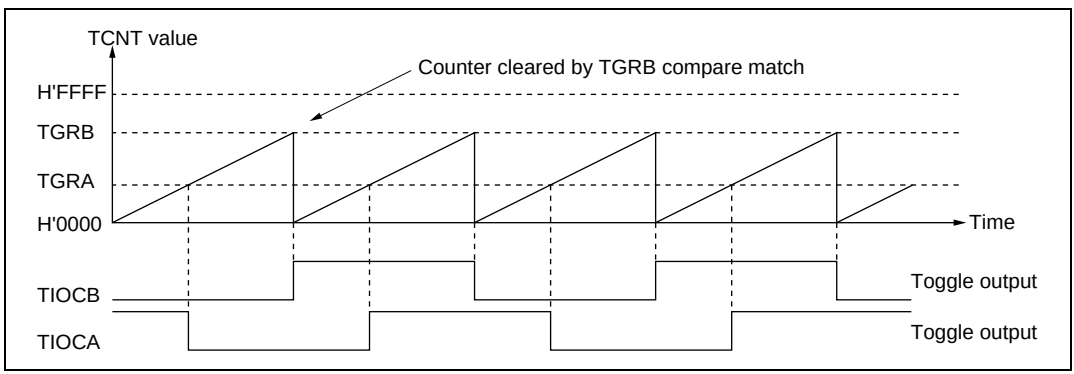


Figure 9.7 Example of Toggle Output Operation

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 detection 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, $\phi/1$ should not be selected as the counter input clock used for input capture input. Input capture will not be generated if $\phi/1$ is selected.

1. Example of setting procedure for input capture operation

   Figure 9.8 shows an example of the setting procedure for input capture operation.

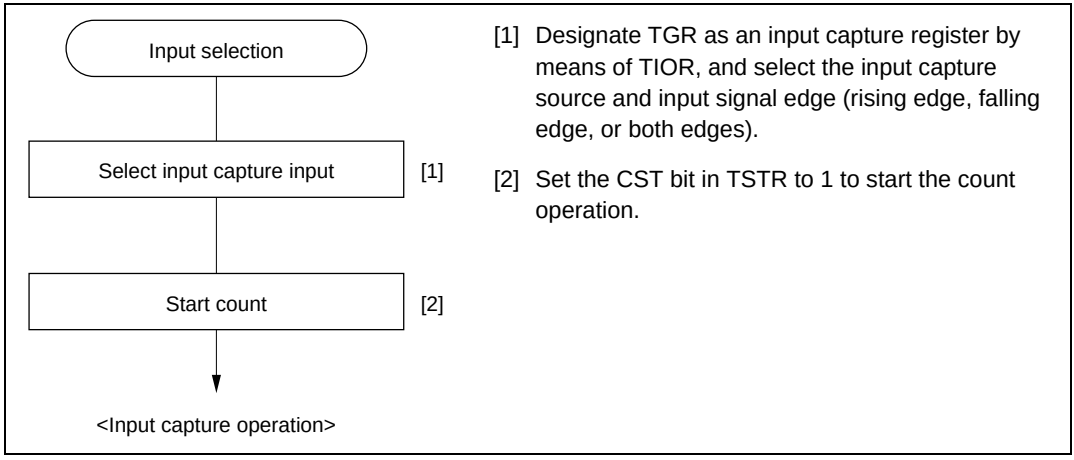


Figure 9.8 Example of Setting Procedure for Input Capture Operation

2. Example of input capture operation

   Figure 9.9 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, 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.

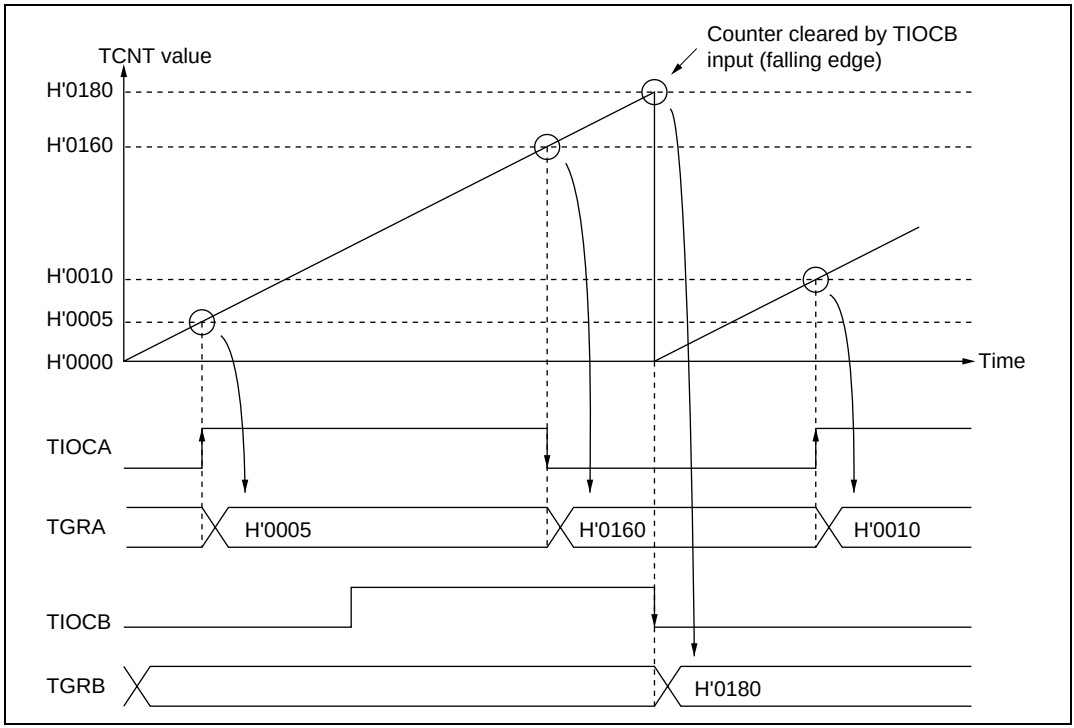


Figure 9.9 Example of Input Capture Operation

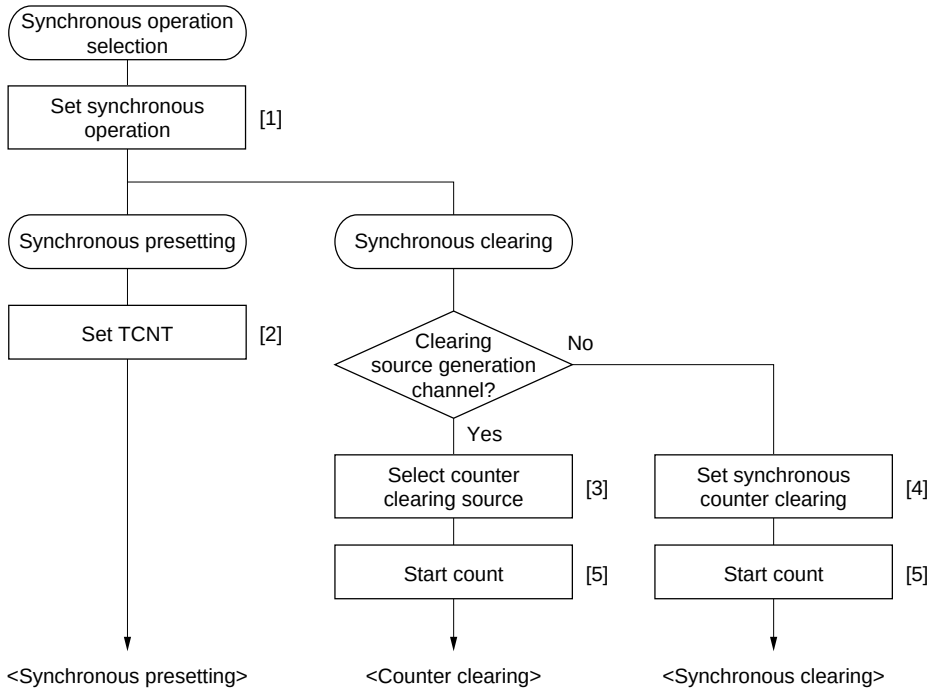
9.4.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be rewritten simultaneously (synchronous presetting). Also, multiple of TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

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 9.10 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 bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 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 9.10 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 9.11 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, is performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details on PWM modes, see section 9.4.5, PWM Modes.

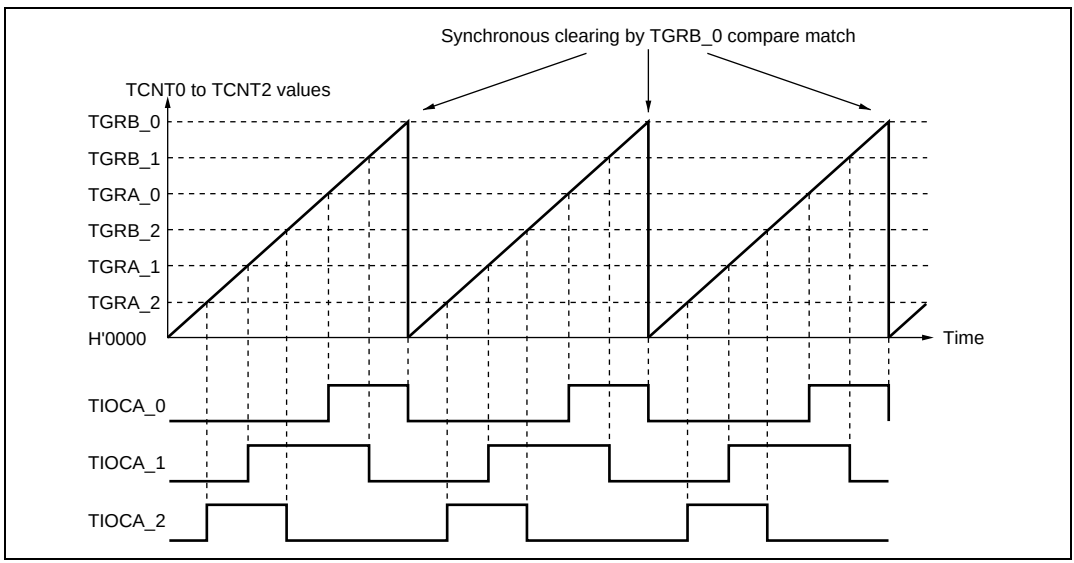


Figure 9.11 Example of Synchronous Operation

9.4.3 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 a compare match register.

Table 9.28 shows the register combinations used in buffer operation.

Table 9.28 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 9.12.

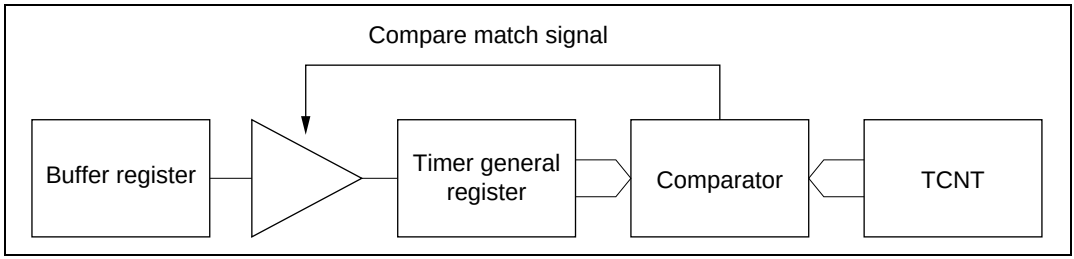


Figure 9.12 Compare Match Buffer Operation

- 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 9.13.

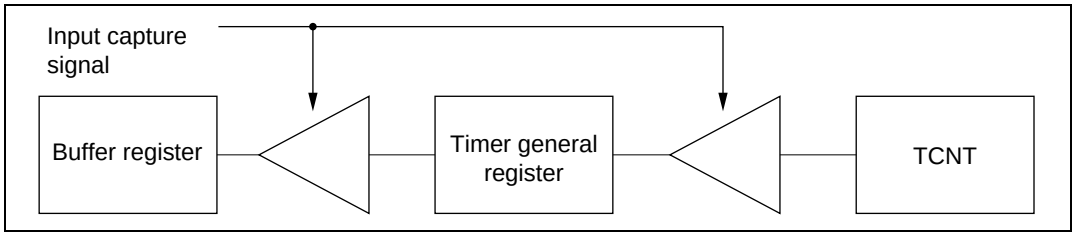


Figure 9.13 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 9.14 shows an example of the buffer operation setting procedure.

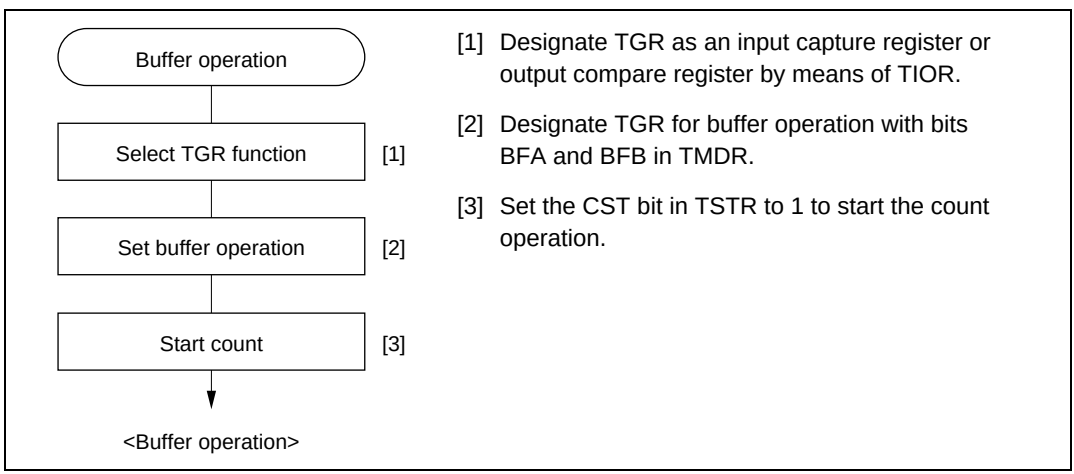


Figure 9.14 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation:

1. When TGR is an output compare register

   Figure 9.15 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 compare match A occurs.

   For details on PWM modes, see section 9.4.5, PWM Modes.

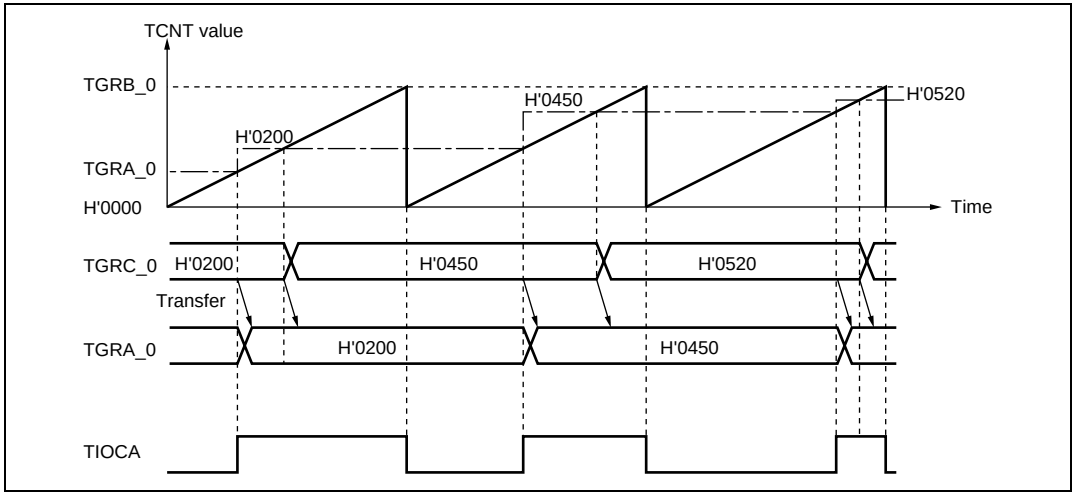


Figure 9.15 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 9.16 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 occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

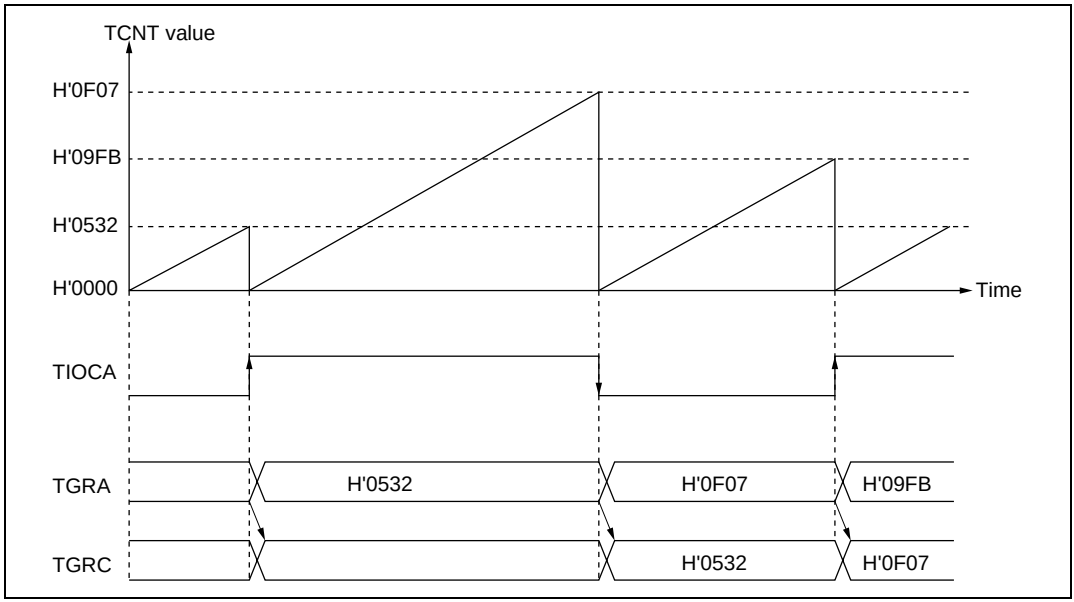


Figure 9.16 Example of Buffer Operation (2)

9.4.4 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 at overflow/underflow of TCNT_2 (TCNT_5) as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 9.29 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 counter operates independently in phase counting mode.

Table 9.29 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 9.17 shows an example of the setting procedure for cascaded operation.

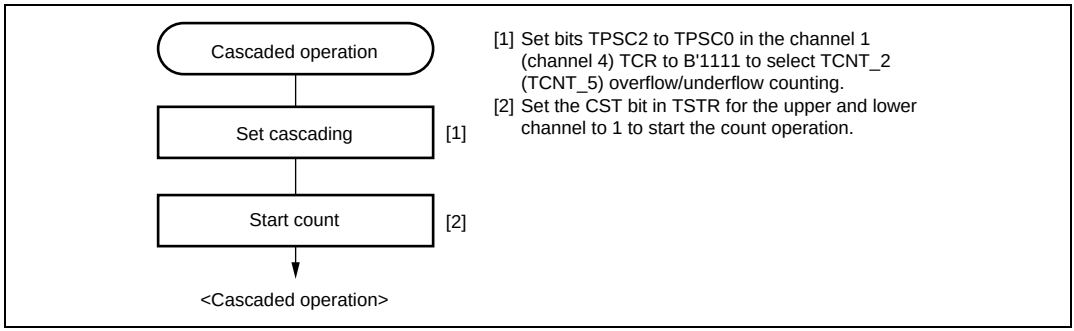


Figure 9.17 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 9.18 illustrates the operation when counting upon TCNT_2 overflow/underflow has been set for TCNT_1, TGRA_1 and TGRA_2 have been designated as input capture registers, and the 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.

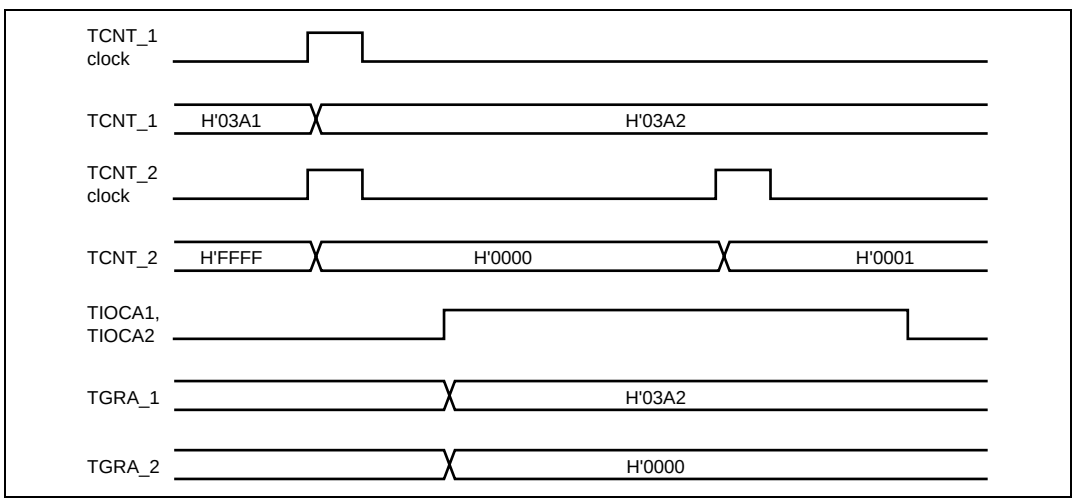


Figure 9.18 Example of Cascaded Operation (1)

Figure 9.19 illustrates the operation when counting upon TCNT_2 overflow/underflow 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.

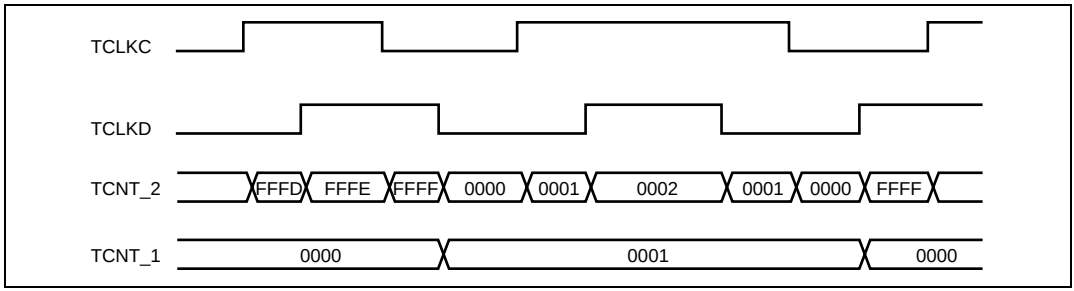


Figure 9.19 Example of Cascaded Operation (2)

9.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Settings of TGR registers can output a PWM waveform in the range of 0–% to 100–% duty cycle.

Designating TGR compare match as the counter clearing source enables the cycle 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 outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the TIOCA and TIOCC pins at compare matches A and C, respectively. The outputs specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D, respectively. 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 cycle 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 cycle 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 by combined use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 9.30.

Table 9.30 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	TGRA_4	TIOCA4	TIOCA4
	TGRB_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 cycle is set.

Example of PWM Mode Setting Procedure: Figure 9.20 shows an example of the PWM mode setting procedure.

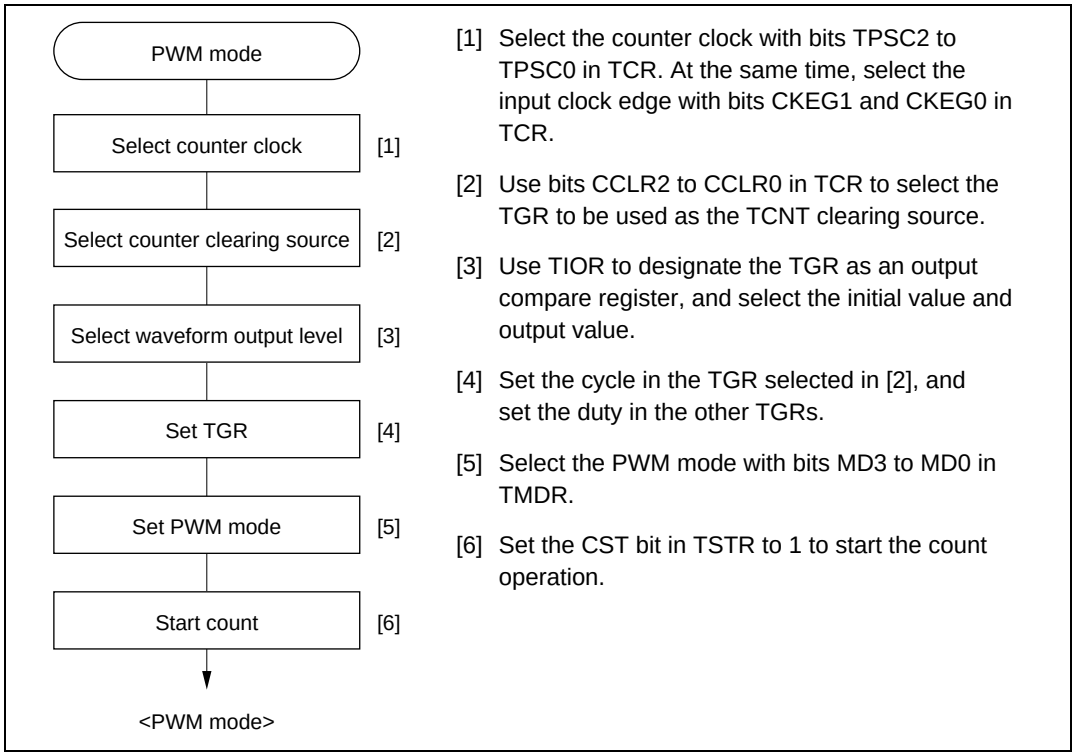


Figure 9.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 9.21 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 cycle, and the values set in TGRB registers as the duty cycle.

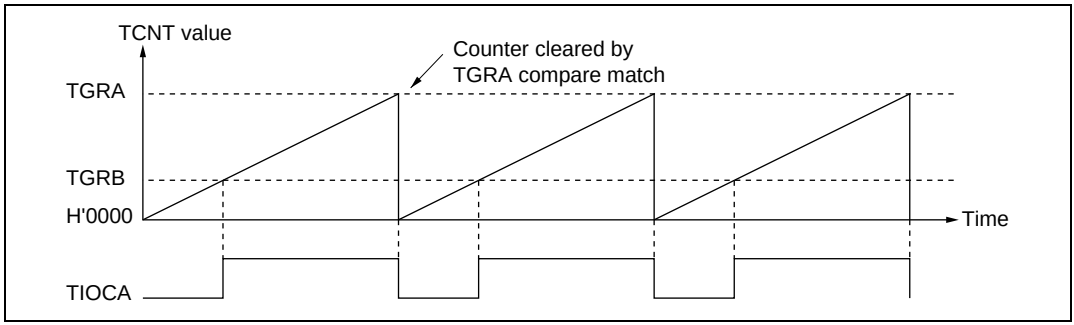


Figure 9.21 Example of PWM Mode Operation (1)

Figure 9.22 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), to output 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 as the duty cycle.

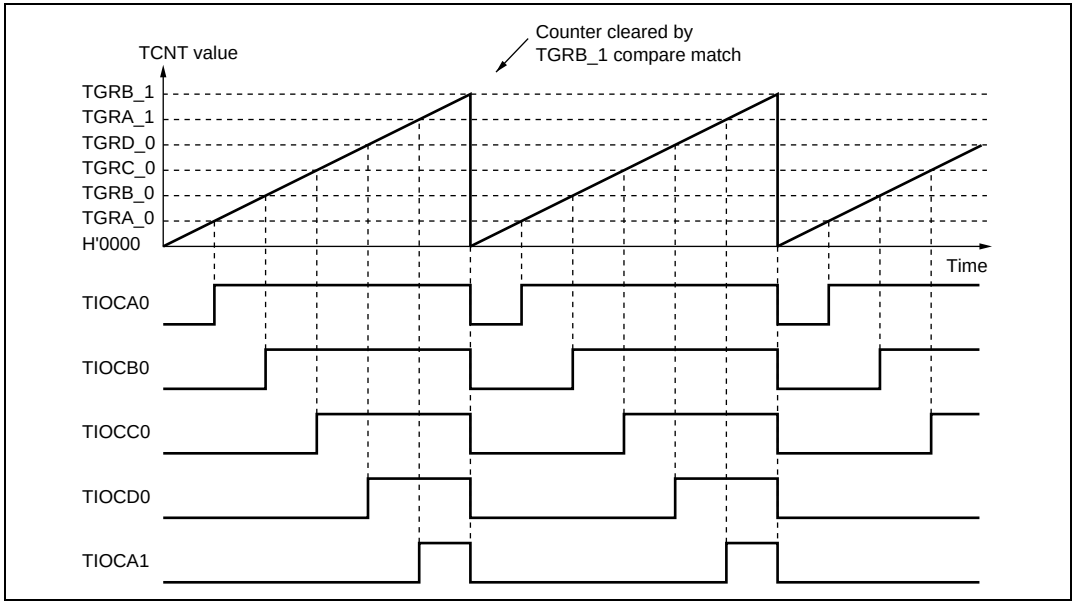


Figure 9.22 Example of PWM Mode Operation (2)

Figure 9.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

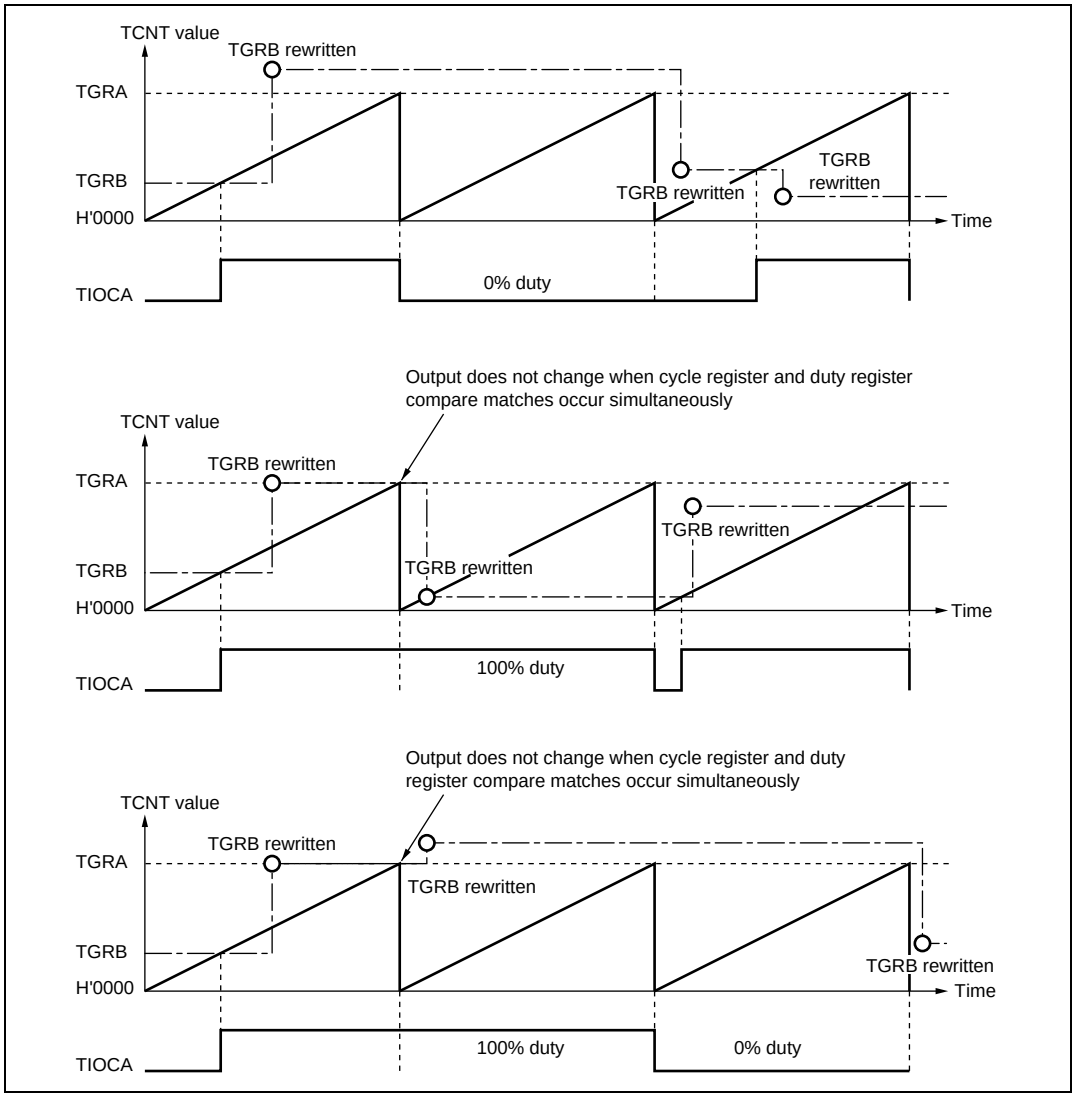


Figure 9.23 Example of PWM Mode Operation (3)

9.4.6 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 bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 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.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 9.31 shows the correspondence between external clock pins and channels.

Table 9.31 Clock Input Pins in Phase Counting Mode

	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 9.24 shows an example of the phase counting mode setting procedure.

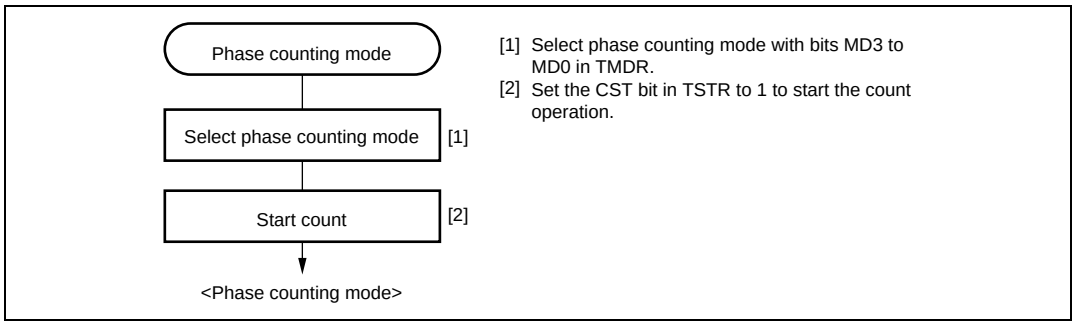


Figure 9.24 Example of Phase Counting Mode Setting Procedure

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 9.25 shows an example of phase counting mode 1 operation, and table 9.32 summarizes the TCNT up/down-count conditions.

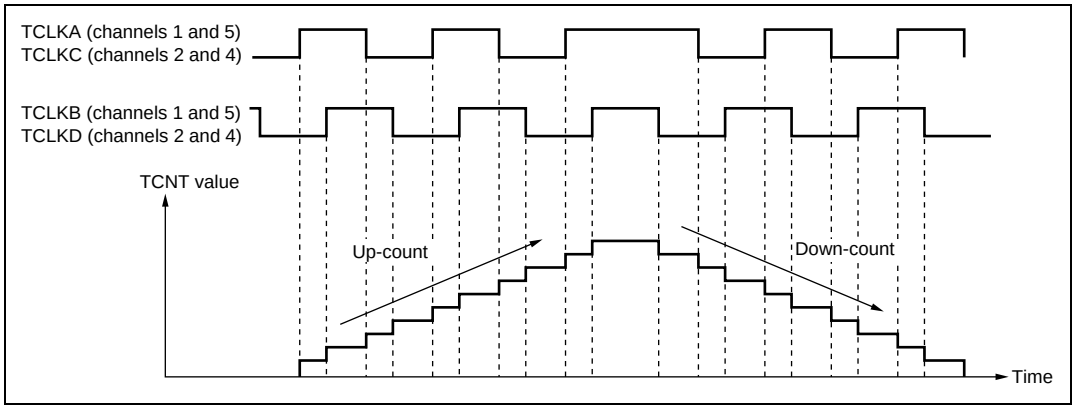


Figure 9.25 Example of Phase Counting Mode 1 Operation

Table 9.32 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 9.26 shows an example of phase counting mode 2 operation, and table 9.33 summarizes the TCNT up/down-count conditions.

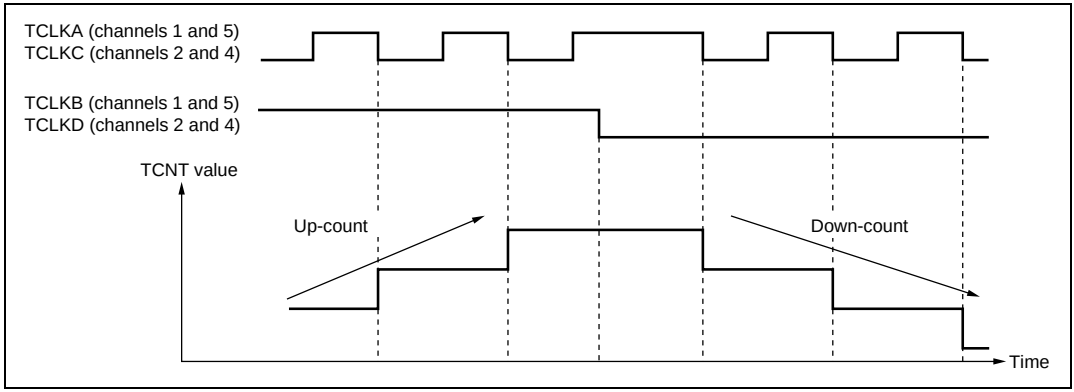


Figure 9.26 Example of Phase Counting Mode 2 Operation

Table 9.33 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	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	
↑	High level	
↓	Low level	Down-count

Legend:

↑ : Rising edge

↓ : Falling edge

3. Phase counting mode 3

   Figure 9.27 shows an example of phase counting mode 3 operation, and table 9.34 summarizes the TCNT up/down-count conditions.

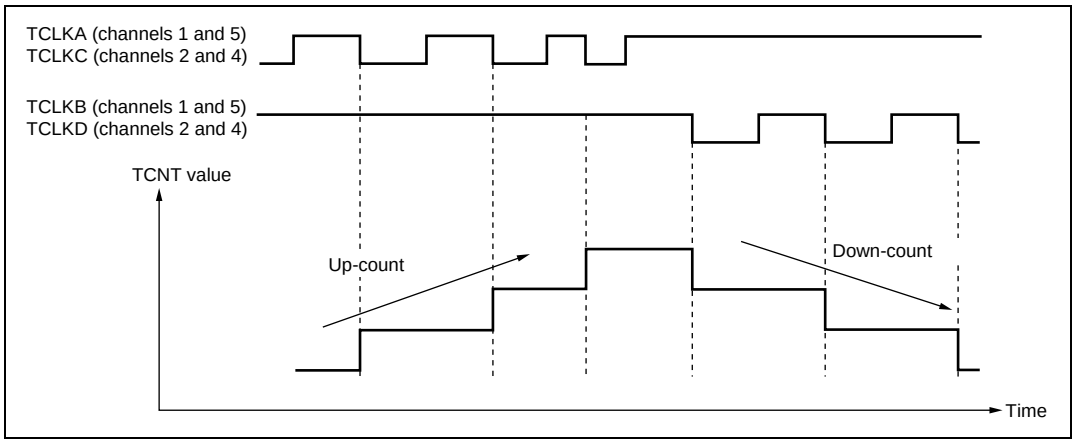


Figure 9.27 Example of Phase Counting Mode 3 Operation

Table 9.34 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	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	
↓	Low level	

Legend:

↑ : Rising edge

↓ : Falling edge

4. Phase counting mode 4

Figure 9.28 shows an example of phase counting mode 4 operation, and table 9.35 summarizes the TCNT up/down-count conditions.

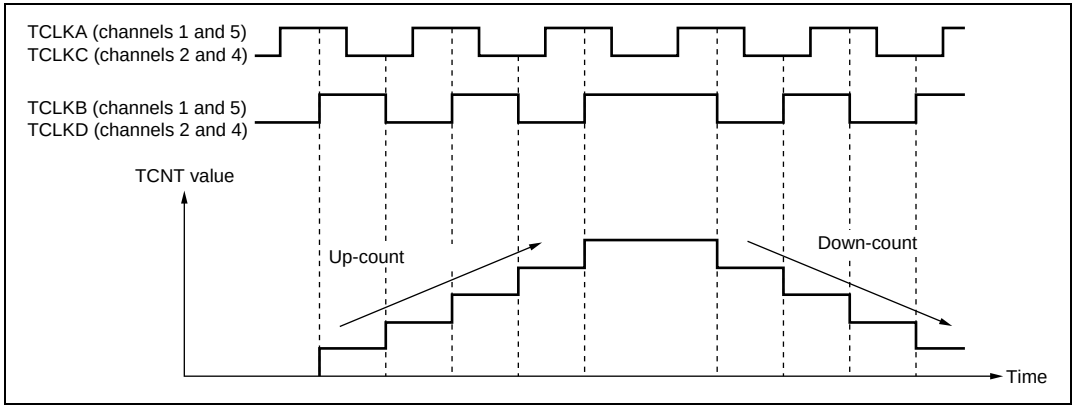


Figure 9.28 Example of Phase Counting Mode 4 Operation

Table 9.35 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 9.29 shows an example in which phase counting mode is designated for channel 1, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect the 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 cycle and position control cycle. 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 detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source, and the up/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

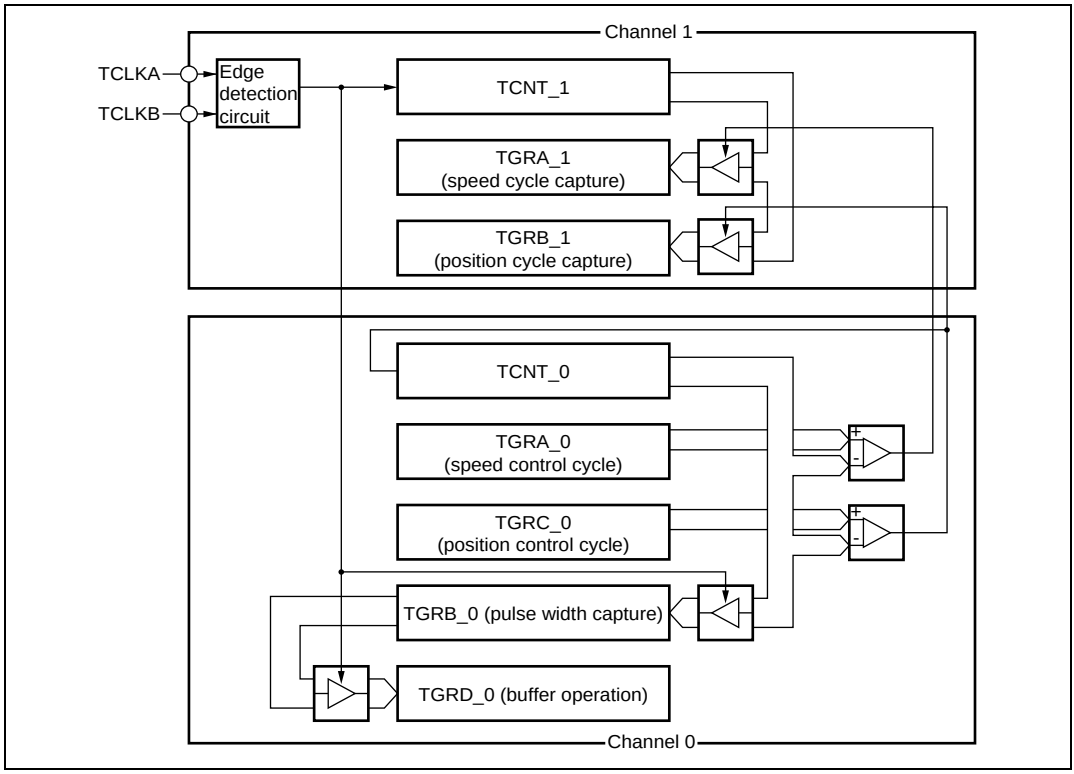


Figure 9.29 Phase Counting Mode Application Example

9.5 Interrupt Sources

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/disable bit, allowing 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, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 9.36 lists the TPU interrupt sources.

Table 9.36 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.

9.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 7, 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.

9.7 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 start 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 started.

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.

9.8 Operation Timing

9.8.1 Input/Output Timing

TCNT Count Timing: Figure 9.30 shows TCNT count timing in internal clock operation, and figure 9.31 shows TCNT count timing in external clock operation.

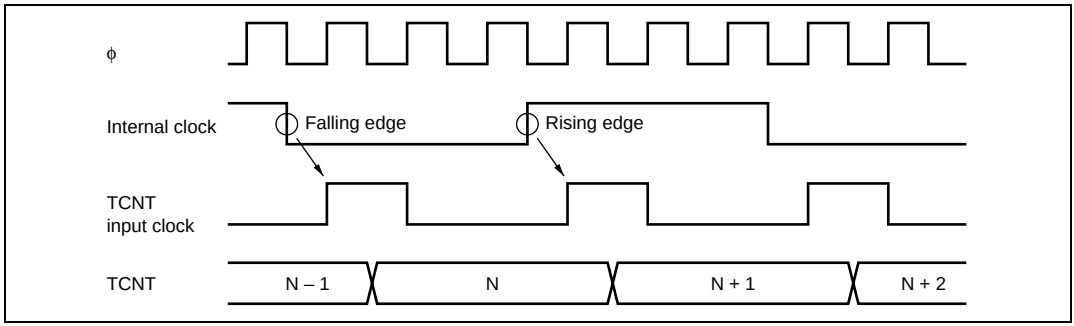


Figure 9.30 Count Timing in Internal Clock Operation

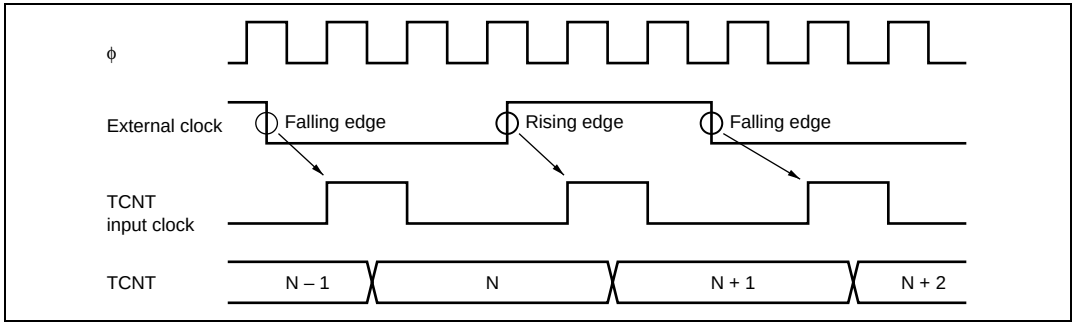


Figure 9.31 Count Timing in External Clock Operation

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 (TIOC pin) TCNT input clock is generated.

Figure 9.32 shows output compare output timing.

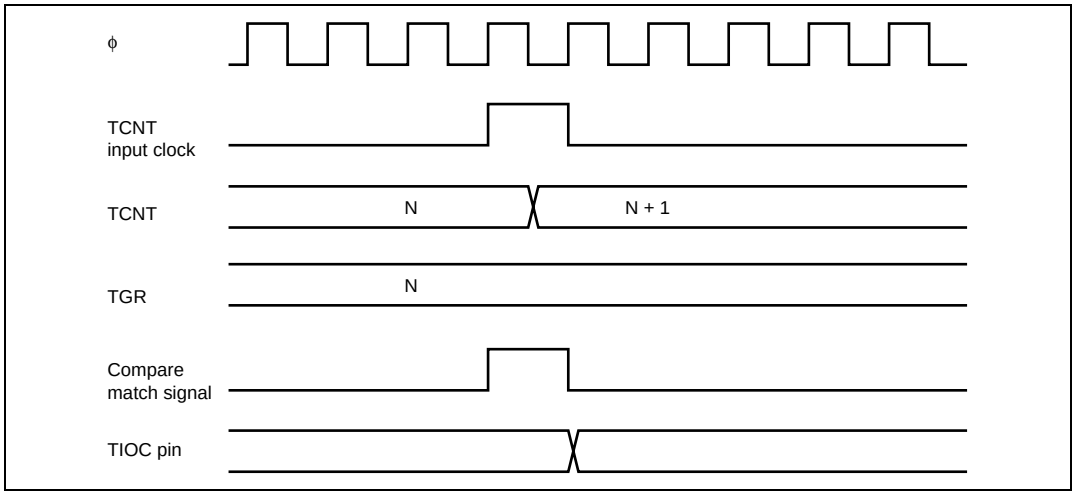


Figure 9.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 9.33 shows input capture signal timing.

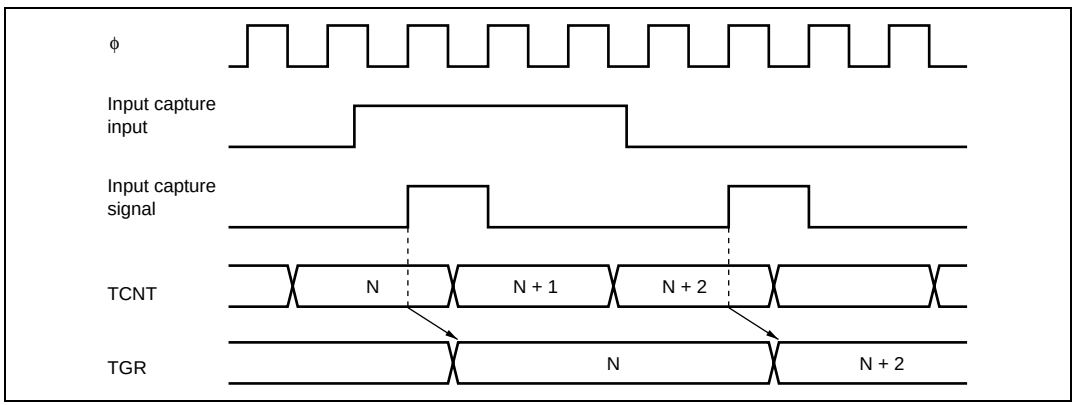


Figure 9.33 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 9.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 9.35 shows the timing when counter clearing by input capture occurrence is specified.

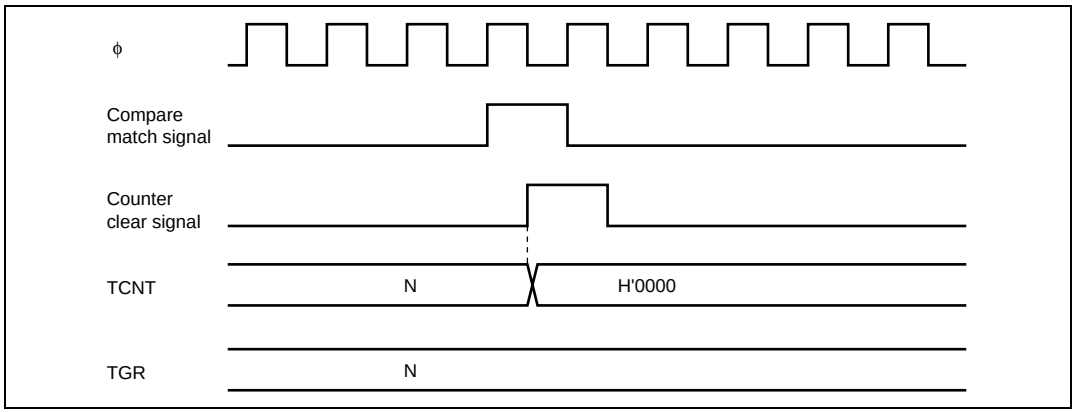


Figure 9.34 Counter Clear Timing (Compare Match)

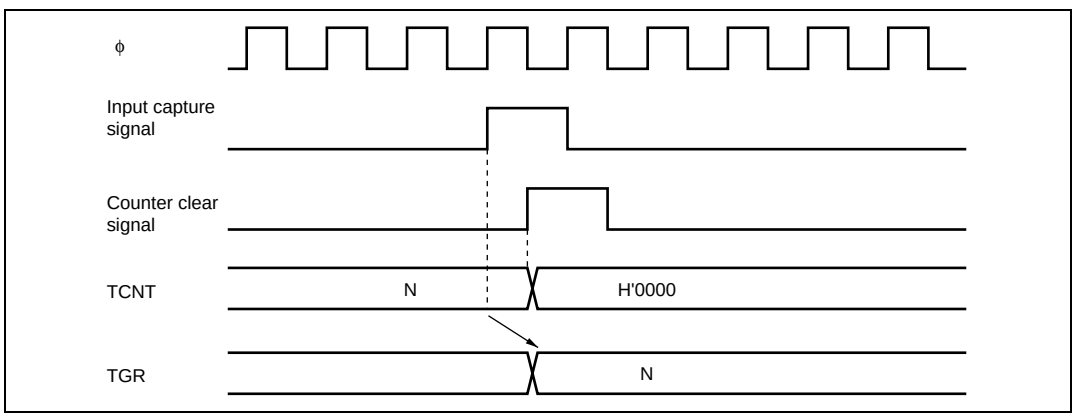


Figure 9.35 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 9.36 and 9.37 show the timings in buffer operation.

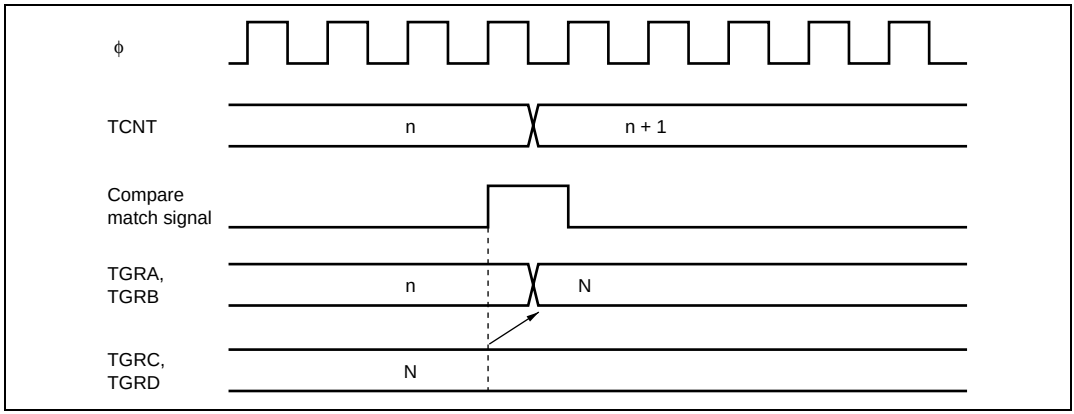


Figure 9.36 Buffer Operation Timing (Compare Match)

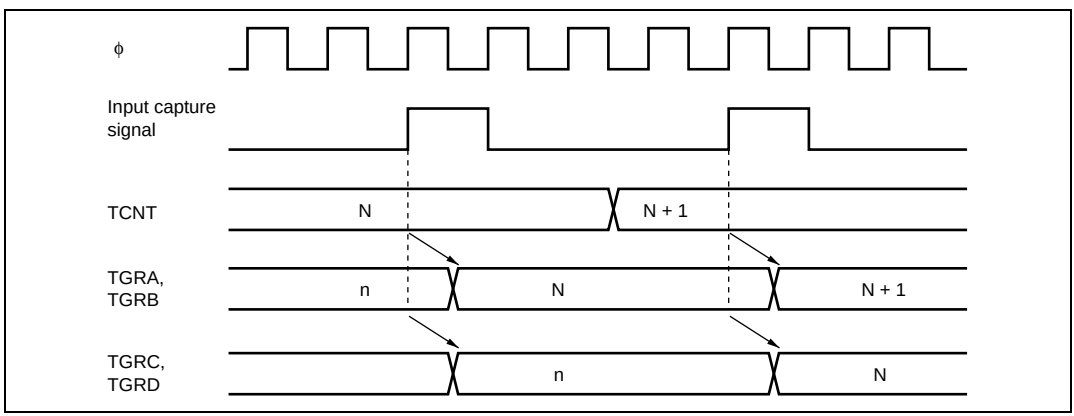


Figure 9.37 Buffer Operation Timing (Input Capture)

9.8.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 9.38 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and the TGI interrupt request signal timing.

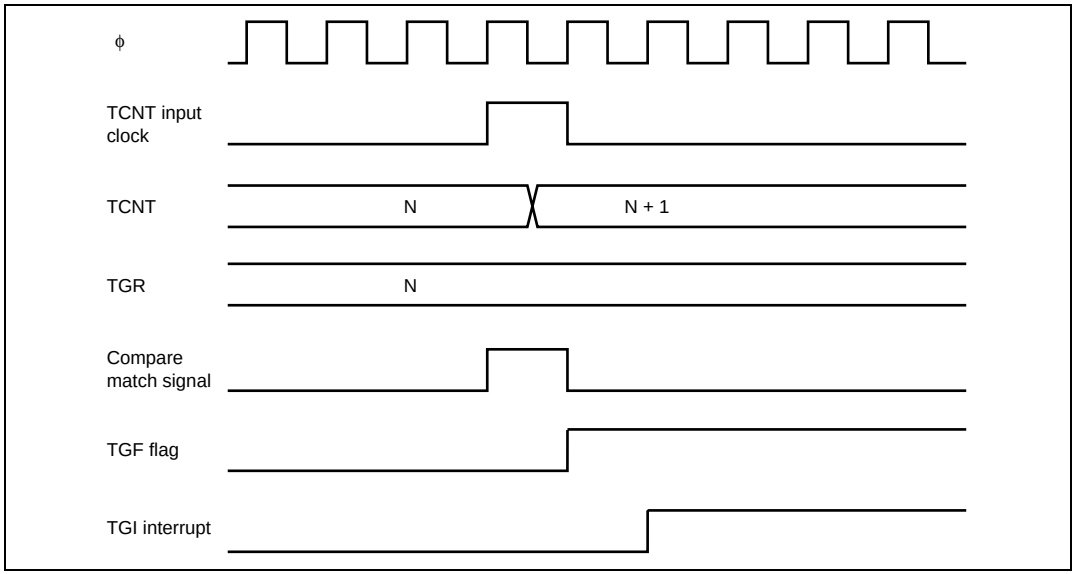


Figure 9.38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 9.39 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and the TGI interrupt request signal timing.

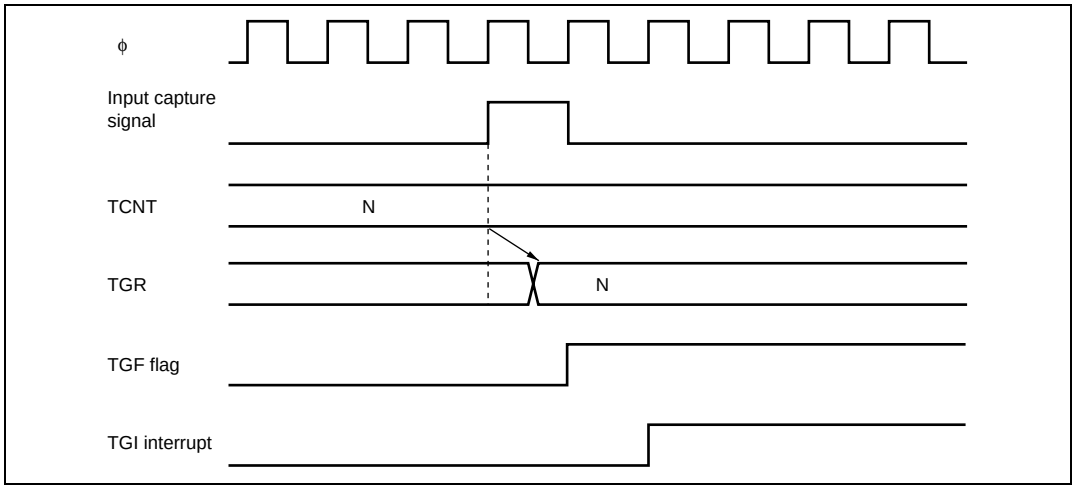


Figure 9.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 9.40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and the TCIV interrupt request signal timing.

Figure 9.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and the TCIU interrupt request signal timing.

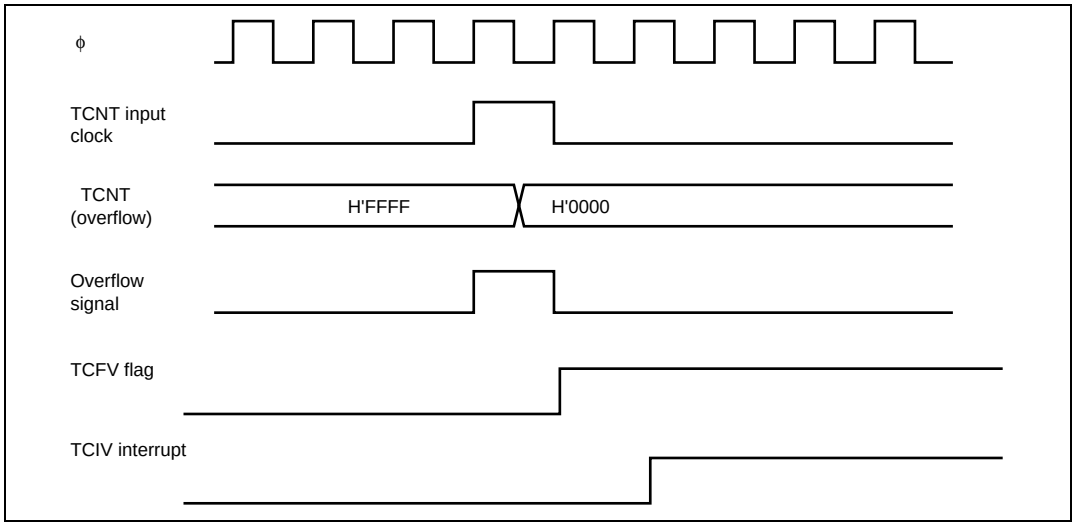


Figure 9.40 TCIV Interrupt Setting Timing

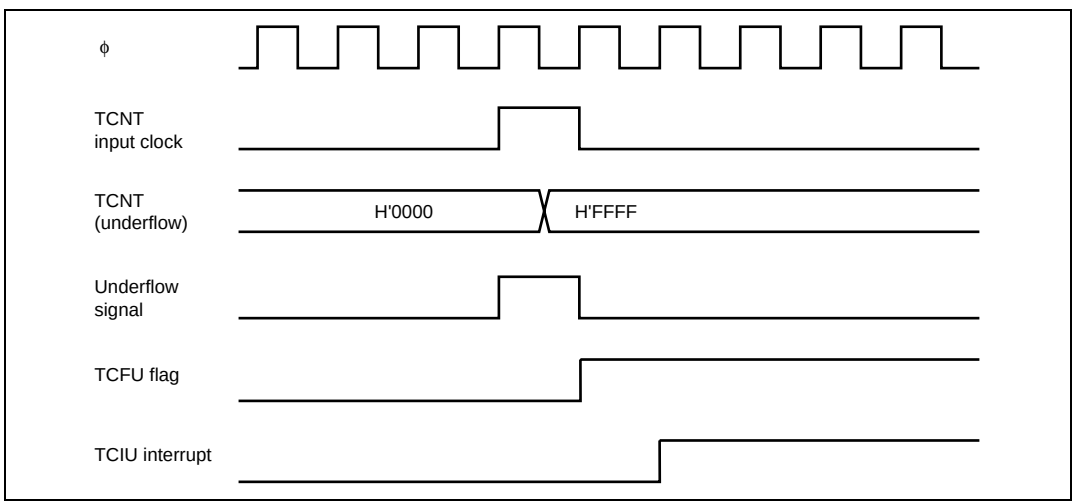


Figure 9.41 TCIU Interrupt Setting Timing

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 9.42 shows the timing for status flag clearing by the CPU, and figure 9.43 shows the timing for status flag clearing by the DTC.

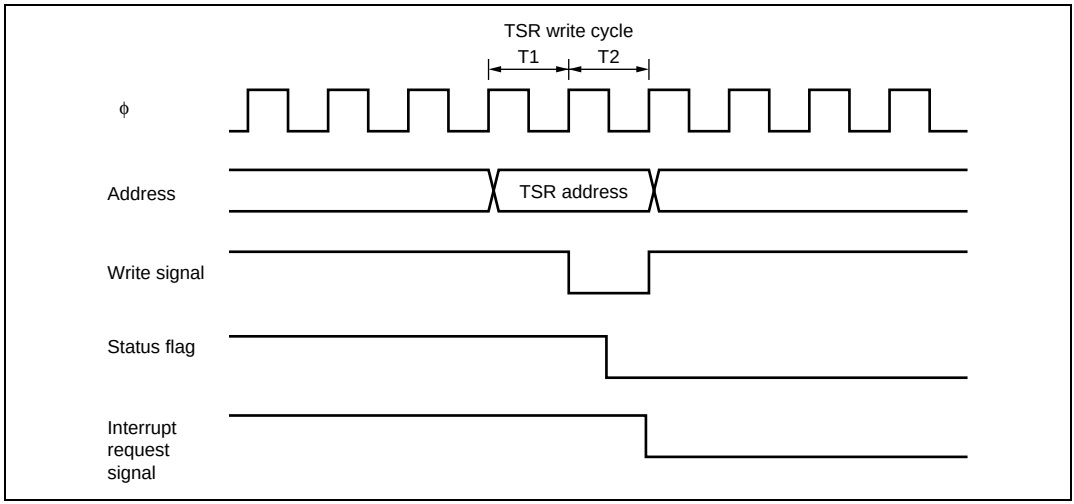


Figure 9.42 Timing for Status Flag Clearing by CPU

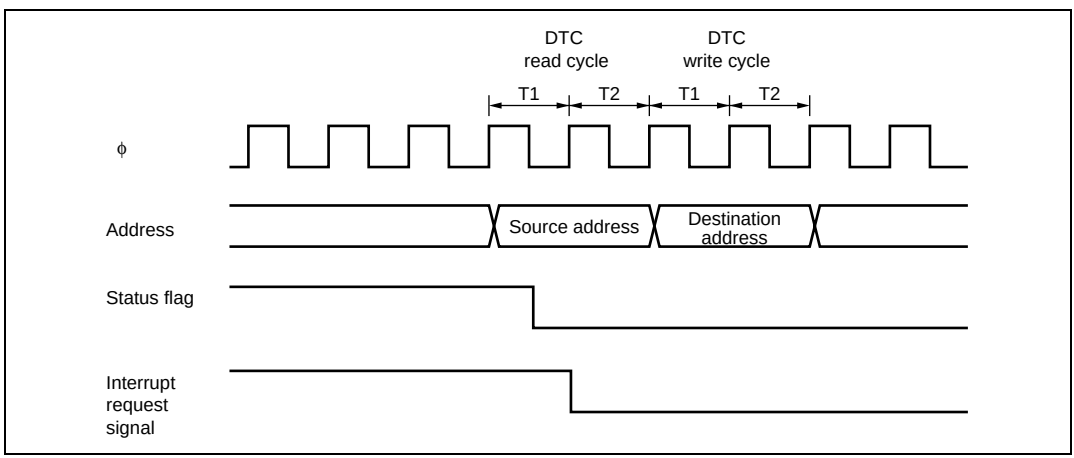


Figure 9.43 Timing for Status Flag Clearing by DTC Activation

9.9 Usage Notes

9.9.1 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 19, Power-Down Modes.

9.9.2 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 with a narrower pulse width.

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 9.44 shows the input clock conditions in phase counting mode.

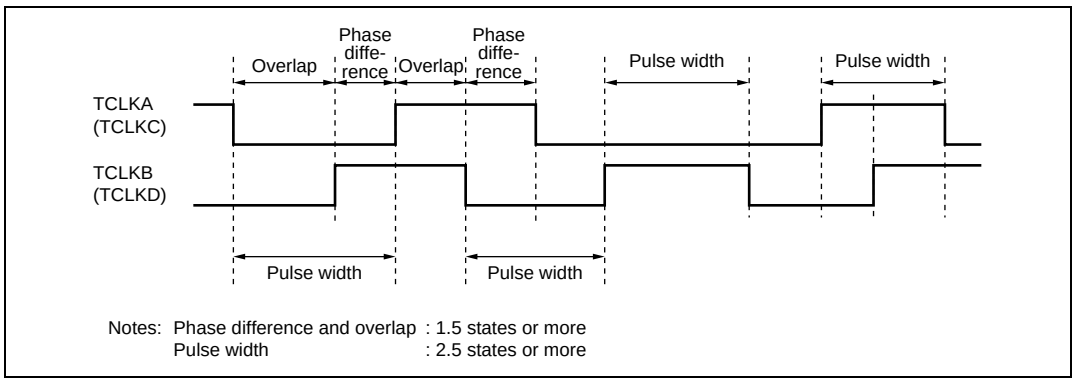


Figure 9.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

9.9.3 Caution on Cycle Setting

When counter clearing by 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

9.9.4 Contention between TCNT Write and Clear Operations

If the counter clearing signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 9.45 shows the timing in this case.

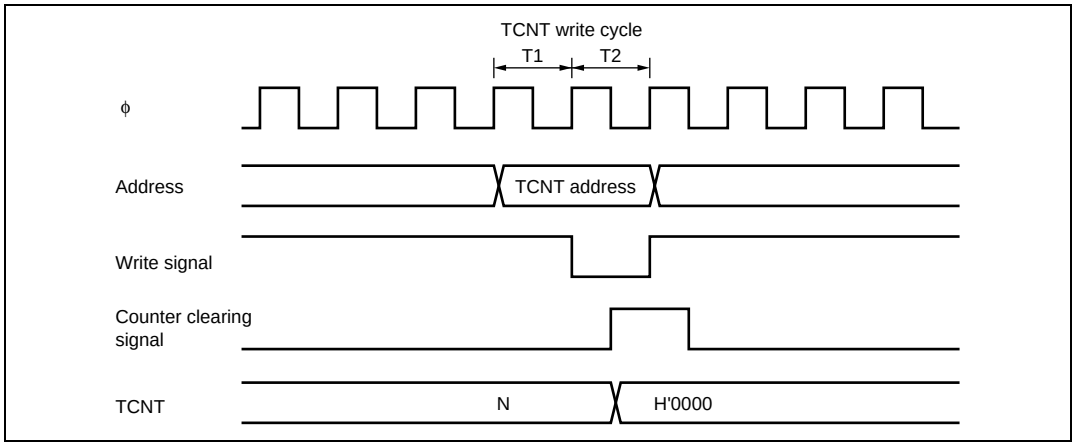


Figure 9.45 Contention between TCNT Write and Clear Operations

9.9.5 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 9.46 shows the timing in this case.

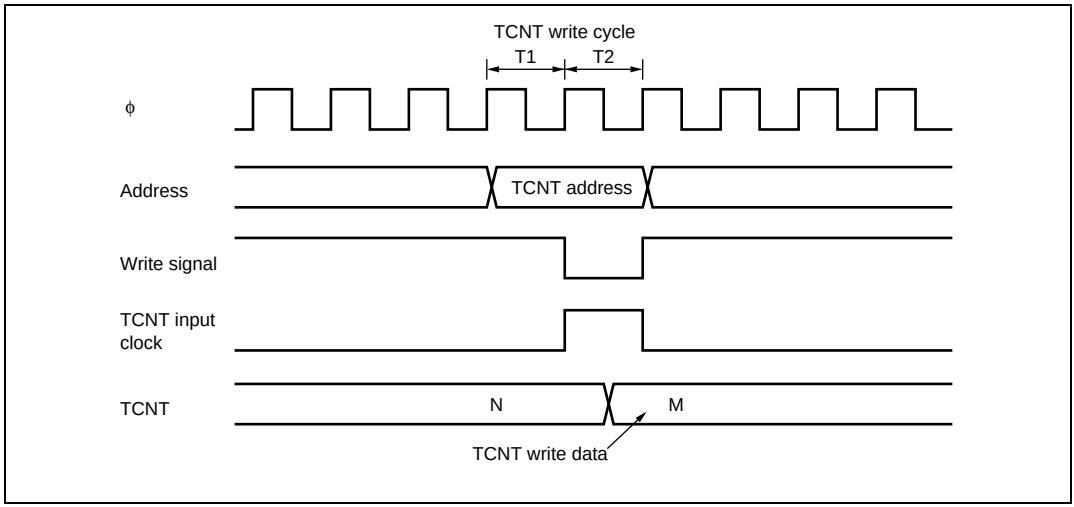


Figure 9.46 Contention between TCNT Write and Increment Operations

9.9.6 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 disabled. A compare match also does not occur when the same value as before is written.

Figure 9.47 shows the timing in this case.

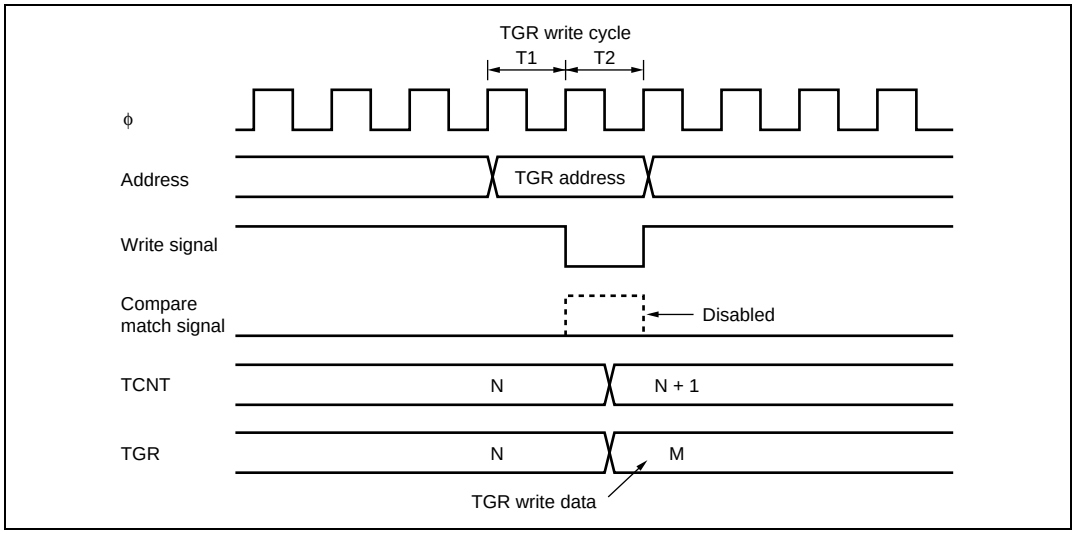


Figure 9.47 Contention between TGR Write and Compare Match

9.9.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write.

Figure 9.48 shows the timing in this case.

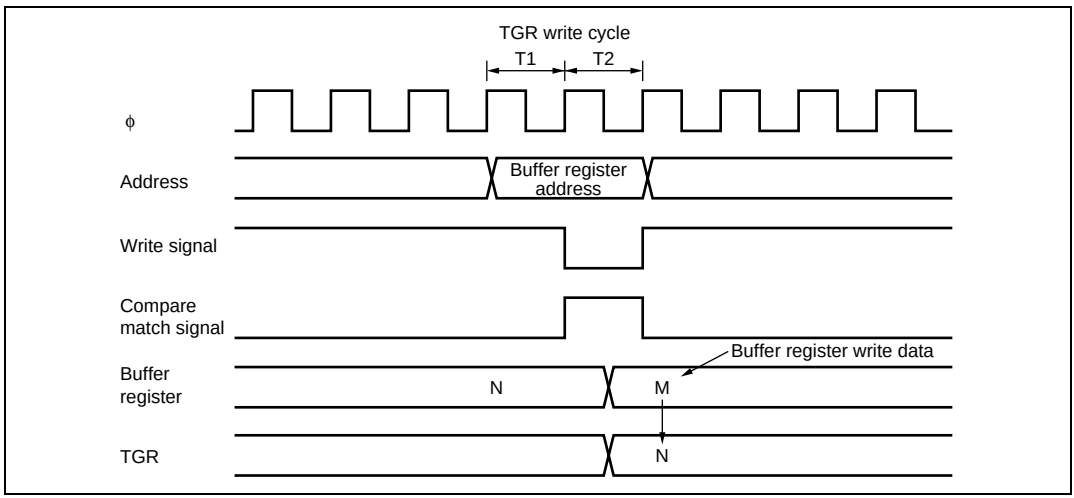


Figure 9.48 Contention between Buffer Register Write and Compare Match

9.9.8 Contention between TGR Read and Input Capture

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer.

Figure 9.49 shows the timing in this case.

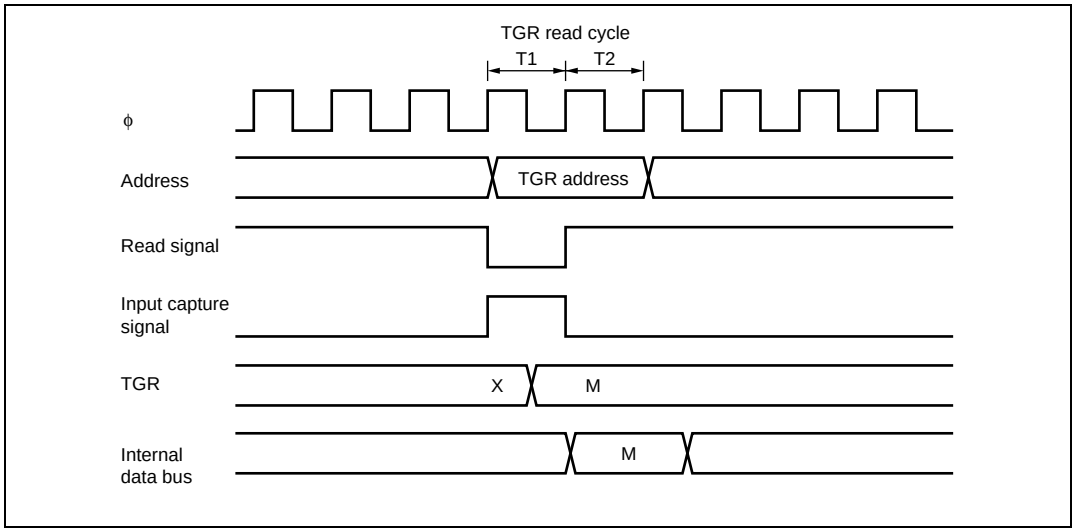


Figure 9.49 Contention between TGR Read and Input Capture

9.9.9 Contention between TGR Write and Input Capture

If the 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 9.50 shows the timing in this case.

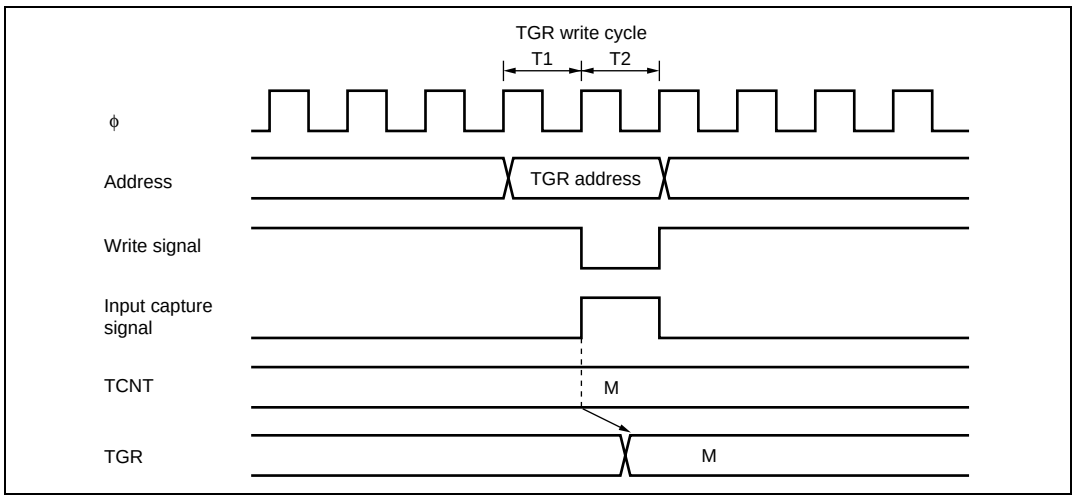


Figure 9.50 Contention between TGR Write and Input Capture

9.9.10 Contention between Buffer Register Write and Input Capture

If the 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 9.51 shows the timing in this case.

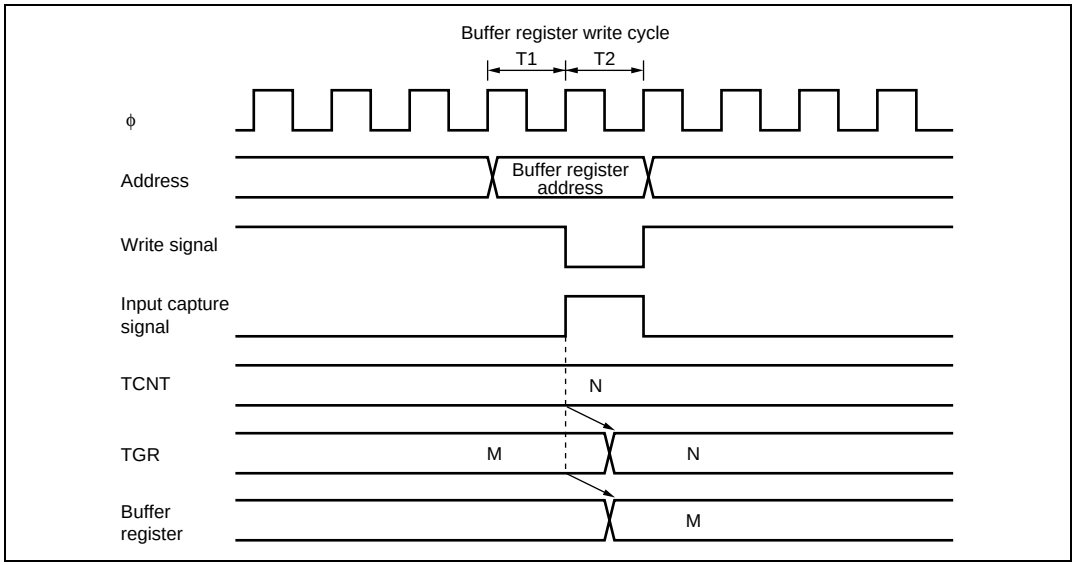


Figure 9.51 Contention between Buffer Register Write and Input Capture

9.9.11 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 9.52 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

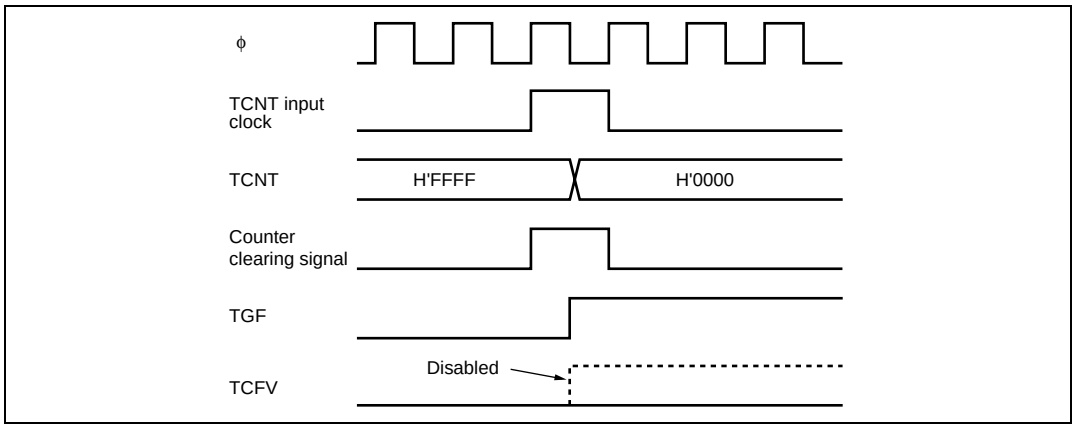


Figure 9.52 Contention between Overflow and Counter Clearing

9.9.12 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, when overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 9.53 shows the operation timing when there is contention between TCNT write and overflow.

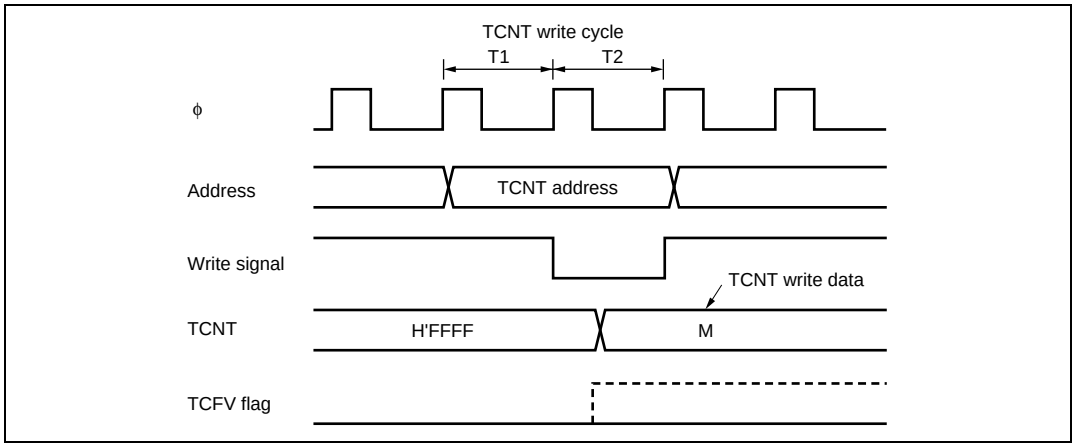


Figure 9.53 Contention between TCNT Write and Overflow

9.9.13 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.

9.9.14 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.

Section 10 Programmable Pulse Generator (PPG)

The programmable pulse generator (PPG) provides pulse outputs by using the 16-bit timer pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (groups 3 to 0) that can operate both simultaneously and independently. The block diagram of PPG is shown in figure 10.1

10.1 Features

- 16-bit output data
- Four output groups
- Selectable output trigger signals
- Non-overlap mode
- Can operate together with the data transfer controller (DTC)
- Settable inverted output
- Module stop mode can be set

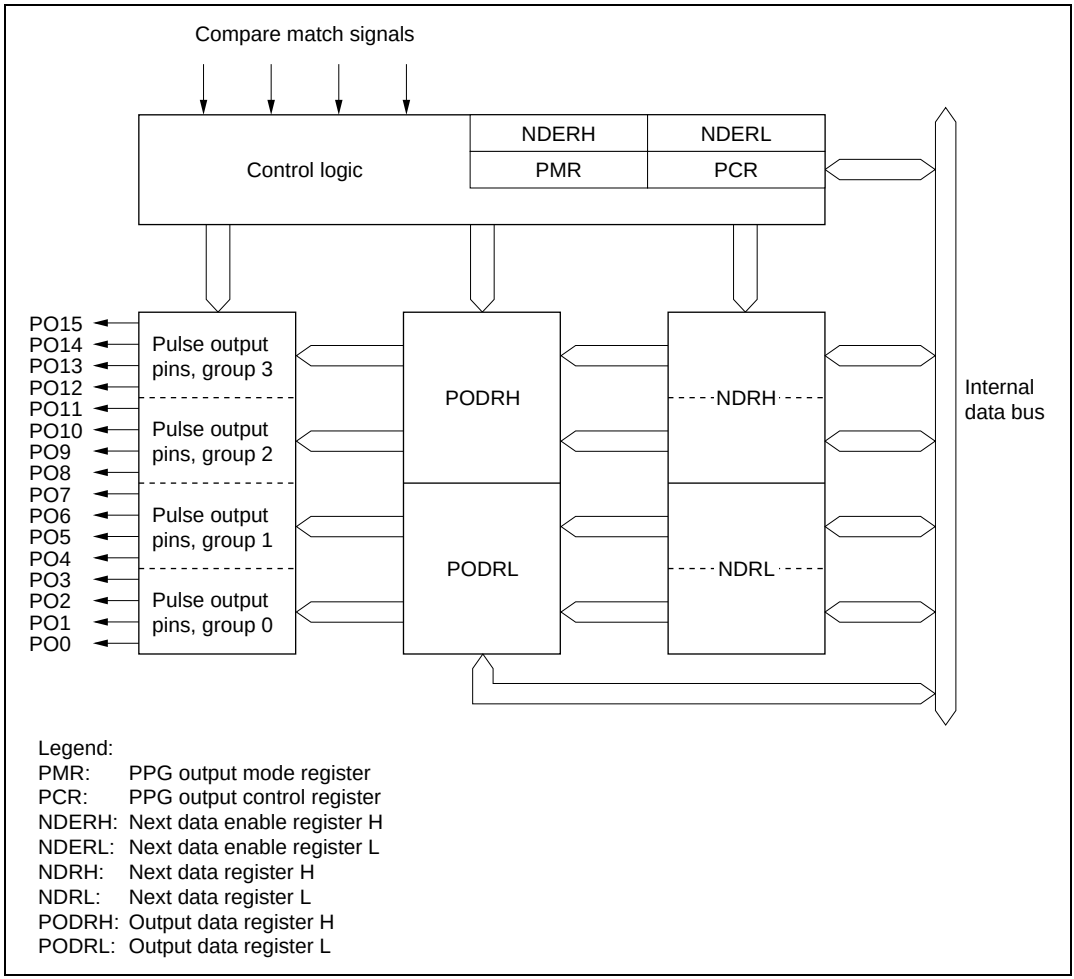


Figure 10.1 Block Diagram of PPG

10.2 Input/Output Pins

Table 10.1 shows the PPG pin configuration.

Table 10.1 Pin Configuration

Pin Name	I/O	Function
PO15	Output	Group 3 pulse output
PO14	Output	
PO13	Output	
PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	
PO7	Output	Group 1 pulse output
PO6	Output	
PO5	Output	
PO4	Output	
PO3	Output	Group 0 pulse output
PO2	Output	
PO1	Output	
PO0	Output	

10.3 Register Descriptions

The PPG has the following registers.

- Next data enable register H (NDERH)
- Next data enable register L (NDERL)
- Output data register H (PODRH)
- Output data register L (PODRL)
- Next data register H (NDRH)
- Next data register L (NDRL)
- PPG output control register (PCR)
- PPG output mode register (PMR)

10.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH, NDERL enable or disable pulse output on a bit-by-bit basis. For outputting pulse by the PPG, set the corresponding DDR to 1.

NDERH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 15 to 8
6	NDER14	0	R/W	When a bit is set to 1, the value in the corresponding NDRH bit is transferred to the PODRH bit by the selected output trigger. Values are not transferred from NDRH to PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

NDERL

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 7 to 0
6	NDER6	0	R/W	When a bit is set to 1, the value in the corresponding NDRL bit is transferred to the PODRL bit by the selected output trigger. Values are not transferred from NDRL to PODRL for cleared bits.
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

10.3.2 Output Data Registers H, L (PODRH, PODRL)

PODRH and PODRL store output data for use in pulse output. A bit that has been set for pulse output by NDER is read-only and cannot be modified.

PODRH

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 15 to 8
6	POD14	0	R/W	For bits which have been set to pulse output by NDERH, the output trigger transfers NDRH values to this register during PPG operation. While NDERH is set to 1, the CPU cannot write to this register. While NDERH is cleared, the initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

PODRL

Bit	Bit Name	Initial Value	R/W	Description
7	POD7	0	R/W	Output Data Register 7 to 0
6	POD6	0	R/W	For bits which have been set to pulse output by NDERL, the output trigger transfers NDRL values to this register during PPG operation. While NDERL is set to 1, the CPU cannot write to this register. While NDERL is cleared, the initial output value of the pulse can be set.
5	POD5	0	R/W	
4	POD4	0	R/W	
3	POD3	0	R/W	
2	POD2	0	R/W	
1	POD1	0	R/W	
0	POD0	0	R/W	

10.3.3 Next Data Registers H, L (NDRH, NDRL)

NDRH, NDRL store the next data for pulse output. The NDR addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

NDRH

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 8
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 12
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	Reserved Always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved Always read as 1 and cannot be modified.
3	NDR11	0	R/W	Next Data Register 11 to 8
2	NDR10	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
1	NDR9	0	R/W	
0	NDR8	0	R/W	

NDRL

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 0
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved Always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved Always read as 1 and cannot be modified.
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

10.3.4 PPG Output Control Register (PCR)

PCR selects output trigger signals on a group-by-group basis. For details on output trigger selection, refer to section 10.3.5, PPG Output Mode Register (PMR).

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 1 and 0
6	G3CMS0	1	R/W	Select output trigger of pulse output group 3. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 1 and 0
4	G2CMS0	1	R/W	Select output trigger of pulse output group 2. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
3	G1CMS1	1	R/W	Group 1 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	Select output trigger of pulse output group 1. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
1	G0CMS1	1	R/W	Group 0 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	Select output trigger of pulse output group 0. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3

10.3.5 PPG Output Mode Register (PMR)

PMR selects the pulse output mode of the PPG for each group. If inverted output is selected, a low-level pulse is output when PODRH is 1 and a high-level pulse is output when PODRH is 0. If non-overlapping operation is selected, PPG updates its output values at compare match A or B of the TPU that becomes the output trigger. For details, refer to section 10.4.4, Non-Overlapping Pulse Output.

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse output group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse output group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for pulse output group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for pulse output group 0. 0: Inverted output 1: Direct output

Bit	Bit Name	Initial Value	R/W	Description
3	G3NOV	0	R/W	Group 3 Non-Overlap Selects normal or non-overlapping operation for pulse output group 3. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
2	G2NOV	0	R/W	Group 2 Non-Overlap Selects normal or non-overlapping operation for pulse output group 2. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
1	G1NOV	0	R/W	Group 1 Non-Overlap Selects normal or non-overlapping operation for pulse output group 1. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)
0	G0NOV	0	R/W	Group 0 Non-Overlap Selects normal or non-overlapping operation for pulse output group 0. 0: Normal operation (output values updated at compare match A in the selected TPU channel) 1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)

10.4 Operation

Figure 10.2 shows an overview diagram of the PPG. PPG pulse output is enabled when the corresponding bits in P1DDR, P2DDR, and NDER are set to 1. An initial output value is determined by its corresponding PODR initial setting. When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values. Sequential output of data of up to 16 bits is possible by writing new output data to NDR before the next compare match.

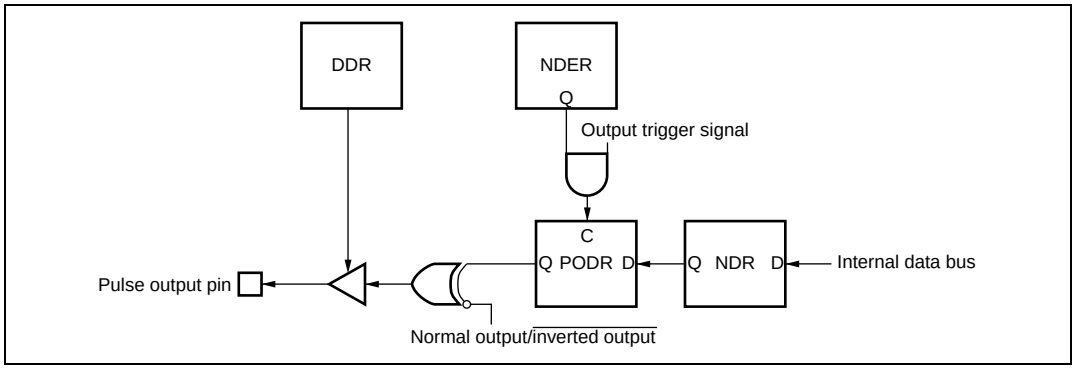


Figure 10.2 Overview Diagram of PPG

10.4.1 Output Timing

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 10.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

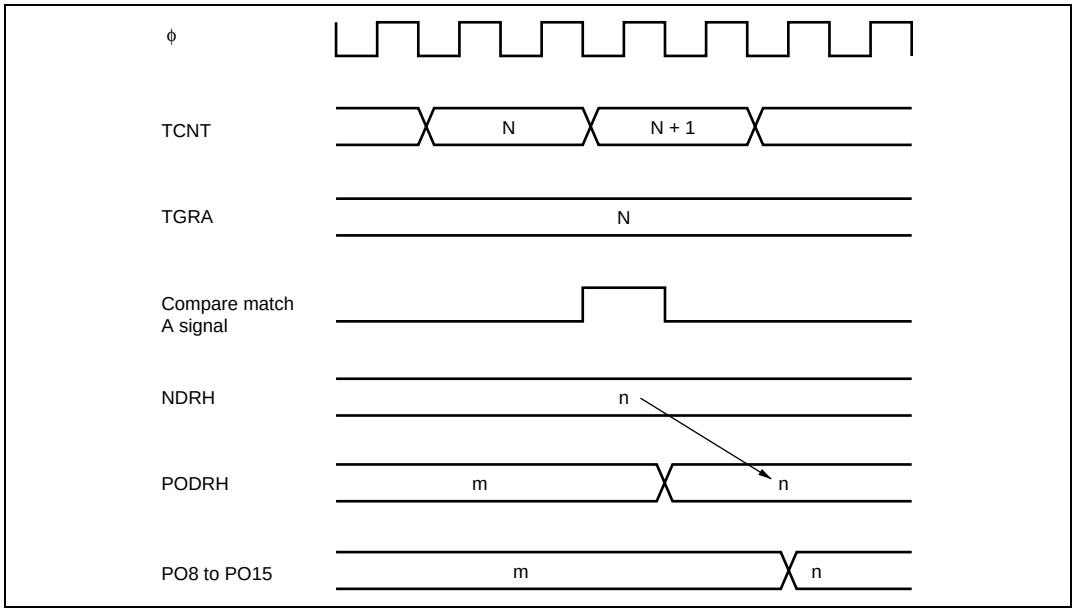


Figure 10.3 Timing of Transfer and Output of NDR Contents (Example)

10.4.2 Sample Setup Procedure for Normal Pulse Output

Figure 10.4 shows a sample procedure for setting up normal pulse output.

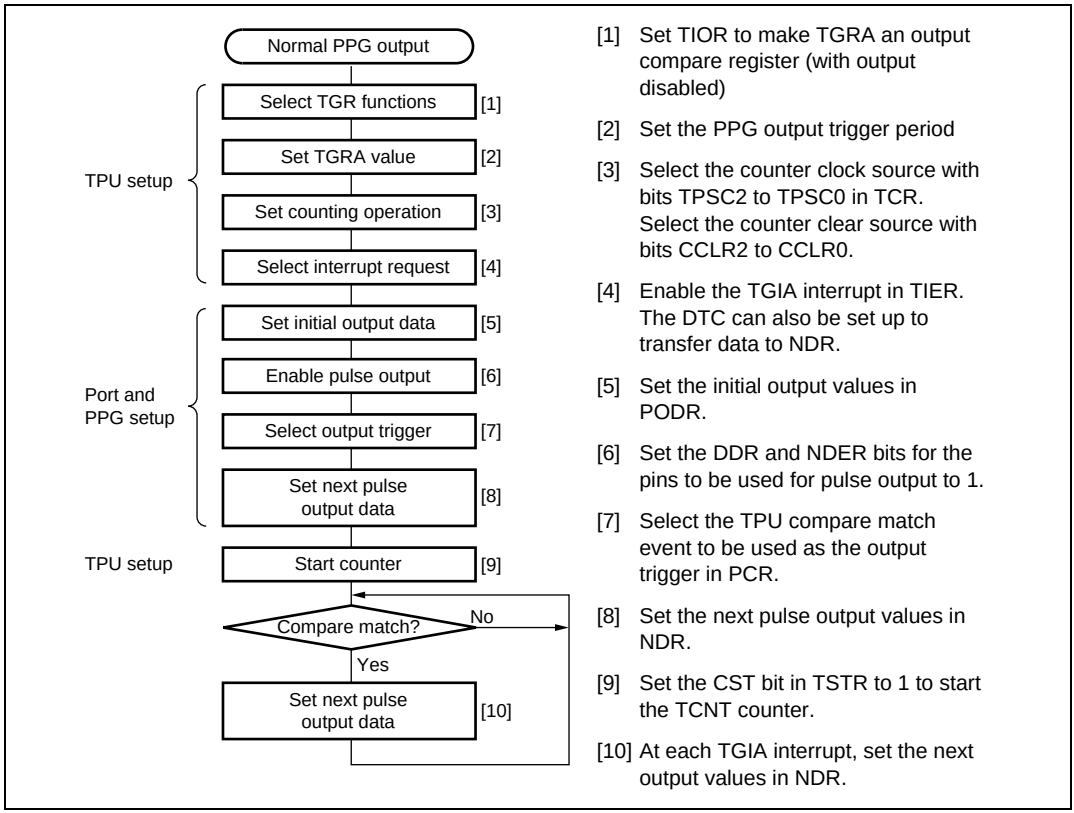


Figure 10.4 Setup Procedure for Normal Pulse Output (Example)

10.4.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 10.5 shows an example in which pulse output is used for cyclic five-phase pulse output.

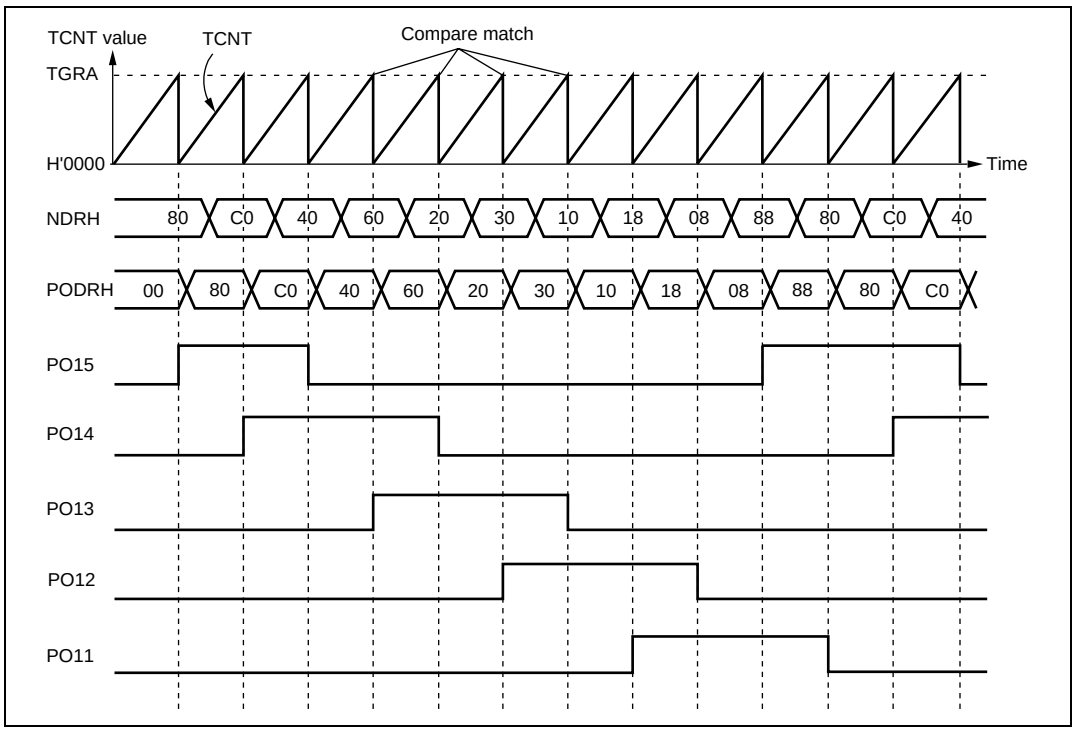


Figure 10.5 Normal Pulse Output Example (Five-Phase Pulse Output)

1. Set up TGRA in TPU which is used as the output trigger to be an output compare register. Set a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA bit in TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupts.

   If the DTC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

10.4.4 Non-Overlapping Pulse Output

During non-overlapping operation, transfer from NDR to PODR is performed as follows:

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 10.6 illustrates the non-overlapping pulse output operation.

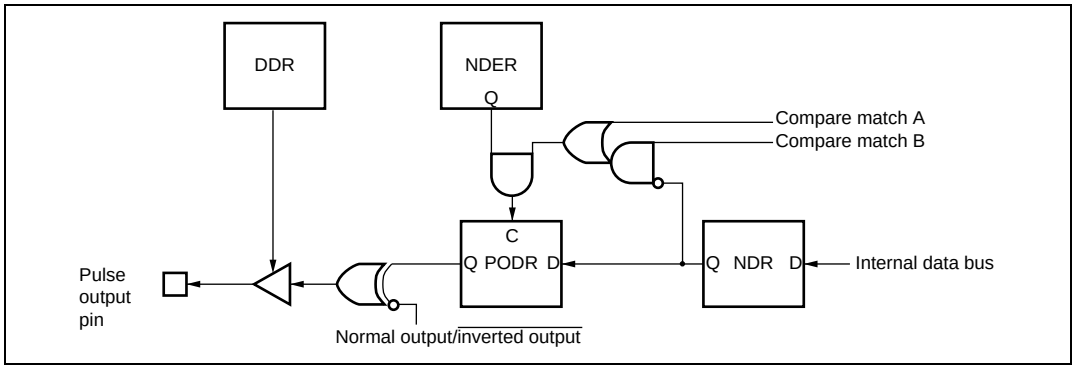


Figure 10.6 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A.

The NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 10.7 shows the timing of this operation.

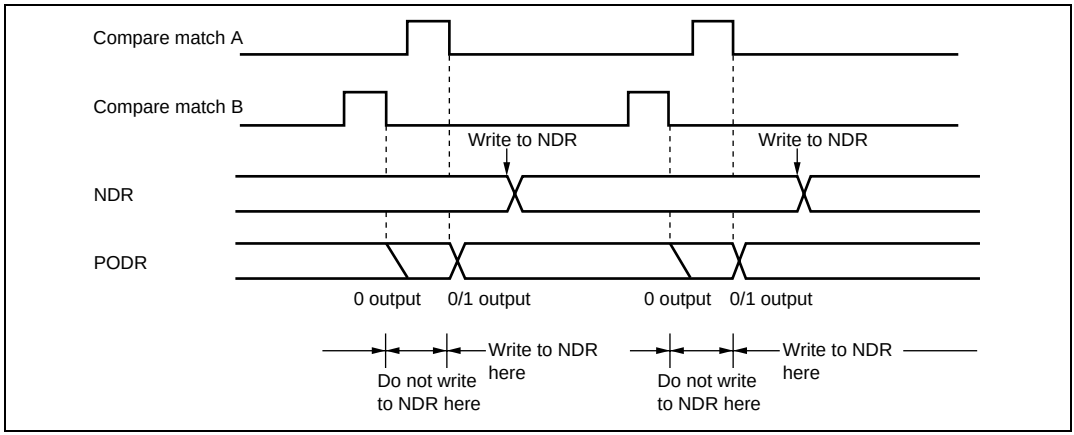


Figure 10.7 Non-Overlapping Operation and NDR Write Timing

10.4.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 10.8 shows a sample procedure for setting up non-overlapping pulse output.

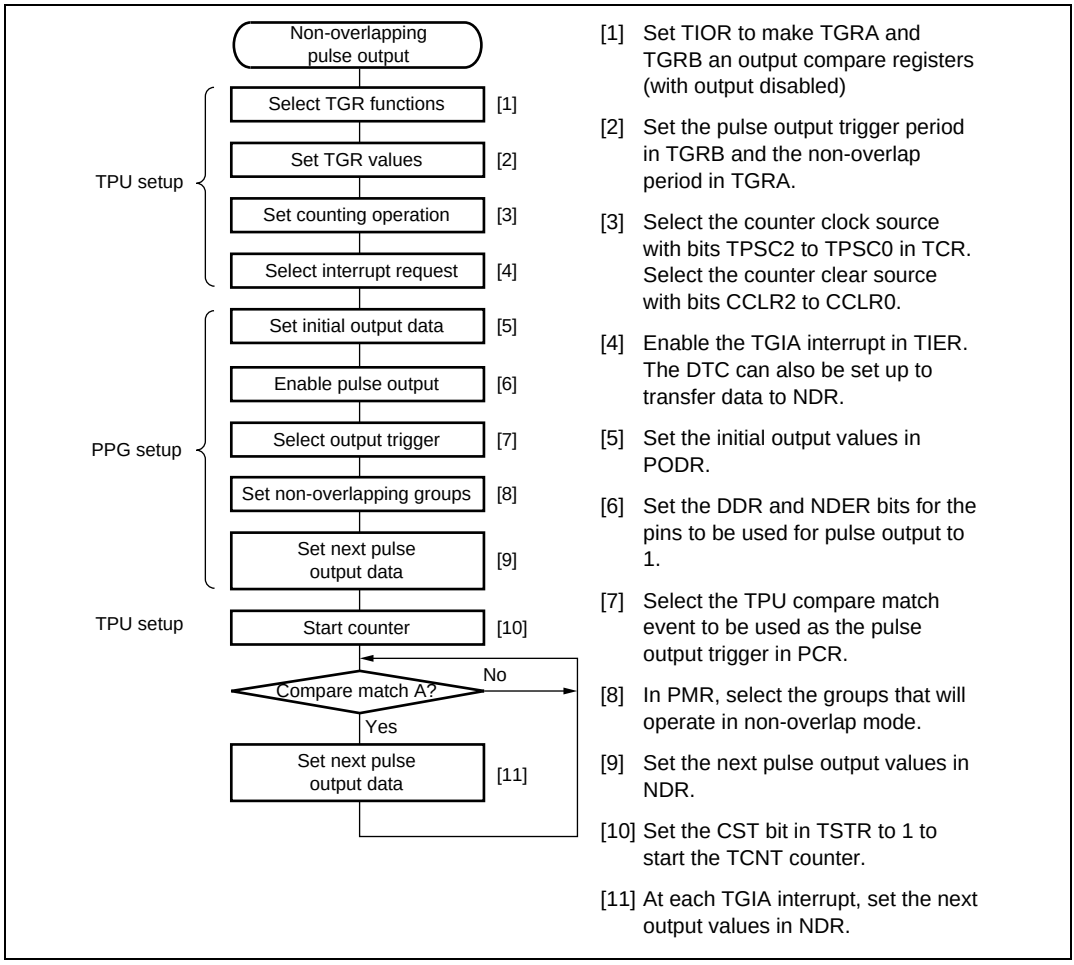


Figure 10.8 Setup Procedure for Non-Overlapping Pulse Output (Example)

10.4.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 10.9 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.

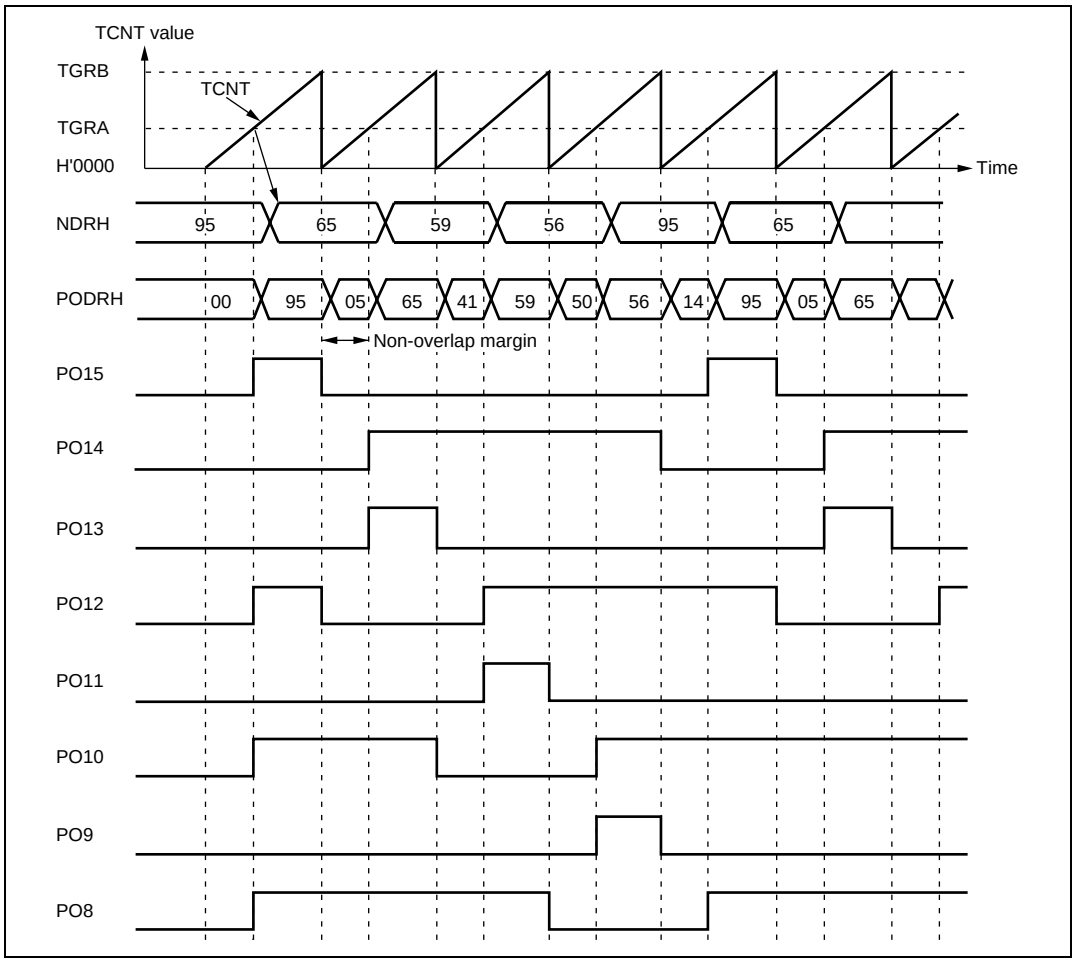


Figure 10.9 Non-Overlapping Pulse Output Example (Four-Phase Complementary)

1. Set up the TPU channel to be used as the output trigger channel so that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.
2. Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.
3. The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts.

   If the DTC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

10.4.7 Inverted Pulse Output

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 10.10 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 10.9.

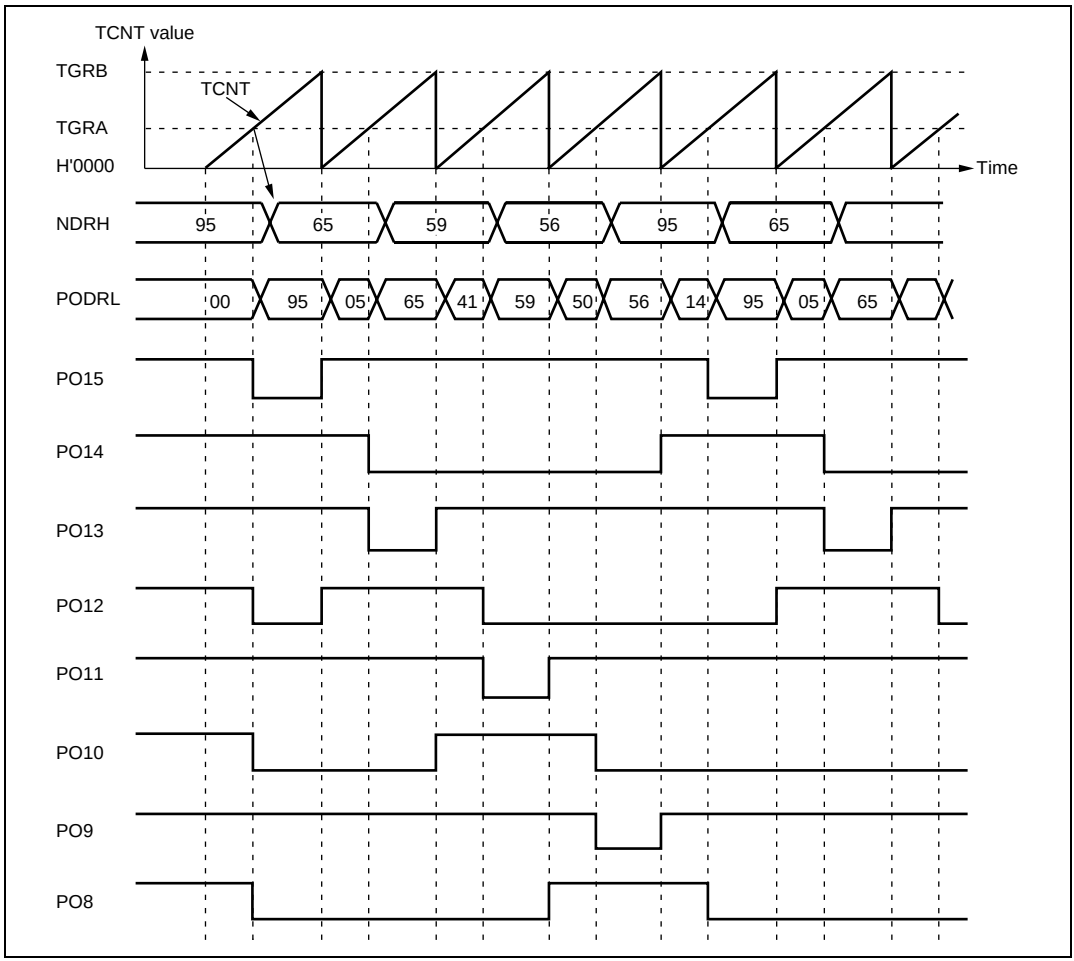


Figure 10.10 Inverted Pulse Output (Example)

10.4.8 Pulse Output Triggered by Input Capture

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 10.11 shows the timing of this output.

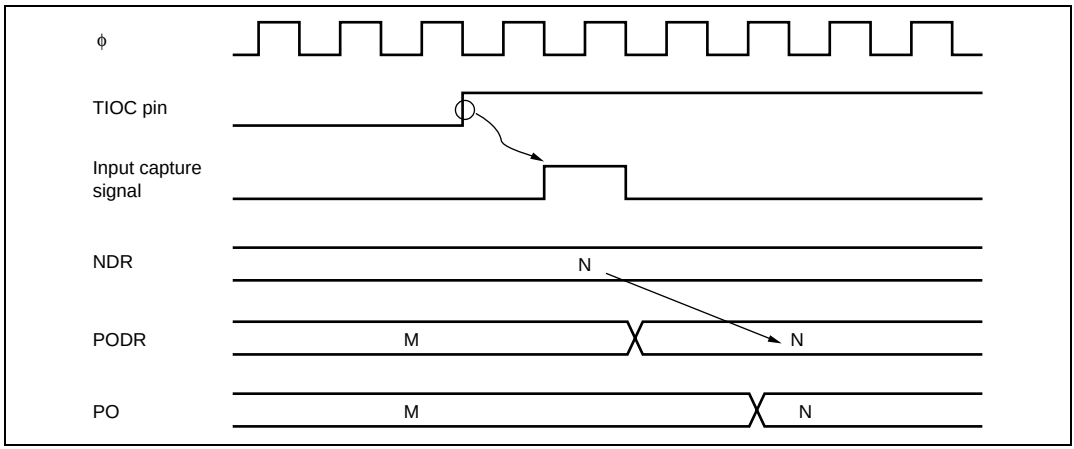


Figure 10.11 Pulse Output Triggered by Input Capture (Example)

10.5 Usage Notes

10.5.1 Module Stop Mode Setting

PPG operation can be disabled or enabled using the module stop control register. The initial value is for PPG operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 19, Power-Down Modes.

10.5.2 Operation of Pulse Output Pins

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

Section 11 8-Bit Timers (TMR)

This LSI has an on-chip 8-bit timer module with two channels operating on the basis of an 8-bit counter. 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

  The counters can be driven by one of three internal clock signals (ϕ/8, ϕ/64, or ϕ/8192) or an external clock input
- 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 control by a combination of two compare match signals

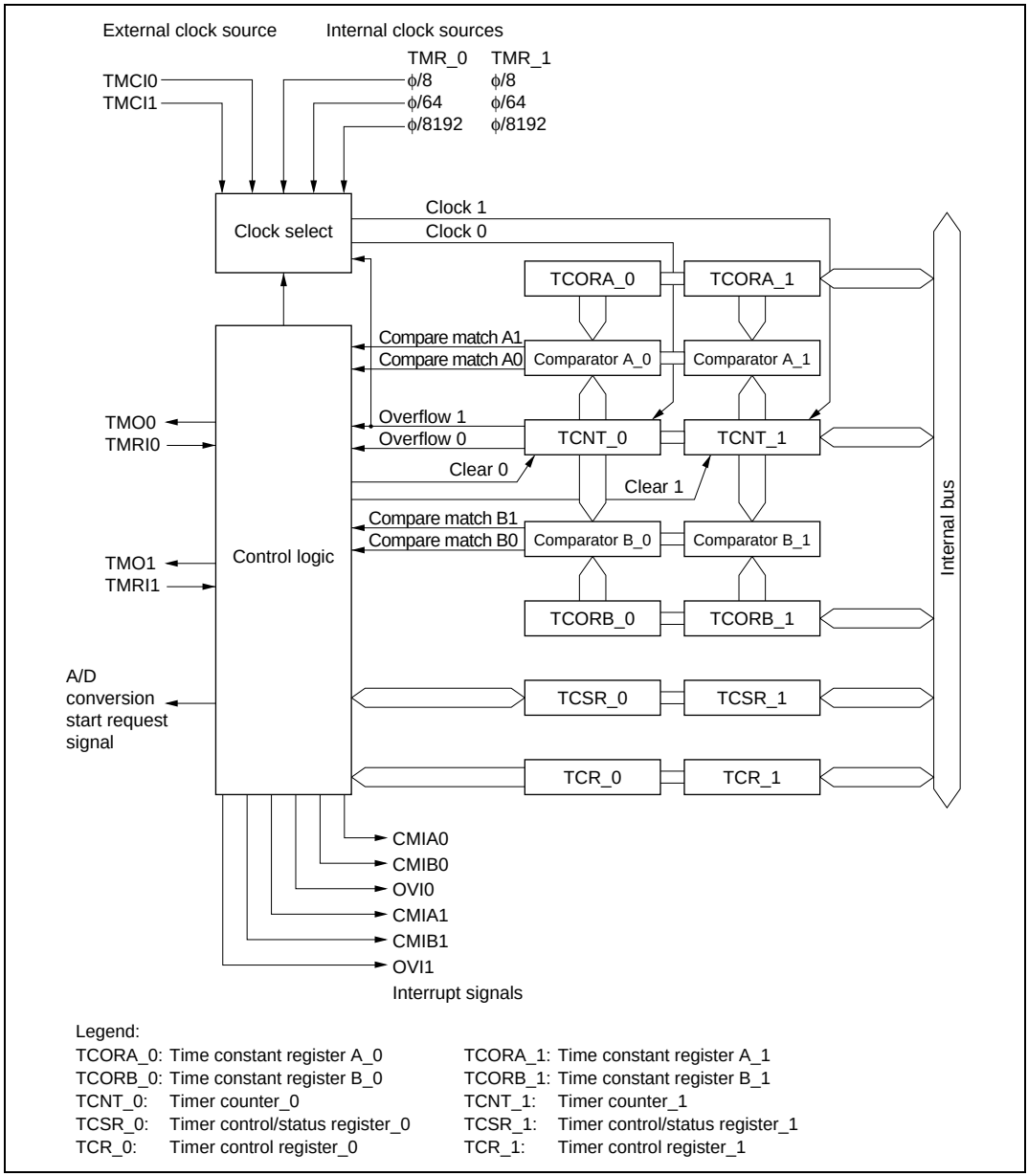
  The timer output signal in each channel is controlled by a combination of two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output
- Provision for cascading of two channels (TMR_0 and TMR_1)

  Operation as a 16-bit timer is possible, using TMR_0 for the upper 8 bits and TMR_1 for the lower 8 bits (16-bit count mode)

  TMR_1 can be used to count TMR_0 compare matches (compare match count mode)
- Three independent interrupts

  Compare match A and B and overflow interrupts can be requested independently
- A/D converter conversion start trigger can be generated

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 shows the pin configuration of the 8-bit timer module.

Table 11.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Timer output pin	TMO0	Output	Outputs at compare match
	Timer clock input pin	TMCI0	Input	Inputs external clock for counter
	Timer reset input pin	TMRI0	Input	Inputs external reset to counter
1	Timer output pin	TMO1	Output	Outputs at compare match
	Timer clock input pin	TMCI1	Input	Inputs external clock for counter
	Timer reset input pin	TMRI1	Input	Inputs external reset to counter

11.3 Register Descriptions

The 8-bit timer module has the following registers. For details on the module stop control register, refer to section 19.1.2 Module Stop Control Registers H, L (MSTPCRH, MSTPCRL).

- 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)
- 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)

11.3.1 Timer Counter (TCNT)

TCNT is 8-bit up-counter. TCNT_0 and TCNT_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. Bits CKS2 to CKS0 in TCR are used to select a clock. TCNT can be cleared by an external reset input or by a compare match signal A or B. Which signal is to be used for clearing is selected by bits CCLR1 and CCLR0 in TCR. When TCNT overflows from H'FF to H'00, OVF in TCSR is set to 1. TCNT is initialized to H'00.

11.3.2 Time Constant Register A (TCORA)

TCORA is 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note, however, that comparison is disabled during the T2 state of a TCORA write cycle.

The timer output from the TMO pin can be freely controlled by this compare match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR.

TCORA is initialized to H'FF.

11.3.3 Time Constant Register B (TCORB)

TCORB is 8-bit readable/writable register. TCORB_0 and TCORB_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag in TCSR is set to 1. Note, however, that comparison is disabled during the T2 state of a TCOBR write cycle.

The timer output from the TMO pin can be freely controlled by this compare match signal (compare match B) and the settings of bits OS3 and OS2 in TCSR.

TCORB is initialized to H'FF.

11.3.4 Timer Control Register (TCR)

TCR selects the clock source and the time at which TCNT is cleared, and controls interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B Selects whether CMFB interrupt requests (CMIB) are enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled
6	CMIEA	0	R/W	Compare Match Interrupt Enable A Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt requests (CMIA) are disabled 1: CMFA interrupt requests (CMIA) are enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are 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: Clear by compare match A 10: Clear by compare match B 11: Clear by rising edge of external reset input
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock input to TCNT and count condition. See table 11.2.
0	CKS0	0	R/W	

Table 11.2 Clock Input to TCNT and Count Condition

Channel	TCR Bit 2 CKS2	TCR Bit 1 CKS1	TCR Bit 0 CKS0	Description
TMR_0	0	0	0	Clock input disabled
			1	Internal clock, counted at falling edge of ϕ/8
		1	0	Internal clock, counted at falling edge of ϕ/64
			1	Internal clock, counted at falling edge of ϕ/8192
	1	0	0	Count at TCNT_1 overflow signal*
TMR_1	0	0	0	Clock input disabled
			1	Internal clock, counted at falling edge of ϕ/8
		1	0	Internal clock, counted at falling edge of ϕ/64
			1	Internal clock, counted at falling edge of ϕ/8192
	1	0	0	Count at TCNT_0 compare match A*
All	1	0	1	External clock, counted at rising edge
		1	0	External clock, counted at falling edge
		1	1	External clock, counted at both rising and falling edges

Note: * If the count input of TMR_0 is the TCNT_1 overflow signal and that of TMR_1 is the TCNT_0 compare match signal, no incrementing clock is generated. Do not use this setting.

11.3.5 Timer Control/Status Register (TCSR)

TCSR displays status flags, and controls compare match output.

TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] • Set when TCNT matches TCORB [Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] • Set when TCNT matches TCORA [Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] Set when TCNT overflows from H'FF to H'00 [Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF
4	ADTE	0	R/W	A/D Trigger Enable Selects enabling or disabling of 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 select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 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 select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 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 bits 7 to 5, to clear these flags.

TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] • Set when TCNT matches TCORB [Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] • Set when TCNT matches TCORA [Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] Set when TCNT overflows from H'FF to H'00 [Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF
4	—	1	R	Reserved This bit is always read as 1 and cannot be modified.

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 select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 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 select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 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 bits 7 to 5, to clear these flags.

11.4 Operation

11.4.1 Pulse Output

Figure 11.2 shows an example that the 8-bit timer is used to generate a pulse output with a selected duty cycle. The control bits are set as follows:

[1] In TCR, bit CCLR1 is cleared to 0 and bit CCLR0 is set to 1 so that the timer counter is cleared at a TCORA compare match.

[2] In TCSR, bits OS3 to OS0 are set to B'0110, causing the output to change to 1 at a TCORA compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

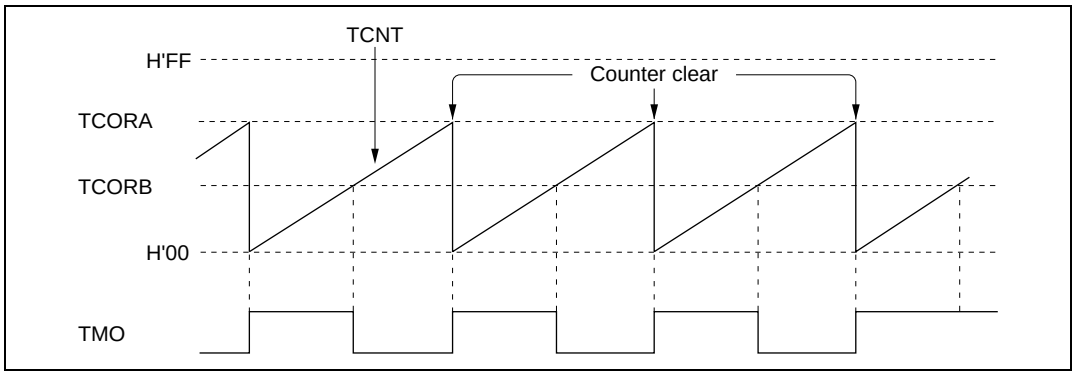


Figure 11.2 Example of Pulse Output

11.5 Operation Timing

11.5.1 TCNT Incrementation Timing

Figure 11.3 shows the count timing for internal clock input. Figure 11.4 shows the count timing for external clock signal. Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, 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.

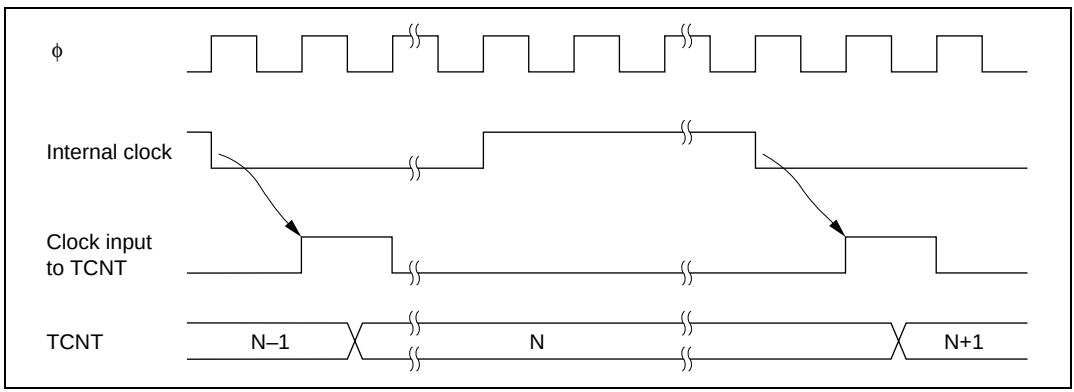


Figure 11.3 Count Timing for Internal Clock Input

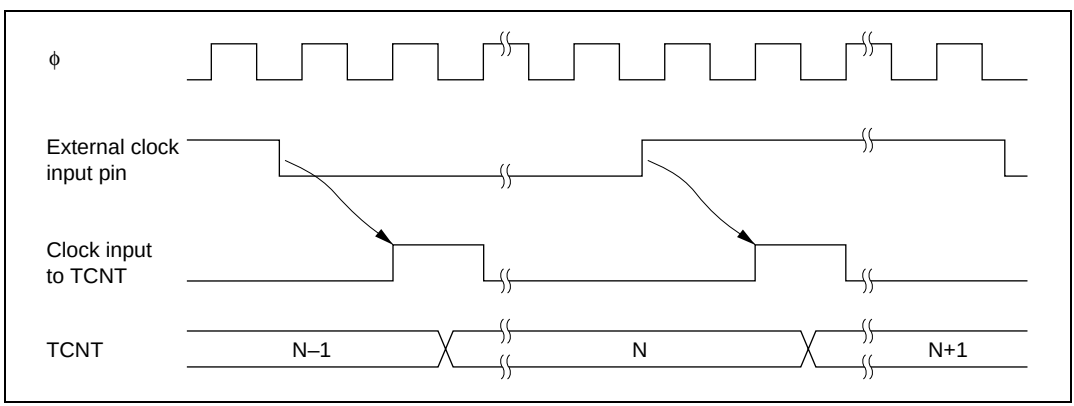


Figure 11.4 Count Timing for External Clock Input

11.5.2 Timing of CMFA and CMFB Setting when 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 this timing.

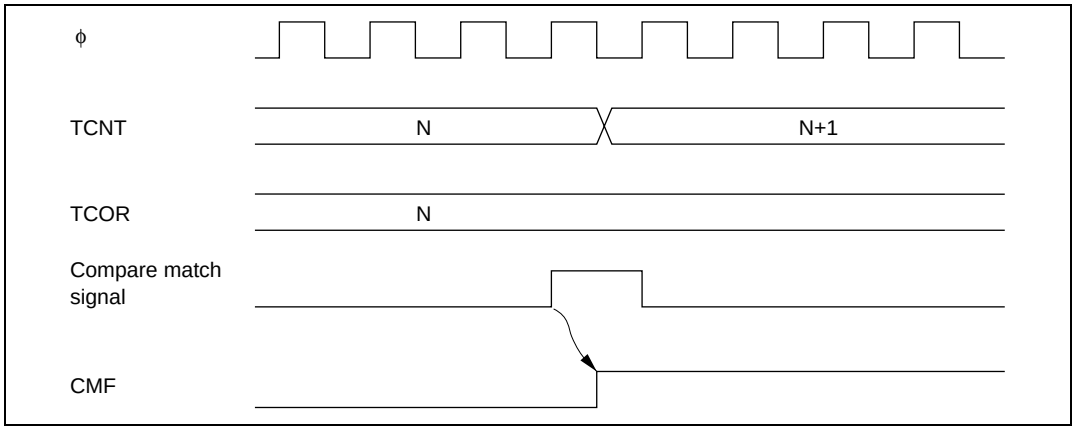


Figure 11.5 Timing of CMF Setting

11.5.3 Timing of Timer Output when Compare-Match Occurs

When compare match A or B occurs, the timer output changes as specified by bits OS3 to OS0 in TCSR.

Figure 11.6 shows the timing when the output is set to toggle at compare match A.

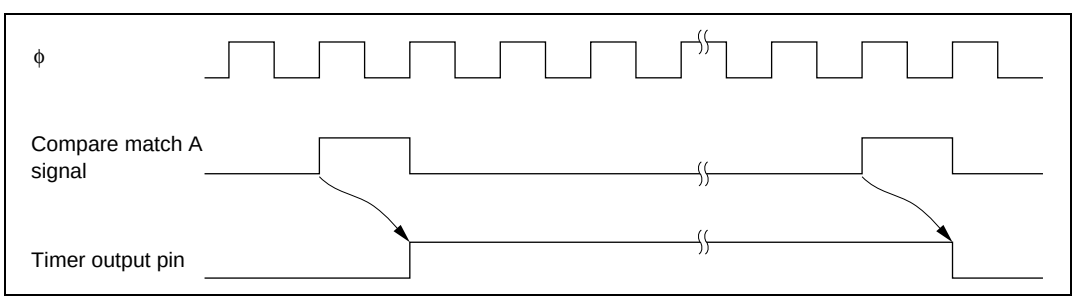


Figure 11.6 Timing of Timer Output

11.5.4 Timing of Compare Match Clear

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.

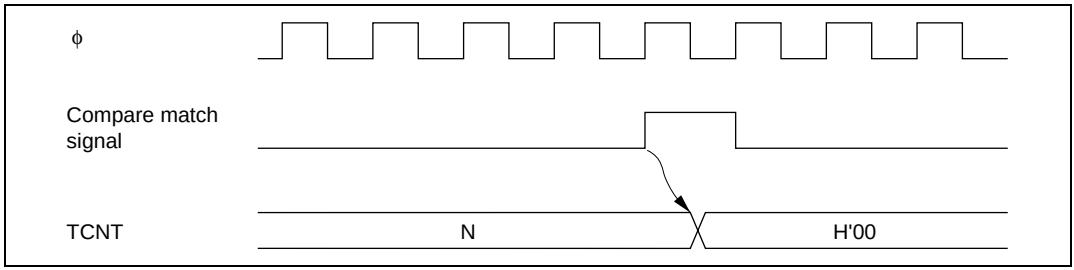


Figure 11.7 Timing of Compare Match Clear

11.5.5 Timing of TCNT External Reset

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 clear pulse width must be at least 1.5 states. Figure 11.8 shows the timing of this operation.

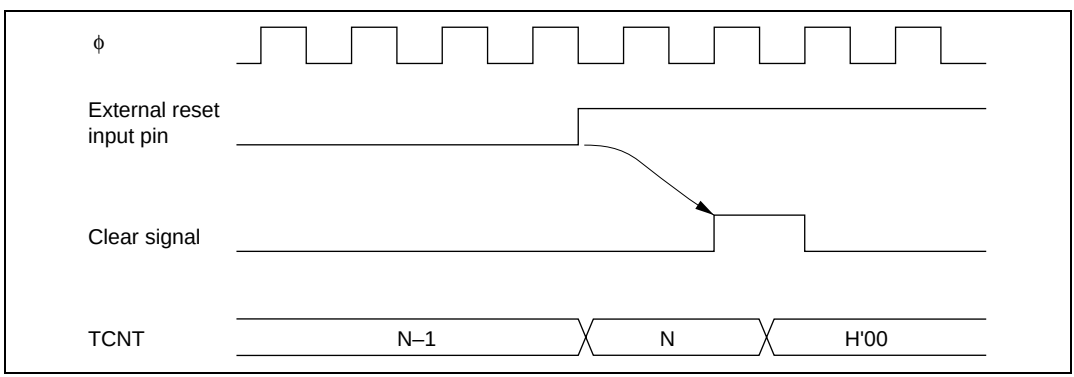


Figure 11.8 Timing of Clearance by External Reset

11.5.6 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when TCNT overflows (changes from H'FF to H'00). Figure 11.9 shows the timing of this operation.

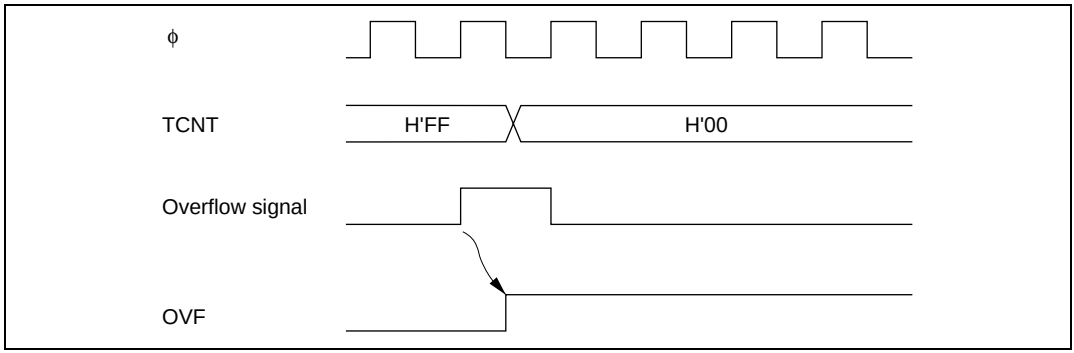


Figure 11.9 Timing of OVF Setting

11.6 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channel 1 (compare match count mode). In this case, the timer operates as below.

11.6.1 16-Bit Counter Mode

When bits CKS2 to CKS0 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.

[1] Setting of compare match flags

- The CMF flag in TCSR_0 is set to 1 when a 16-bit compare match event occurs.
- The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare match event occurs.

[2] Counter clear specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare match, the 16-bit counters (TCNT_0 and TCNT_1 together) are cleared when a 16-bit compare match event occurs. The 16-bit counters (TCNT0 and TCNT1 together) are cleared even if counter clear by the TMRI0 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.

[3] Pin output

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordance with the lower 8-bit compare match conditions.

11.6.2 Compare Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B'100, TCNT_1 counts compare match A’s 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 clear are in accordance with the settings for each channel.

11.7 Interrupt Sources

11.7.1 Interrupt Sources and DTC Activation

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative priorities are shown in table 11.3. Each interrupt source is set as enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, and independent interrupt requests are sent for each to the interrupt controller. It is also possible to activate the DTC by means of CMIA and CMIB interrupts.

Table 11.3 8-Bit Timer Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
CMIA0	TCORA_0 compare match	CMFA	Possible	High
CMIB0	TCORB_0 compare match	CMFB	Possible	↑
OVI0	TCNT_0 overflow	OVF	Not possible	Low
CMIA1	TCORA_1 compare match	CMFA	Possible	High
CMIB1	TCORB_1 compare match	CMFB	Possible	↑
OVI1	TCNT_1 overflow	OVF	Not possible	Low

11.7.2 A/D Converter Activation

The A/D converter can be activated only by TMR_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 TMR_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.

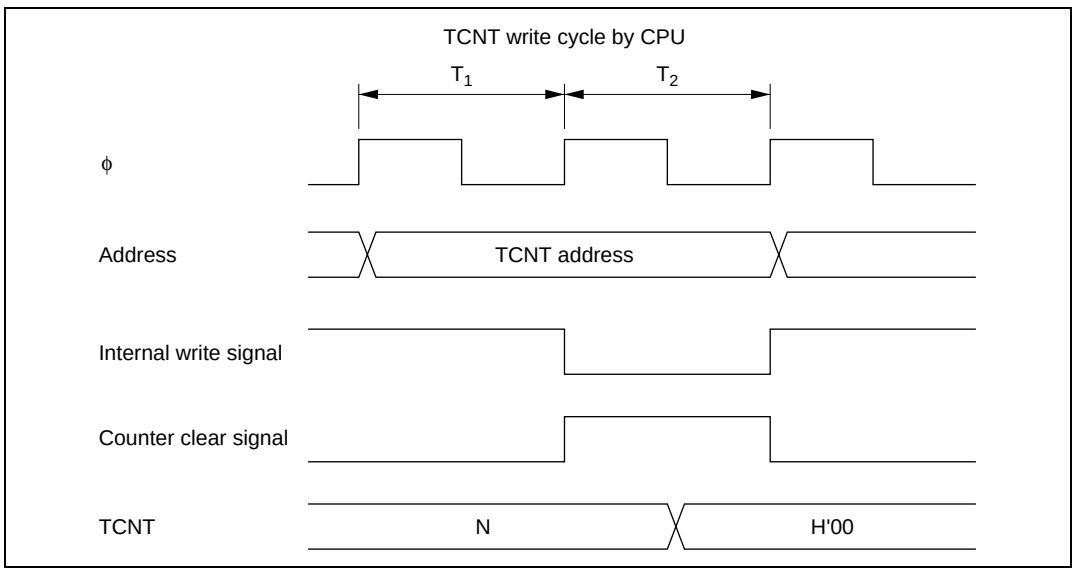


Figure 11.10 Contention between TCNT Write and Clear

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.

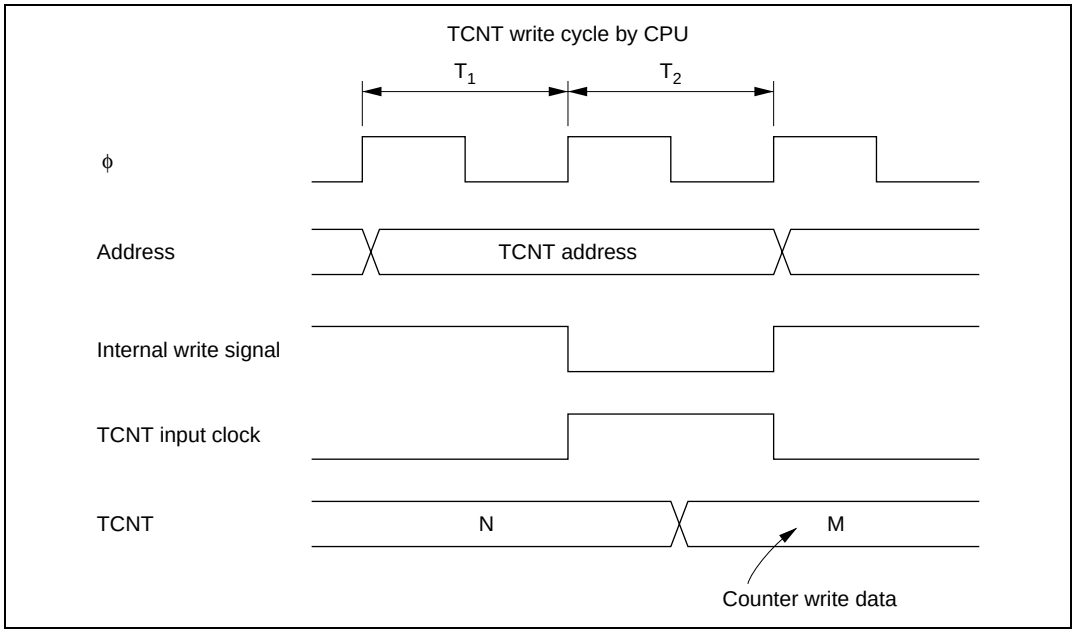


Figure 11.11 Contention between TCNT Write and Increment

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 and the compare match signal is inhibited even if a compare match event occurs as shown in figure 11.12.

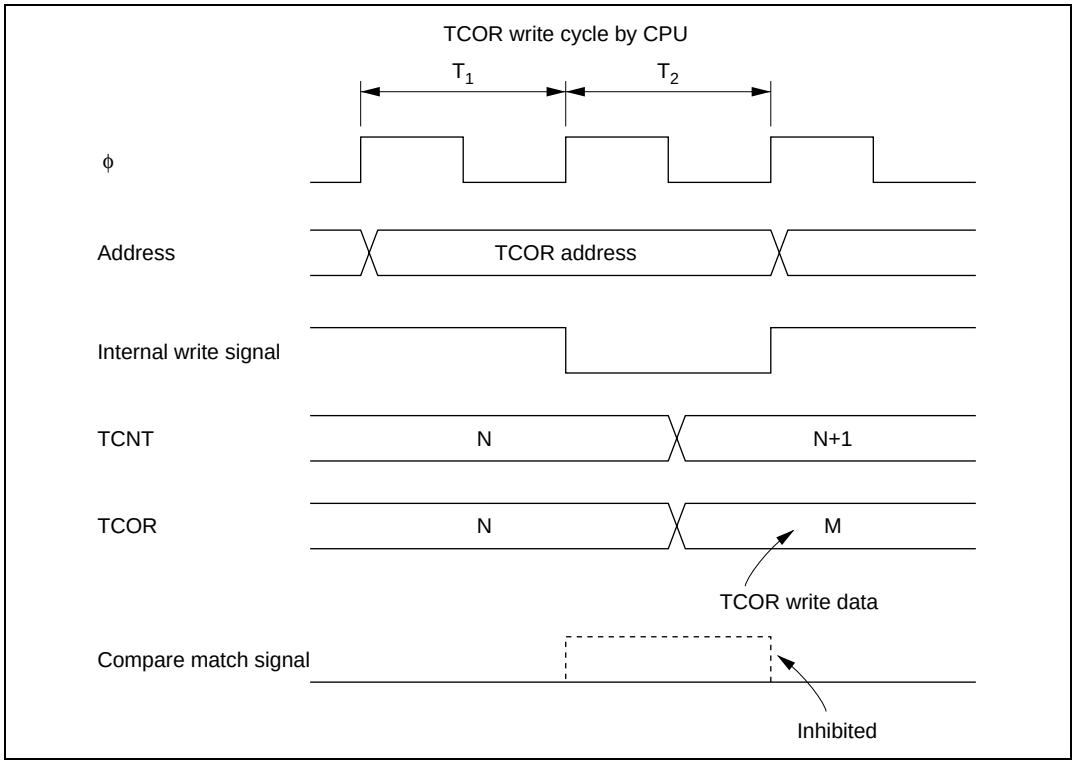


Figure 11.12 Contention between TCOR Write and Compare Match

11.8.4 Contention between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in table 11.4.

Table 11.4 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

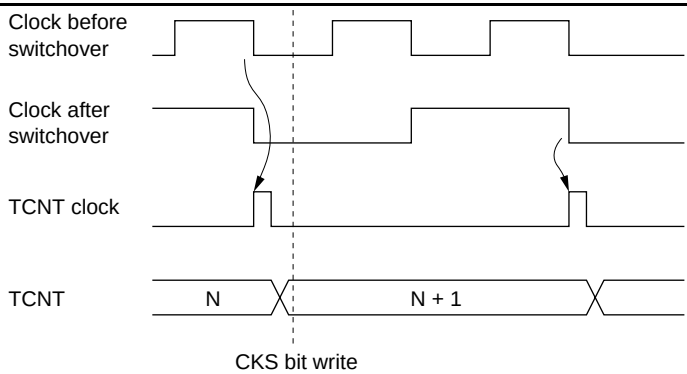
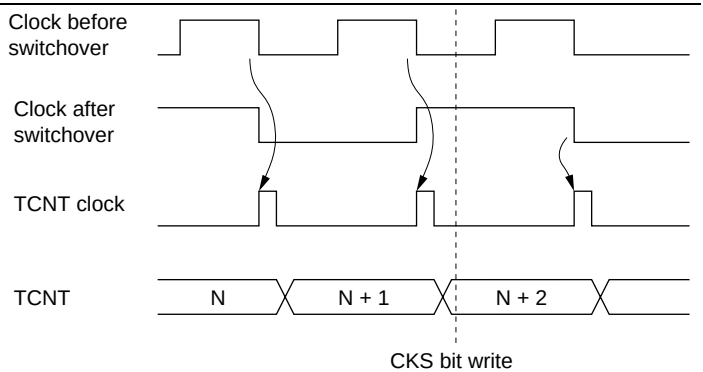
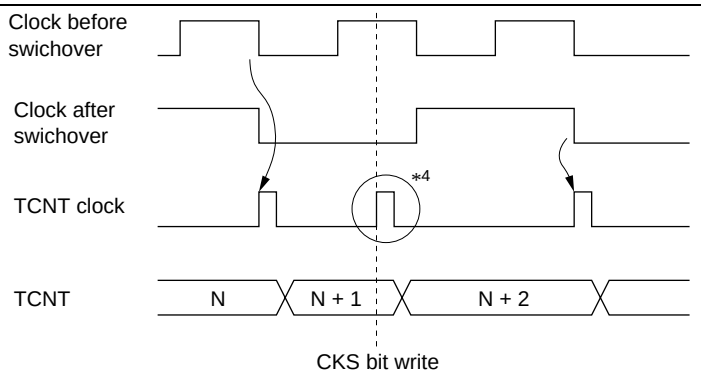
11.8.5 Switching of Internal Clocks and TCNT Operation

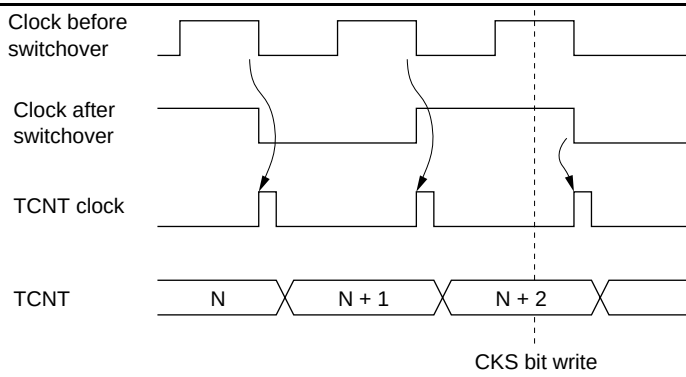
TCNT may increment erroneously when the internal clock is switched over. Table 11.5 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

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 case 3 in table 11.5, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

The erroneous incrementation can also happen when switching between internal and external clocks.

Table 11.5 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	
3	Switching from high to low*3	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
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 Mode Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TCNT_0 and TCNT_1 are not generated, and the counter stops. Do not specify 16-bit counter and compare match count modes simultaneously.

11.8.7 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 12 Watchdog Timer

The watchdog timer (WDT) is an 8-bit timer that outputs an overflow signal ($\overline{\text{WDTOVF}}$) if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

At the same time, the WDT can also generate an internal reset signal.

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
- Switchable between watchdog timer mode and interval timer mode

In watchdog timer mode

- If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whether or not the entire chip is reset at the same time.

In interval timer mode

- If the counter overflows, the WDT generates an interval timer interrupt (WOVI).

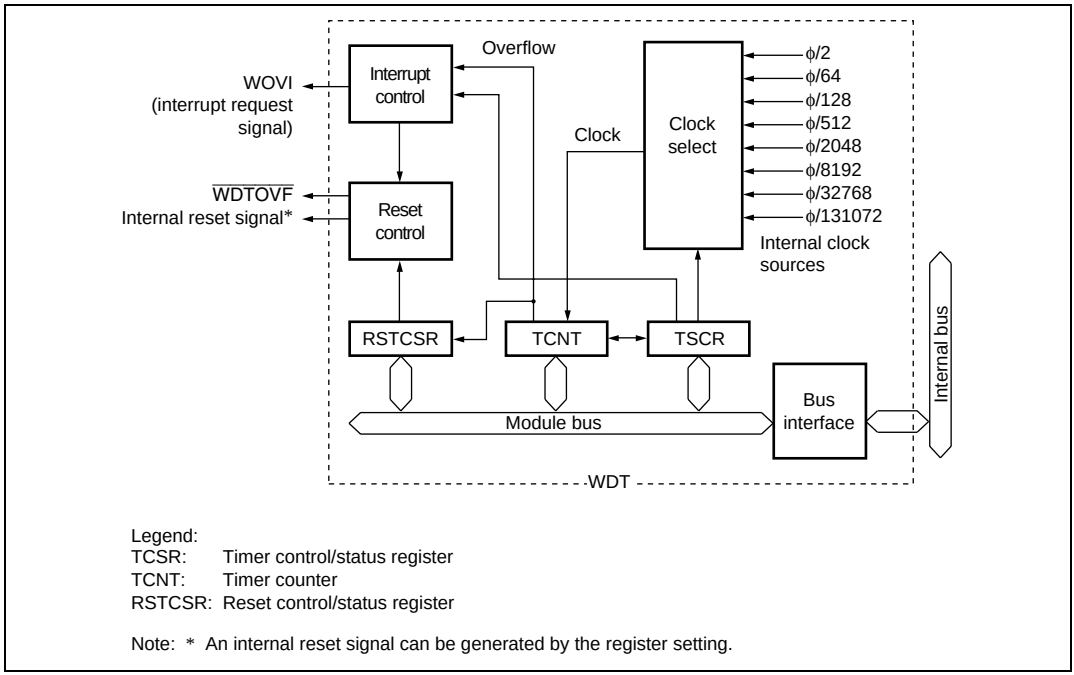


Figure 12.1 Block Diagram of WDT

12.2 Input/Output Pin

Table 12.1 shows the WDT pin configuration.

Table 12.1 Pin Configuration

Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs counter overflow signal in watchdog timer mode

12.3 Register Descriptions

The WDT has the following three registers. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to in a method different from normal registers. For details, refer to section 12.6.1, Notes on Register Access.

- 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 selects the clock source to be input to TCNT, and the timer mode.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed in interval timer mode. Only a write of 0 is permitted, to clear the flag. [Setting condition] When TCNT overflows in interval timer mode (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF

Bit	Bit Name	Initial Value	R/W	Description
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 When TCNT overflows, an interval timer interrupt (WOVI) is requested. 1: Watchdog timer mode When TCNT overflows, the $\overline{WDTOVF}$ signal is output.
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 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow frequency for ϕ = 20 MHz is enclosed in parentheses.
0	CKS0	0	R/W	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)

Note: * Only a write of 0 is permitted, to clear the flag.

12.3.3 Reset Control/Status Register (RSTCSR)

RSTCSR controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal. RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, but not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Timer 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. [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	—	0	R/W	Reserved Can be read and written, but does not affect operation.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Note: * Only a write of 0 is permitted, to clear the flag.

12.4 Operation

12.4.1 Watchdog Timer Mode

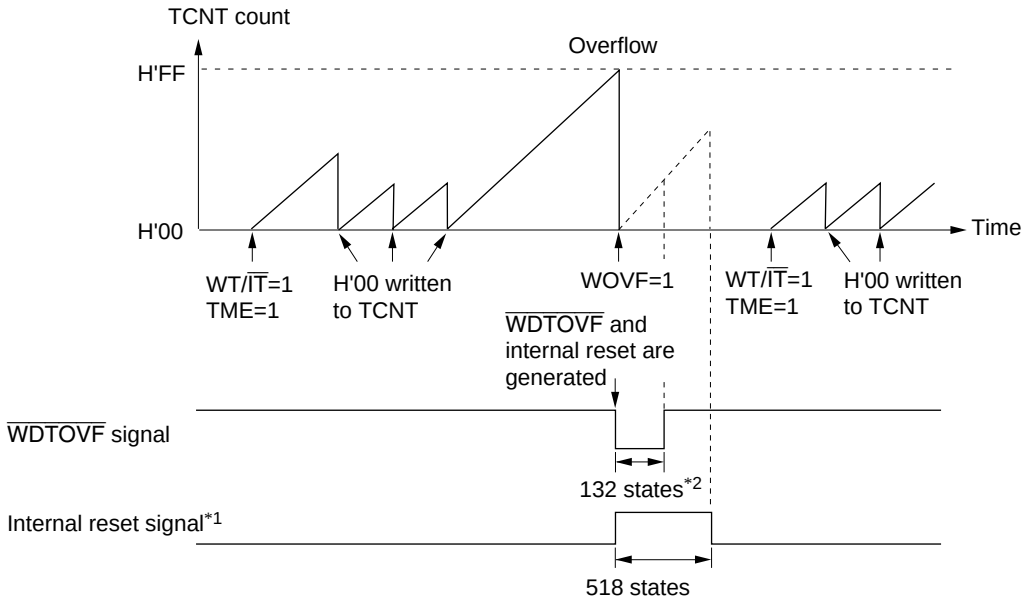
To use the WDT as a watchdog timer mode, set the WT/$\overline{\text{IT}}$ and TME bits in TCSR to 1.

If TCNT overflows without being rewritten because of a system crash or other error, the $\overline{\text{WDTOVF}}$ signal is output. This ensures that TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflow occurs. This $\overline{\text{WDTOVF}}$ signal can be used to reset the chip internally in watchdog timer mode.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets this LSI internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal. If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

The $\overline{\text{WDTOVF}}$ signal is output for 132 states when RSTE = 1, and for 130 states when RSTE = 0. The internal reset signal is output for 518 states.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, an internal reset signal is generated to the entire chip.



Legend:
WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit

Notes: 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset signal is generated.
2. 130 states when the RSTE bit is cleared to 0.

Figure 12.2 Operation in Watchdog Timer Mode

12.4.2 Interval Timer Mode

To use the WDT as an interval timer, set the WT/$\overline{IT}$ bit to 0 and TME bit in TCSR 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 intervals.

When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the same time the OVF bit in the TCSR is set to 1.

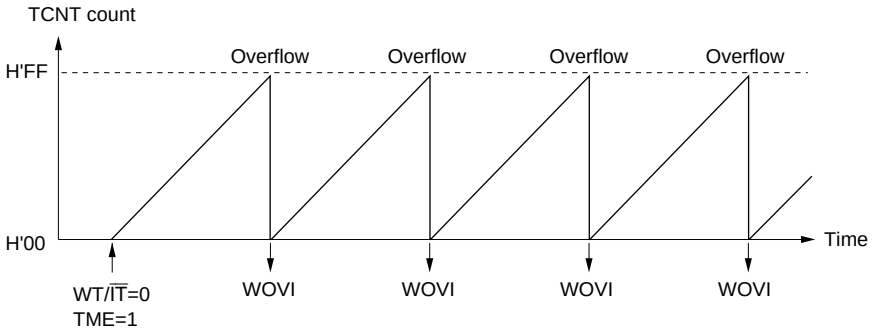


Figure 12.3 Operation in Interval Timer Mode

12.5 Interrupt Source

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

Table 12.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	OVF	Impossible

12.6 Usage Notes

12.6.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR 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

TCNT and TCSR 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, satisfy the relative condition shown in figure 12.4 to write to TCNT or TCSR. The transfer instruction writes the lower byte data to TCNT or TCSR according to the satisfied condition.

To write to RSTCSR, execute a word transfer instruction for address H'FFBE. A byte transfer instruction cannot perform writing to RSTCSR.

The method of writing 0 to the WOVF bit differs from that of writing to the RSTE bit. To write 0 to the WOVF bit, satisfy the lower condition shown in figure 12.4.

If satisfied, the transfer instruction clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, satisfy the above condition shown in figure 12.4. If satisfied, the transfer instruction writes the value in bit 6 of the lower byte into the RSTE bit, but has no effect on the WOVF bit.

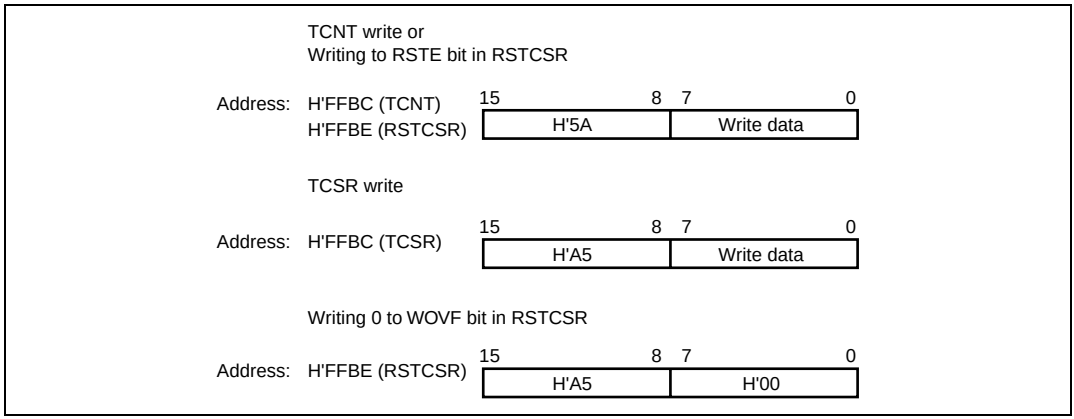


Figure 12.4 Writing to TCNT, TCSR, and RSTCSR

Reading TCNT, TCSR, and RSTCSR

These registers are read in the same way as other registers. The read addresses are H'FFBC for TCSR, H'FFBD for TCNT, and H'FFBF for RSTCSR.

12.6.2 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the next cycle after the T2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 12.5 shows this operation.

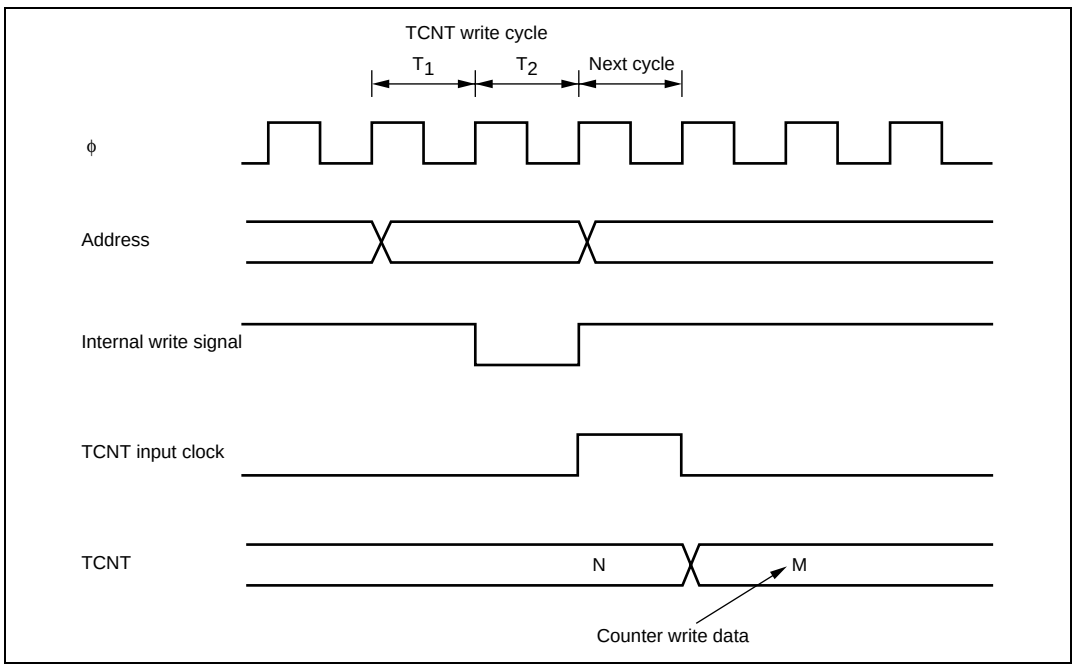


Figure 12.5 Contention between TCNT Write and Increment

12.6.3 Changing Value of CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS2 to CKS0.

12.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

12.6.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer mode operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Also note that a read of the WOVF flag is not recognized during this period. To clear the WOVF flag, therefore, read TCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF flag.

12.6.6 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ output signal is input to the $\overline{\text{RES}}$ pin, the chip will not be initialized correctly. Make sure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin.

To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 12.6.

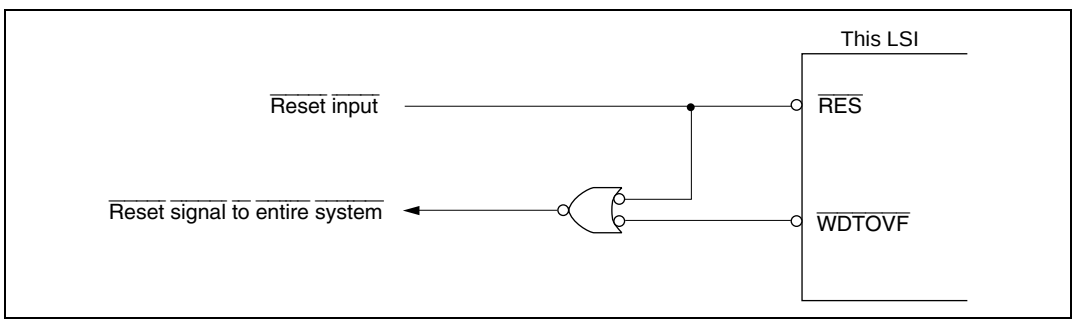


Figure 12.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

Section 13 Serial Communication Interface (SCI, IrDA)

This LSI has three independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function) in asynchronous mode. The SCI also supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as an asynchronous serial communication interface extension function. One of the three SCI channels (SCI_0) can generate an IrDA communication waveform conforming to IrDA specification version 1.0.

Figure 13.1 shows a block diagram of the SCI.

13.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used 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 for 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

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests. The transmit-data-empty interrupt and receive data full interrupts can 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 case of a framing error
- Average transfer rate generator (only for SCI_2)
  115.152 or 460.606 kbps at 10.667 MHz operation
  115.196, 460.784 or 720 kbps at 16 MHz operation
  720 kbps at 32 MHz operation

Clocked Synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

Smart Card Interface

- Automatic transmission of error signal (parity error) in receive mode
- Error signal detection and automatic data retransmission in transmit mode
- Direct convention and inverse convention both supported

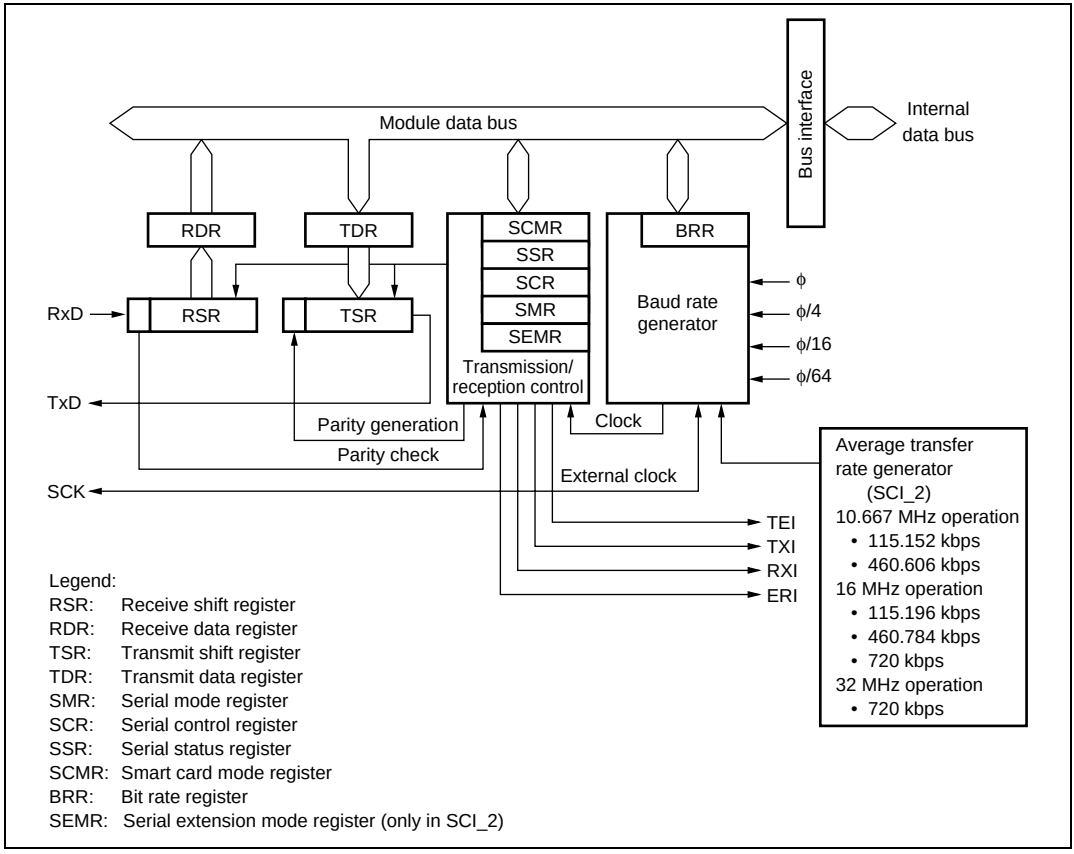


Figure 13.1 Block Diagram of SCI

13.2 Input/Output Pins

Table 13.1 shows the pin configuration of the serial communication interface.

Table 13.1 Pin Configuration

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	Channel 0 clock input/output
	RxD0/IrRxD	Input	Channel 0 receive data input (normal/IrDA)
	TxD0/IrTxD	Output	Channel 0 transmit data output (normal/IrDA)
1	SCK1	I/O	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

13.3 Register Descriptions

The SCI has the following registers. 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 their bit functions partially differ.

- Receive shift register_0 (RSR_0)
- Transmit shift register_0 (TSR_0)
- Receive data register_0 (RDR_0)
- Transmit data register_0 (TDR_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)
- IrDA control register_0 (IrCR_0)
- Receive shift register_1 (RSR_1)
- Transmit shift register_1 (TSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_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)
- Receive shift register_2 (RSR_2)
- Transmit shift register_2 (TSR_2)
- Receive data register_2 (RDR_2)
- Transmit data register_2 (TDR_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)
- Serial extension mode register (SEMR)

13.3.1 Receive Shift Register (RSR)

RSR is a shift register used to receive serial data that is 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. Since RSR and RDR function as a double buffer in this way, enables continuous receive operations to be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU.

13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR during serial transmission, 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 for only once after confirming that the TDRE bit in SSR is set to 1.

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 starting. 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 on-chip baud rate generator clock source.

Some bit functions of SMR differ in 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.
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.
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 regardless of the STOP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit character.

Bit	Bit Name	Initial Value	R/W	Description
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.
1	CKS1	0	R/W	Clock Select 1 and 0:
0	CKS0	0	R/W	These bits select the clock source for the on-chip 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 relation between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).

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 before 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 select 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. S stands for the value of S in BRR (see section 13.3.9, Bit Rate Register (BRR)).

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 1 and 0: These bits select the clock source for the on-chip 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 relation between the bit rate register setting and the baud rate, see section 13.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 13.3.9, Bit Rate Register (BRR)).
0	CKS0	0	R/W	

13.3.6 Serial Control Register (SCR)

SCR performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer/receive clock source. For details on interrupt requests, refer to section 13.9, Interrupts Sources. Some bit functions of SCR differ in 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, TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit s set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable: When this bit is set to 1, reception is enabled.

Bit	Bit Name	Initial Value	R/W	Description
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.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is enabled.
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)

Note: 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.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
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.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in Smart Card interface mode.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0		Enables or disables 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

Note: 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 in 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 Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and transfers data to TDR
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 transferred data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • When the next serial reception is completed while RDRF = 1 [Clearing condition] • When 0 is written to ORER after reading ORER = 1
4	FER	0	R/(W)*	Framing Error [Setting condition] • When the stop bit is 0 [Clearing condition] • When 0 is written to FER after reading FER = 1 In 2-stop-bit mode, only the first stop bit is checked.
3	PER	0	R/(W)*	Parity Error [Setting condition] • When a parity error is detected during reception [Clearing condition] • When 0 is written to PER after reading PER = 1
2	TEND	1	R	Transmit End [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 and writes data to TDR

Bit	Bit Name	Initial Value	R/W	Description
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 sets the multiprocessor bit to be added to the transmit data.

Note: * Only 0 can be written, 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 [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and transfers data to TDR
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 transferred data from RDR The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • When the next serial reception is completed while RDRF = 1 [Clearing condition] • When 0 is written to ORER after reading ORER = 1
4	ERS	0	R/(W)*	Error Signal Status [Setting condition] • When the low level of the error signal is sampled [Clearing conditions] • When 0 is written to ERS after reading ERS = 1
3	PER	0	R/(W)*	Parity Error [Setting condition] • When a parity error is detected during reception [Clearing condition] • When 0 is written to PER after reading PER = 1

Bit	Bit Name	Initial Value	R/W	Description
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 • If the ERS bit is 0 and the TDRE bit is 1 after the specified interval after transmission of 1-byte data Timing to set this bit differs according to the register settings. GM = 0, BLK = 0: 2.5 etu after transmission GM = 0, BLK = 1: 1.5 etu after transmission GM = 1, BLK = 0: 1.0 etu after transmission GM = 1, BLK = 1: 1.0 etu after transmission [Clearing conditions] • When 0 is written to TEND after reading TEND = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
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 clear the flag.

13.3.8 Smart Card Mode Register (SCMR)

SCMR 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.
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.
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

Mode	Bit Rate	Error
Asynchronous Mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$
Clocked Synchronous Mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	
Smart Card Interface Mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} - 1 \right\} \times 100$

Note: B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \le N \le 255$)
ϕ: Operating frequency (MHz)
n and S: Determined by the SMR settings shown in the following tables.

SMR Setting

CKS1	CKS0	n
0	0	0
0	1	1
1	0	2
1	1	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) (1)

	Operating Frequency ϕ (MHz)											
	2			2.097152			2.4576			3		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	—	—	—	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	–2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	0	1	0.00	—	—	—

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency ϕ (MHz)											
	3.6864			4			4.9152			5		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	1.73

	Operating Frequency ϕ (MHz)											
	6			6.144			7.3728			8		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	Operating Frequency ϕ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	Operating Frequency ϕ (MHz)											
	14			14.7456			16			17.2032		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	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	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (4)

	Operating Frequency ϕ (MHz)											
	18			19.6608			20			25		
Bit Rate (bit/s)	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	3	64	0.16	3	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	2	64	0.16	2	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	1	64	0.16	1	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

	Operating Frequency ϕ (MHz)					
	30			33		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)
110	3	132	0.13	3	145	0.33
150	3	97	–0.35	3	106	0.39
300	2	194	0.16	2	214	–0.07
600	2	97	–0.35	2	106	0.39
1200	1	194	0.16	1	214	–0.07
2400	1	97	–0.35	1	106	0.39
4800	0	194	0.16	0	214	–0.07
9600	0	97	–0.35	0	106	0.39
19200	0	48	–0.35	0	53	–0.54
31250	0	29	0	0	32	0
38400	0	23	1.73	0	26	–0.54

Table 13.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
2	62500	0	0	10	312500	0	0
2.097152	65536	0	0	12	375000	0	0
2.4576	76800	0	0	12.288	384000	0	0
3	93750	0	0	14	437500	0	0
3.6864	115200	0	0	14.7456	460800	0	0
4	125000	0	0	16	500000	0	0
4.9152	153600	0	0	17.2032	537600	0	0
5	156250	0	0	18	562500	0	0
6	187500	0	0	19.6608	614400	0	0
6.144	192000	0	0	20	625000	0	0
7.3728	230400	0	0	25	781250	0	0
8	250000	0	0	30	937500	0	0
9.8304	307200	0	0	33	1031250	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)	ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.5000	31250	10	2.5000	156250
2.097152	0.5243	32768	12	3.0000	187500
2.4576	0.6144	38400	12.288	3.0720	192000
3	0.7500	46875	14	3.5000	218750
3.6864	0.9216	57600	14.7456	3.6864	230400
4	1.0000	62500	16	4.0000	250000
4.9152	1.2288	76800	17.2032	4.3008	268800
5	1.2500	78125	18	4.5000	281250
6	1.5000	93750	19.6608	4.9152	307200
6.144	1.5360	96000	20	5.0000	312500
7.3728	1.8432	115200	25	6.2500	390625
8	2.0000	125000	30	7.5000	468750
9.8304	2.4576	153600	33	8.2500	515625

Table 13.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate	Operating Frequency φ (MHz)													
	2		4		8		10		16		20		25	
(bit/s)	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—										
250	2	124	2	249	3	124	—	—	3	249				
500	1	249	2	124	2	249	—	—	3	124	—	—		
1 k	1	124	1	249	2	124	—	—	2	249	—	—	3	97
2.5 k	0	199	1	99	1	199	1	249	2	99	2	124	2	155
5 k	0	99	0	199	1	99	1	124	1	199	1	249	2	77
10 k	0	49	0	99	0	199	0	249	1	99	1	124	1	155
25 k	0	19	0	39	0	79	0	99	0	159	0	199	0	249
50 k	0	9	0	19	0	39	0	49	0	79	0	99	0	124
100 k	0	4	0	9	0	19	0	24	0	39	0	49	0	62
250 k	0	1	0	3	0	7	0	9	0	15	0	19	0	24
500 k	0	0*	0	1	0	3	0	4	0	7	0	9	—	—
1 M			0	0*	0	1			0	3	0	4	—	—
2.5 M							0	0*			0	1	—	—
5 M											0	0*	—	—

Bit Rate (bit/s)	Operating Frequency ϕ (MHz) 30 n	30 N	33 n	33 N
110				
250				
500	3	233		
1 k	3	116	3	128
2.5 k	2	187	2	205
5 k	2	93	2	102
10 k	1	187	1	205
25 k	1	74	1	82
50 k	0	149	0	164
100 k	0	74	0	82
250 k	0	29	0	32
500 k	0	14	—	—
1 M	—	—	—	—
2.5 M	0	2	—	—
5 M	—	—	—	—

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 (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.3333	333333.3	16	2.6667	2666666.7
4	0.6667	666666.7	18	3.0000	3000000.0
6	1.0000	1000000.0	20	3.3333	3333333.3
8	1.3333	1333333.3	25	4.1667	4166666.7
10	1.6667	1666666.7	30	5.0000	5000000.0
12	2.0000	2000000.0	33	5.5000	5500000.0
14	2.3333	2333333.3			

Table 13.8 Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode) (when n = 0 and S = 372)

	Operating Frequency ϕ (MHz)											
	7.1424			10.00			10.7136			13.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	1	1	0.00	0	1	30	0	1	25	0	1	8.99

	Operating Frequency ϕ (MHz)											
	14.2848			16.00			18.00			20.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6.60

	Operating Frequency ϕ (MHz)								
	25.00			30.00			33.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	12.49	0	3	5.01	0	4	7.59

Table 13.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
18.00	24194	0	0
20.00	26882	0	0
25.00	33602	0	0
30.00	40323	0	0
33.00	44355	0	0

13.3.10 IrDA Control Register (IrCR)

IrCR selects the function of SCI_0.

Bit	Bit Name	Initial Value	R/W	Description
7	IrE	0	R/W	IrDA Enable Specifies normal SCI mode or IrDA mode for SCI_0 input/output. 0: Pins TxD0/IrTxD and RxD0/IrRxD function as TxD0 and RxD0 1: Pins TxD0/IrTxD and RxD0/IrRxD function as IrTxD and IrRxD
6	IrCKS2	0	R/W	IrDA Clock Select 2 to 0 Specifies the high pulse width in IrTxD output pulse encoding when the IrDA function is enabled. 000: Pulse width = B × 3/16 (3/16 of bit rate) 001: Pulse width = ϕ/2 010: Pulse width = ϕ/4 011: Pulse width = ϕ/8 100: Pulse width = ϕ/16 101: Pulse width = ϕ/32 110: Pulse width = ϕ/64 111: Pulse width = ϕ/128
5	IrCKS1	0	R/W	
4	IrCKS0	0	R/W	
3 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

13.3.11 Serial Extension Mode Register (SEMR)

SEMR selects the clock source in asynchronous mode. The basic clock can be automatically set by selecting the average transfer rate. SEMR is supported only in SCI_2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, an undefined value will be returned and cannot be modified.
3	ABCS	0	R/W	Asynchronous basic clock selection (valid only in asynchronous mode) Selects the basic clock for 1-bit period in asynchronous mode. 0: Operates on a basic clock with a frequency of 16 times the transfer rate. 1: Operates on a basic clock with a frequency of 8 times the transfer rate.

Bit	Bit Name	Initial Value	R/W	Description
2	ACS2	0	R/W	Asynchronous clock source selection (valid when CKS1 = 1 in asynchronous mode)
1	ACS1	0	R/W	
0	ACS0	0	R/W	Selects the clock source for the average transfer rate. The basic clock can be automatically set by selecting the average transfer rate in spite of the value of ABCS. 000: External clock input 001: Selects 115.152 kbps which is the average transfer rate dedicated for ϕ= 10.667 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 010: Selects 460.606 kbps which is the average transfer rate dedicated for ϕ= 10.667 MHz. (Operates on a basic clock with a frequency of 8 times the transfer rate.) 011: Selects 720 kbps which is the average transfer rate dedicated for ϕ = 32 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 100: Reserved 101: Selects 115.196 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.) 110: Selects 460.784 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.) 111: Selects 720 kbps which is the average transfer rate dedicated for ϕ = 16 MHz (Operates on a basic clock with a frequency of 8 times the transfer rate.) Note that the average transfer rate does not correspond to the frequency other than 10.667, 16, or 32 MHz.

13.4 Operation in Asynchronous Mode

Figure 13.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by transfer 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. In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the communication 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.

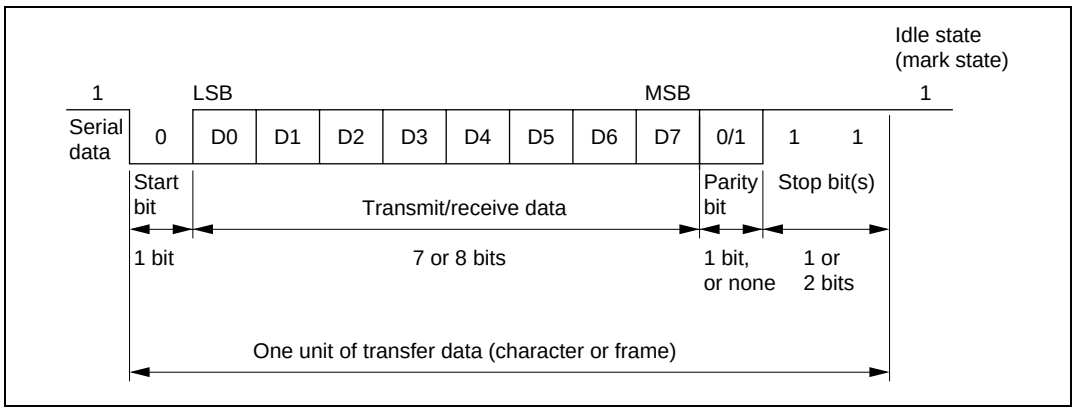


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

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 bit 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 at the middle of each bit by sampling the data at the rising edge of the 8th pulse of the basic clock as shown in figure 13.3. Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left\{(0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F)\right\} \times 100\ [\%] \quad \text{... Formula (1)}$$

Where M: Reception Margin

N: Ratio of bit rate to clock (N = 16)

D: Clock duty cycle (D = 0.5 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin is given by formula below.

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

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

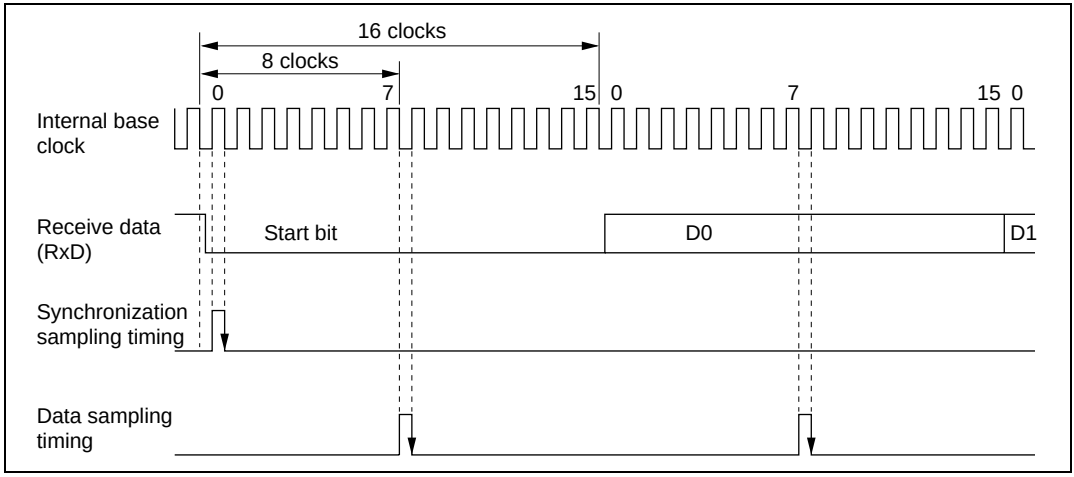


Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode

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 CKE1 and CKE0 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. 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.

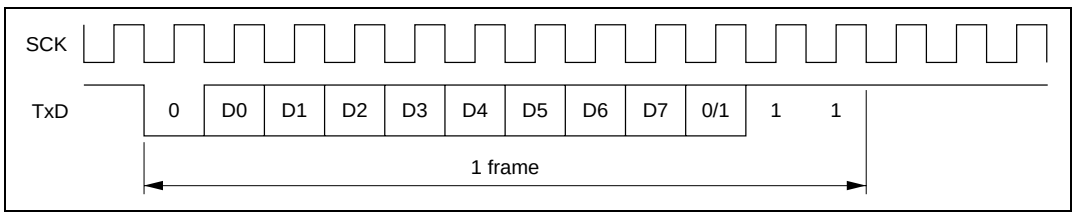


Figure 13.4 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

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 shown in figure 13.5. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. 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.

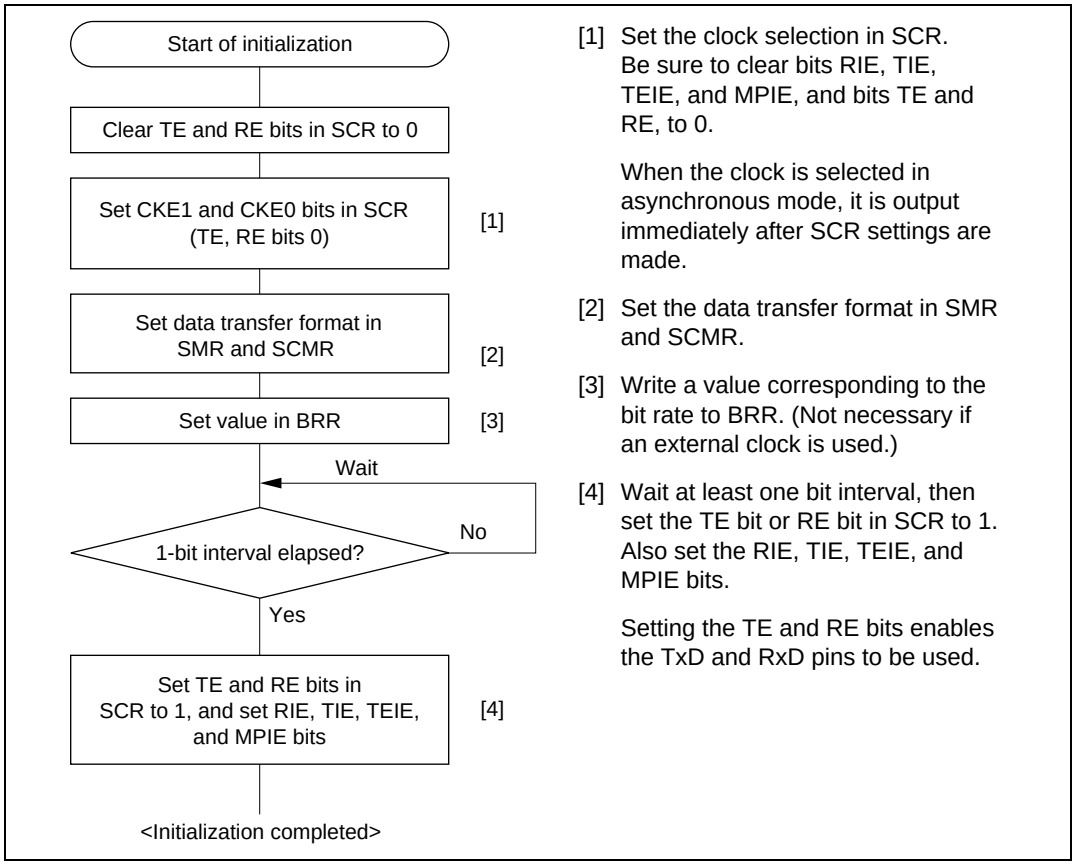


Figure 13.5 Sample SCI Initialization Flowchart

13.4.5 Data Transmission (Asynchronous Mode)

Figure 13.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if is cleared to 0, 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. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
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 13.7 shows a sample flowchart for transmission in asynchronous mode.

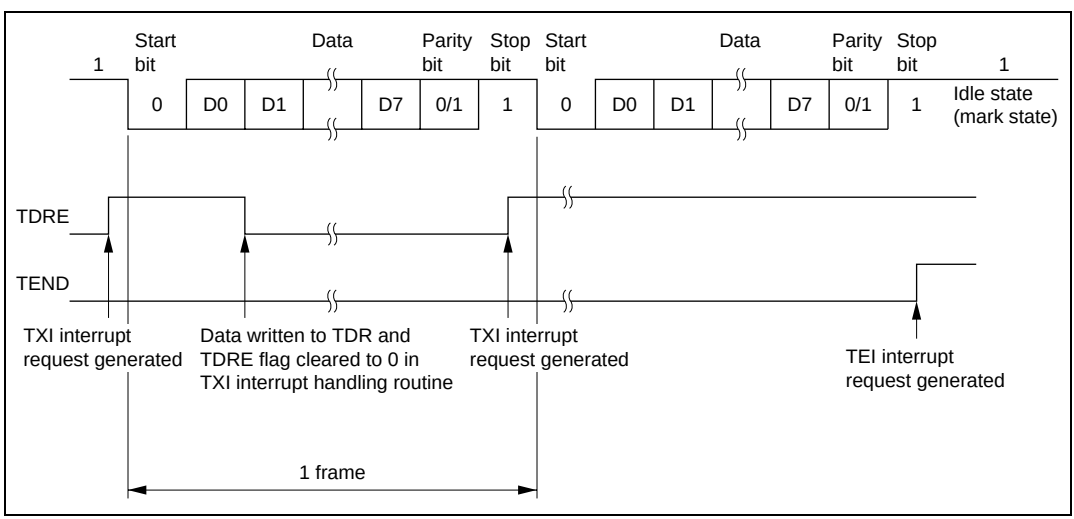
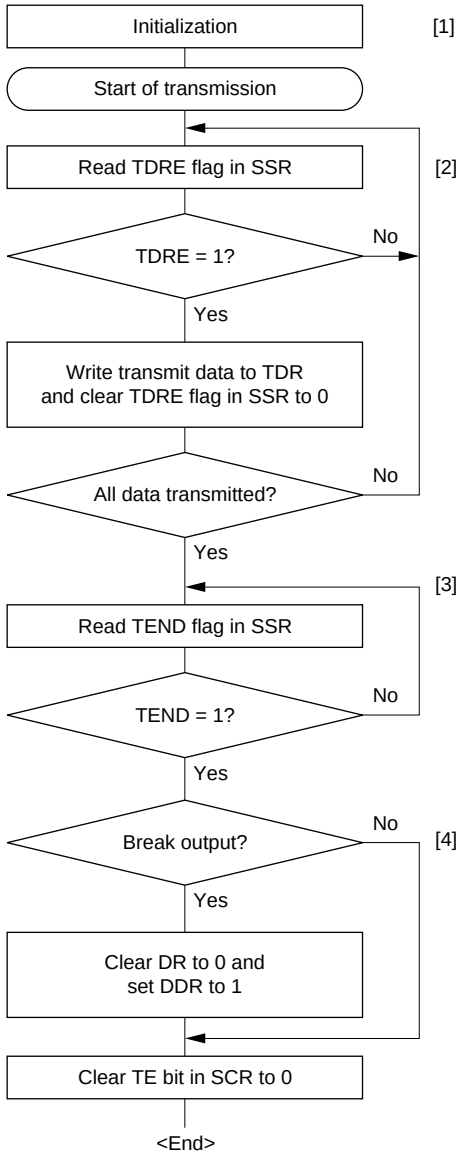


Figure 13.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[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 and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, 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 DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 13.7 Sample Serial Transmission Flowchart

13.4.6 Serial Data Reception (Asynchronous Mode)

Figure 13.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, 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 (when the stop bit is 0) is detected, 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 finishes 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. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

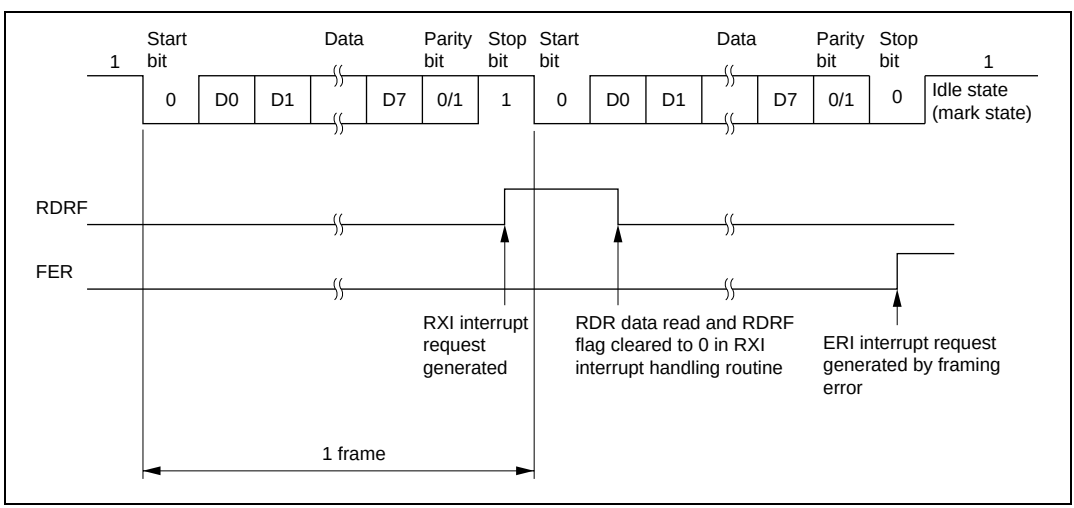


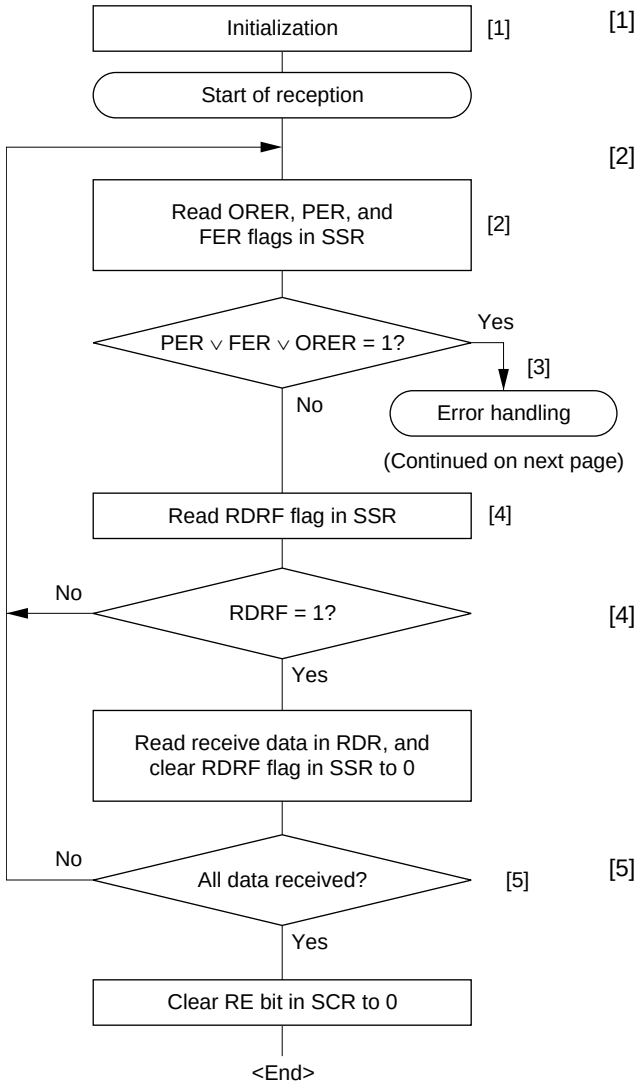
Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

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 its state before data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling 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 the DTC is activated by an RXI interrupt and the RDR value is read.

Figure 13.9 Sample Serial Reception Data Flowchart (1)

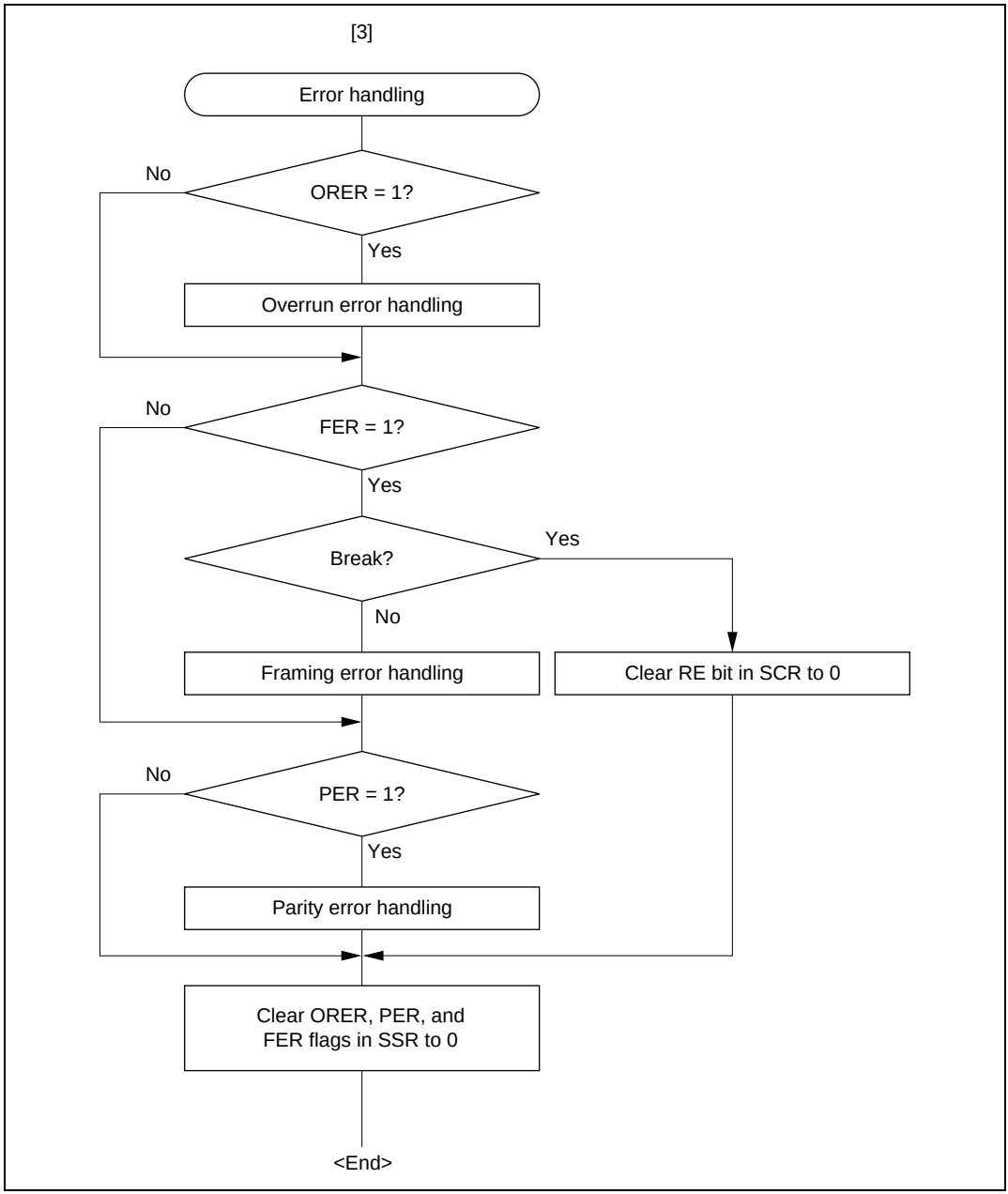


Figure 13.9 Sample Serial Reception Data Flowchart (2)

13.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle to the specified receiving station. 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, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 13.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends communication data with a 1 multiprocessor bit added to the ID code of the receiving station. 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 ID does 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 receive character with a 1 multiprocessor bit, the MPBR 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 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.

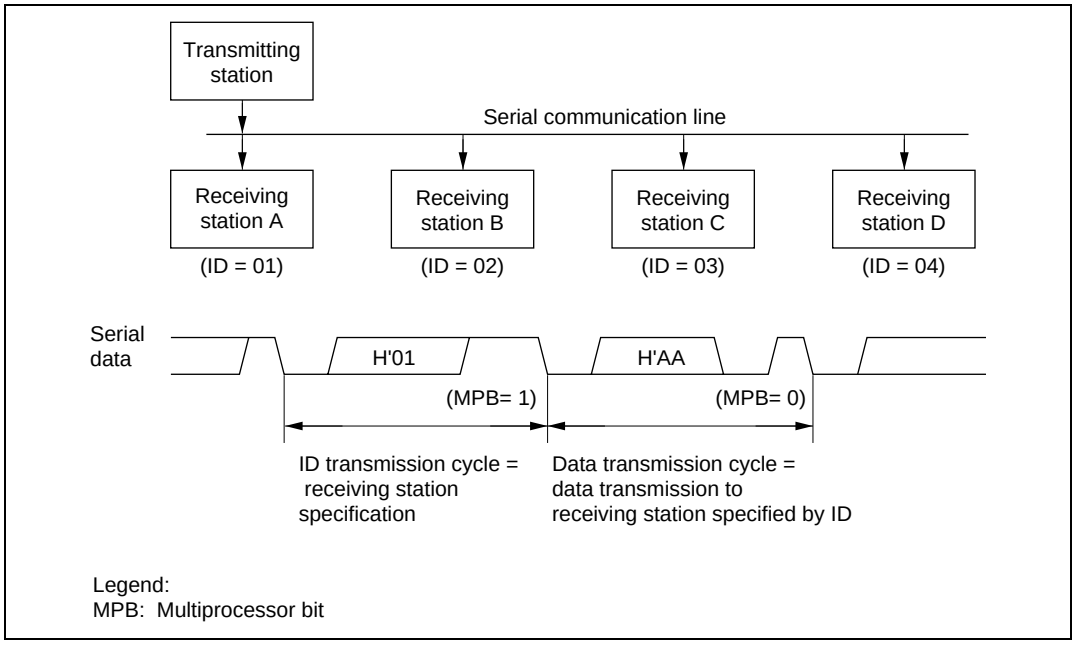
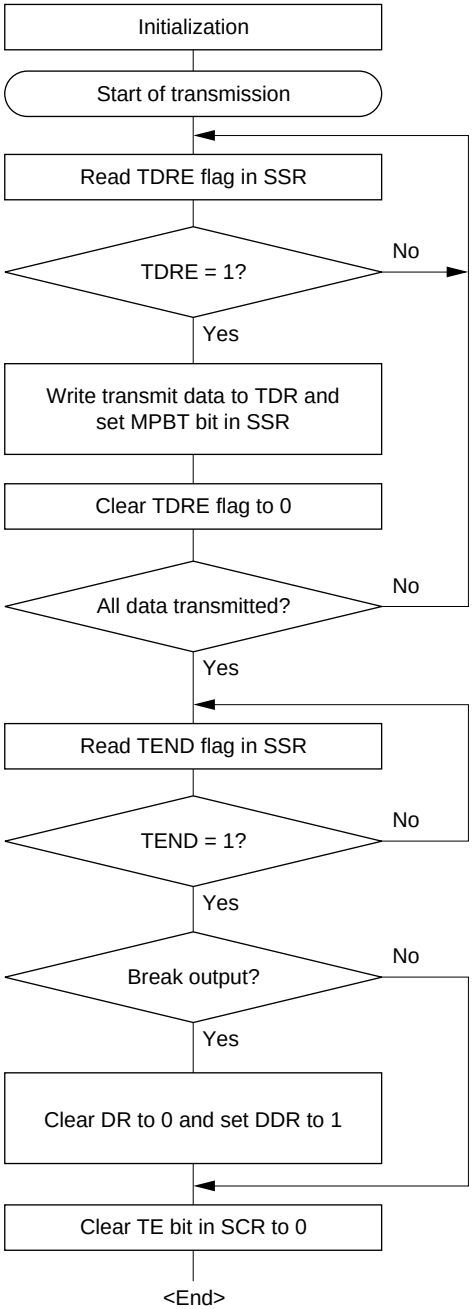


Figure 13.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

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 DDR to 1, clear DR to 0, 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 received. 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.

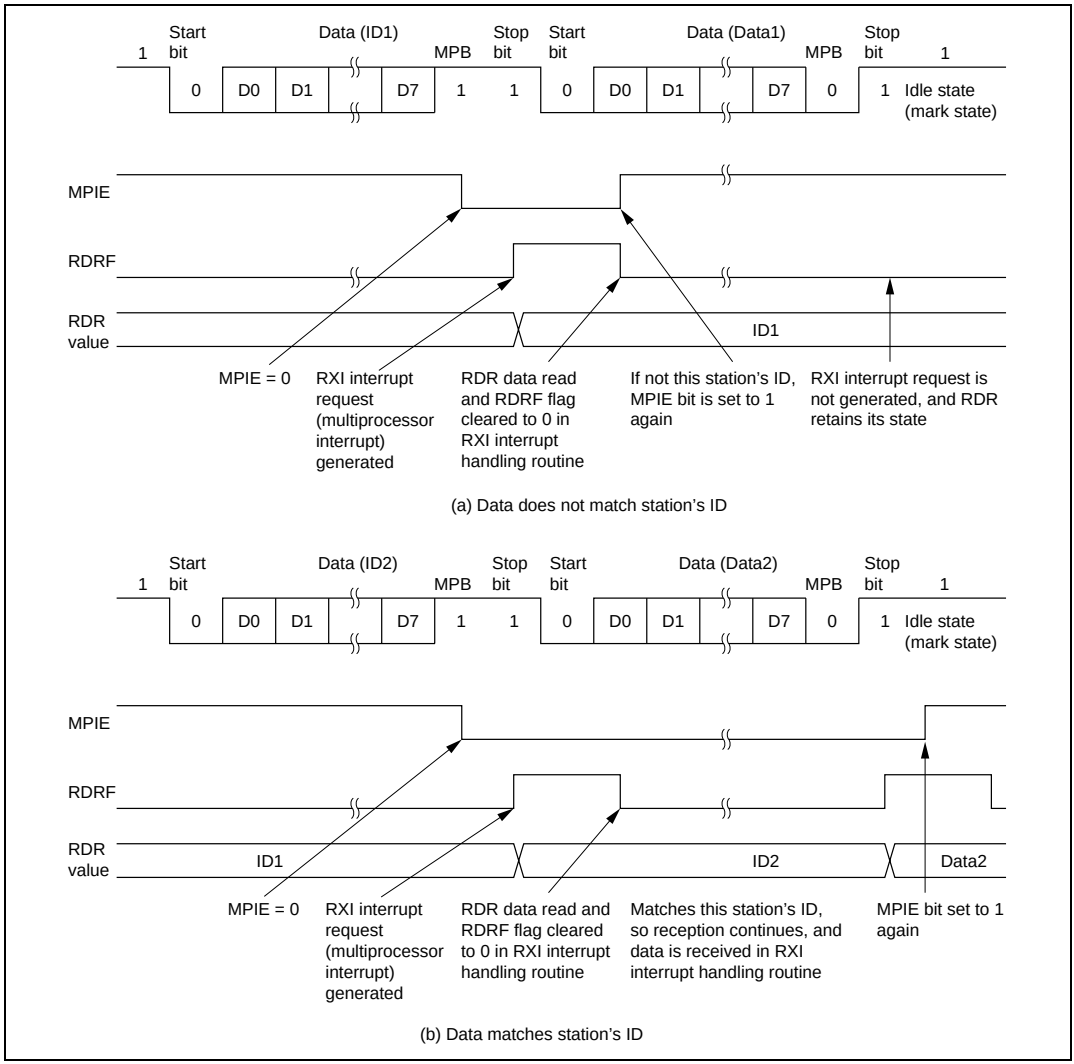
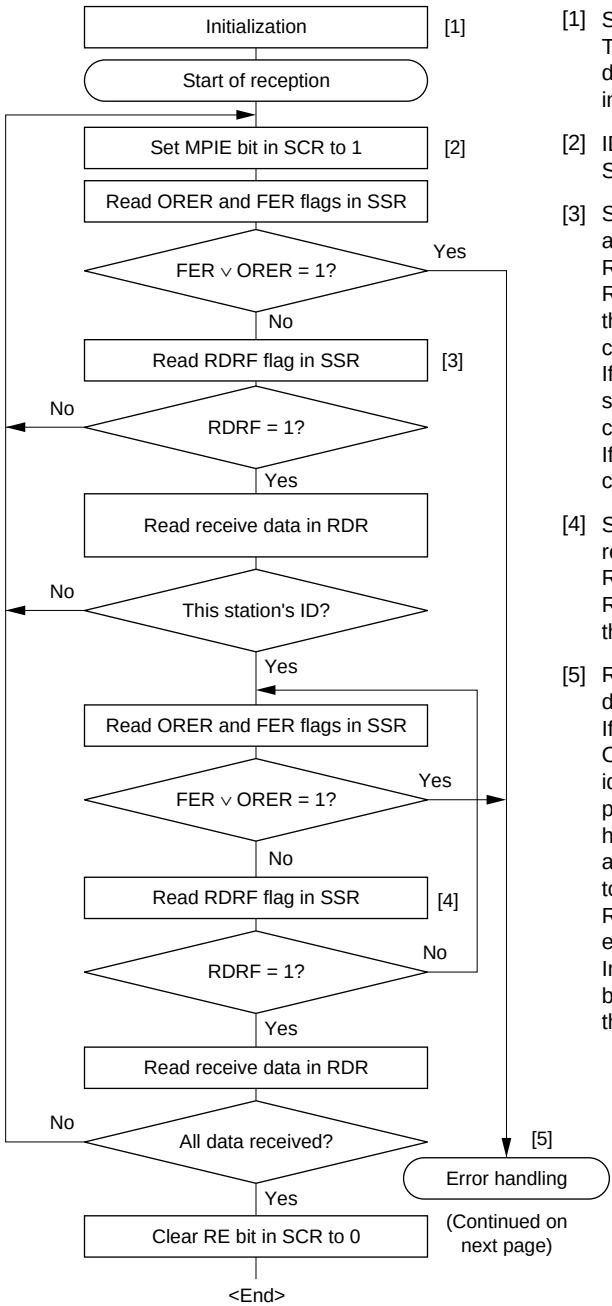


Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error handling and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error handling, ensure that the ORER and FER flags are both cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

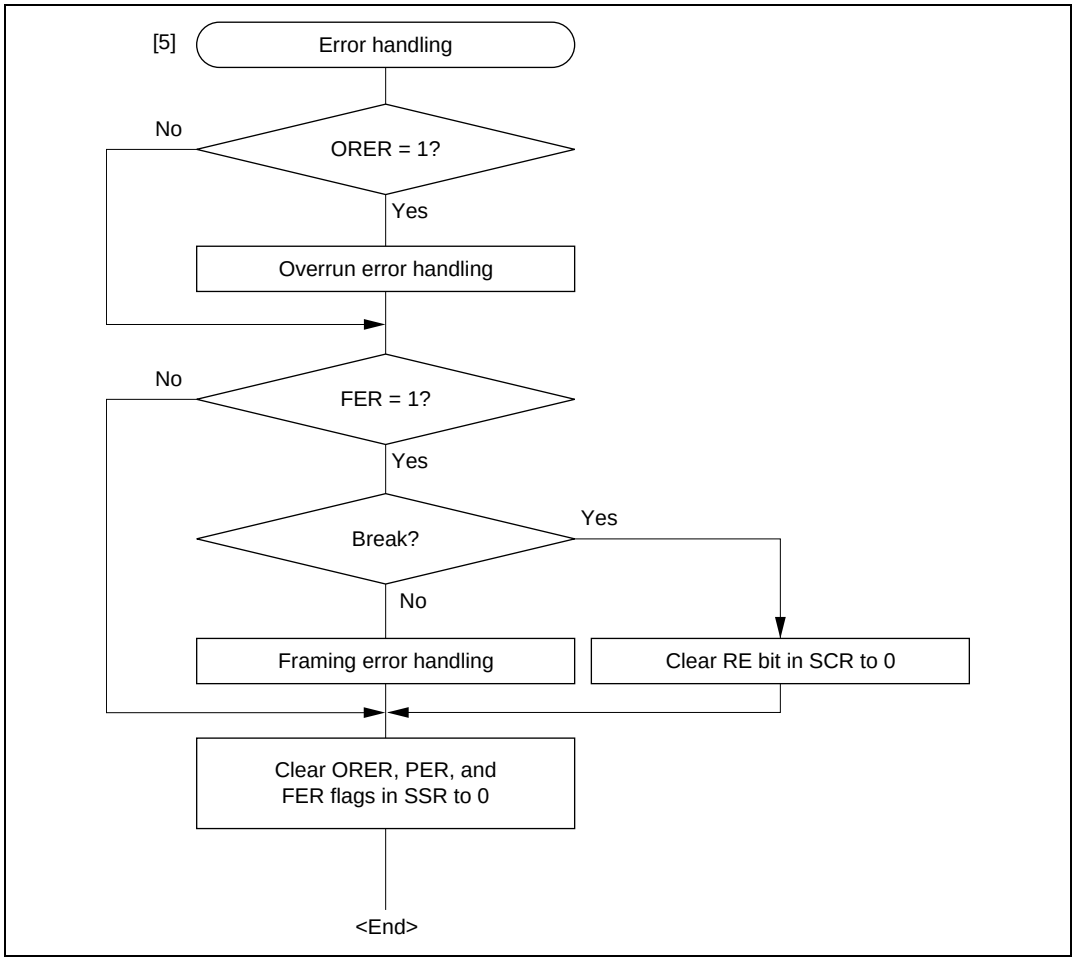


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)

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 in synchronization with clock pulses. One character of communication data consists of 8-bit data. 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 synchronization 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 by use of a common clock. 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.

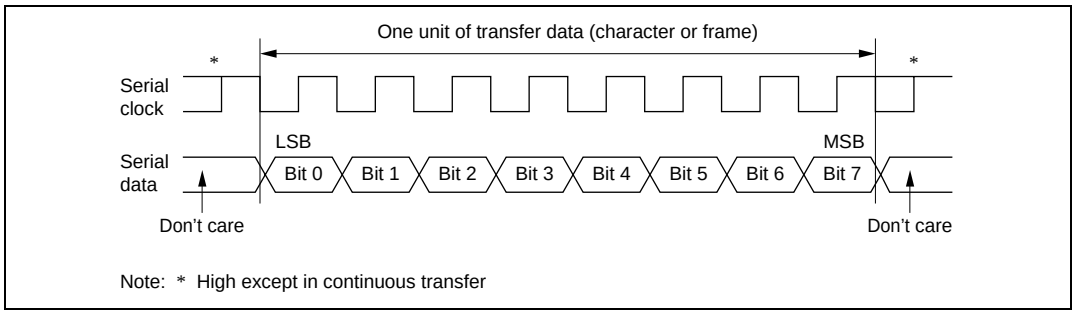


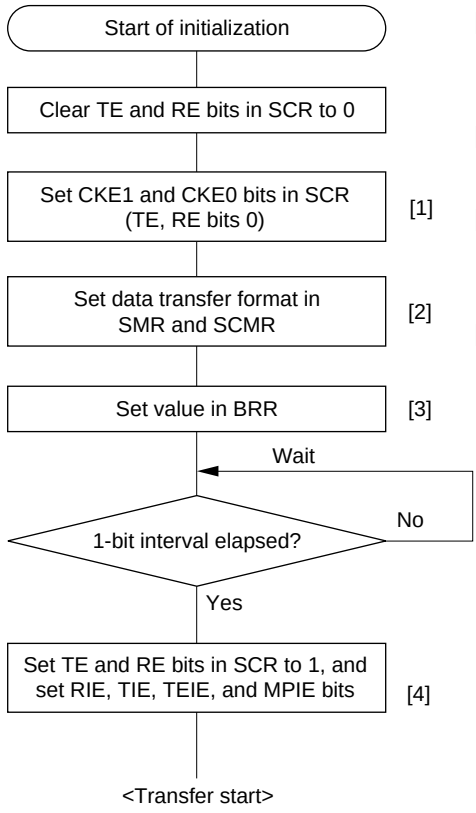
Figure 13.14 Data Format in Clocked Synchronous Communication (For LSB-First)

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 CKE1 and CKE0 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, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 13.15. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. 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 and RE bits in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enable 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 is 0, 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 TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
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.
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 TxD pin 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 to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

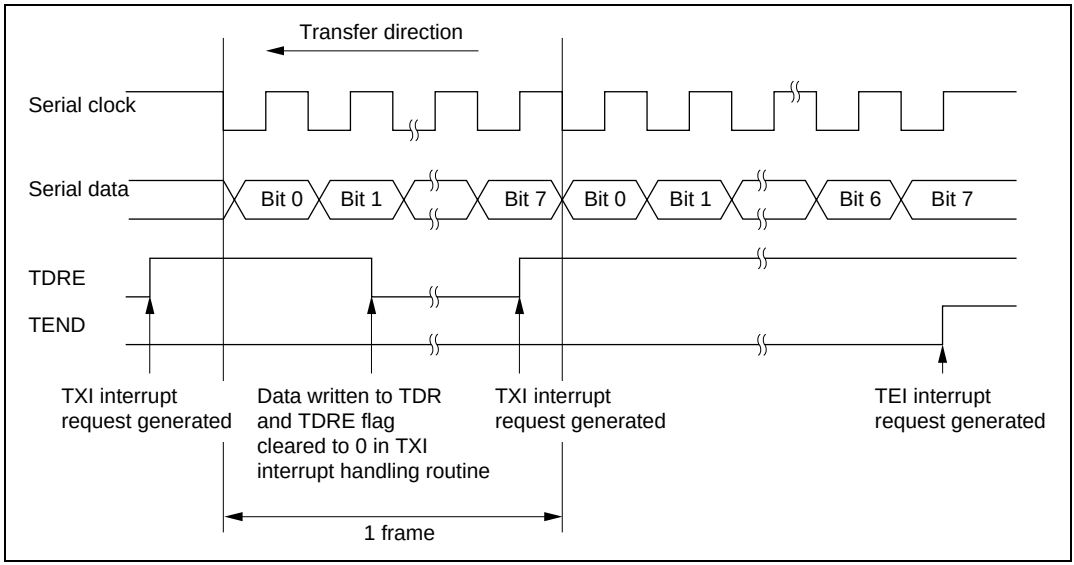
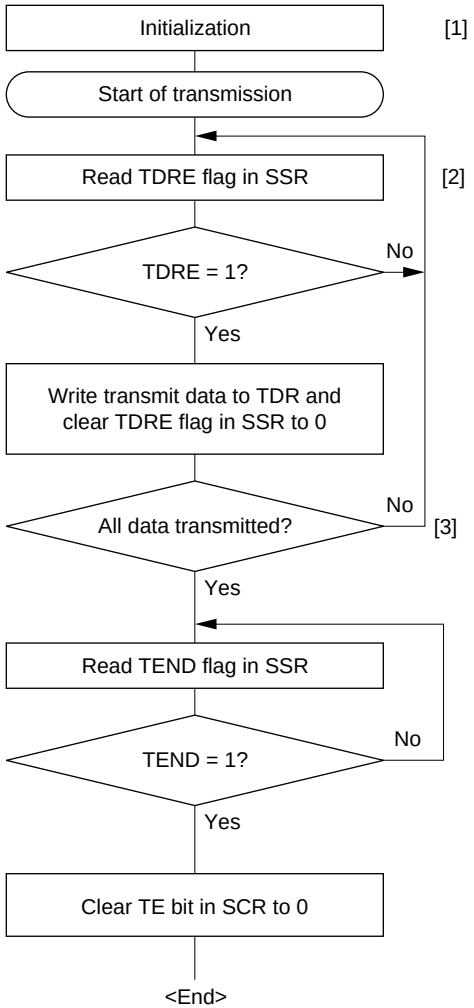


Figure 13.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[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 in synchronization with a synchronization clock input or output, starts receiving data, and stores the received data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, 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 reception finishes 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. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

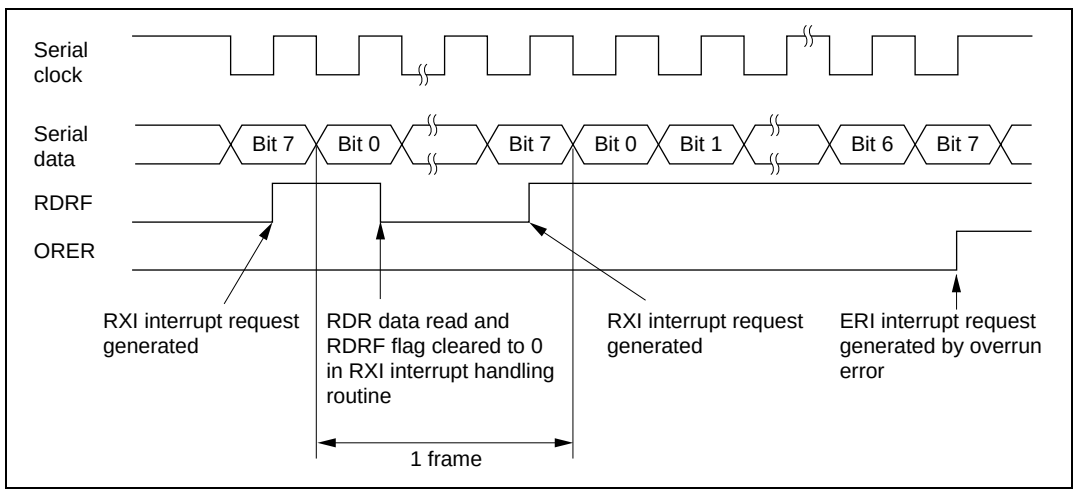
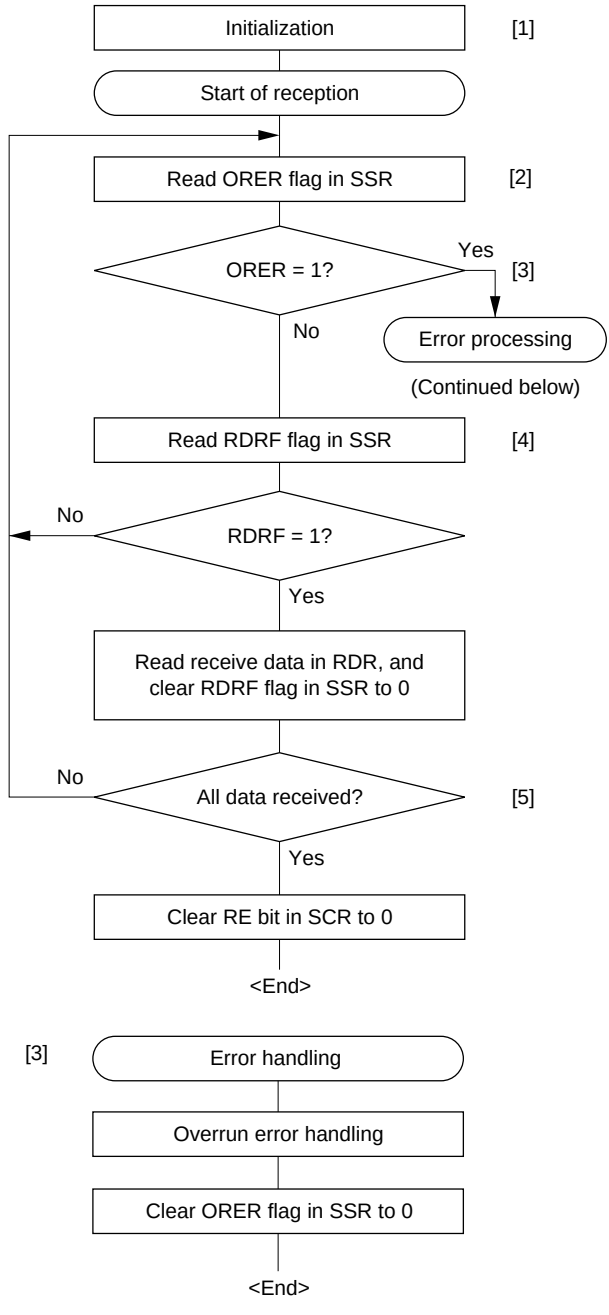


Figure 13.18 Example of SCI Operation in Reception

Transfer 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 flowchart for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

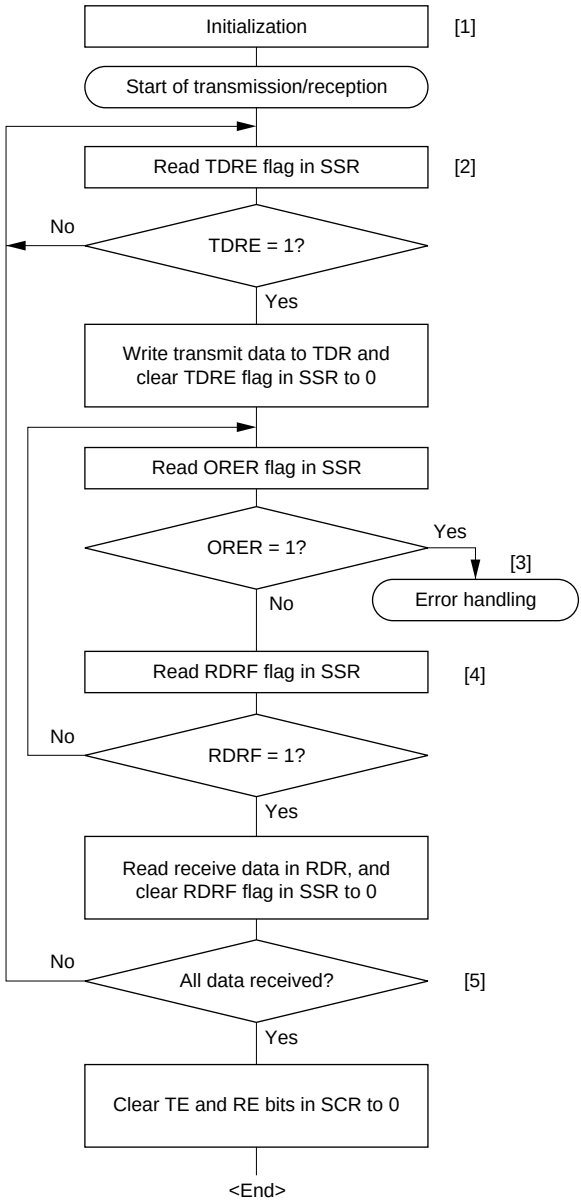
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. 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 after the SCI is initialized. 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 handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, 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 MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) 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 and RE bits to 0, then set both these bits to 1 simultaneously.

Figure 13.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

13.7 Operation in Smart Card Interface Mode

The SCI supports an IC card (Smart Card) interface conforming 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 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, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected with the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply 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 used by an IC card, the SCK pin output is input to the CLK pin of the IC card. This LSI port output is used as the reset signal.

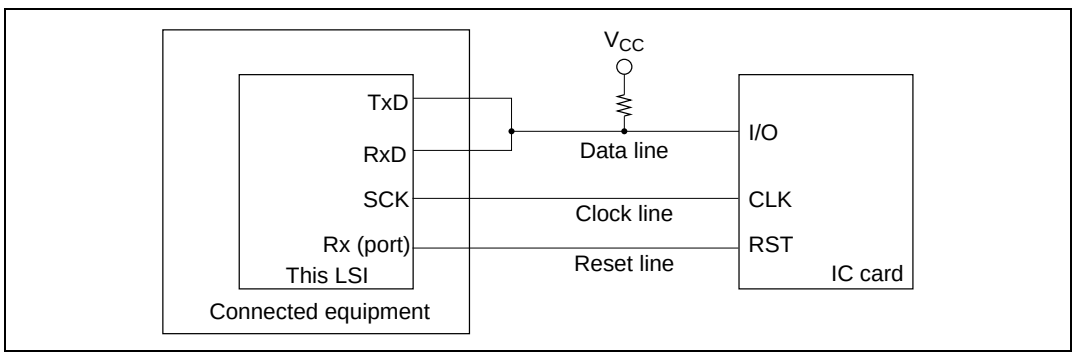


Figure 13.21 Schematic Diagram of Smart Card Interface Pin Connections

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 the elapse of 2 etu or longer.

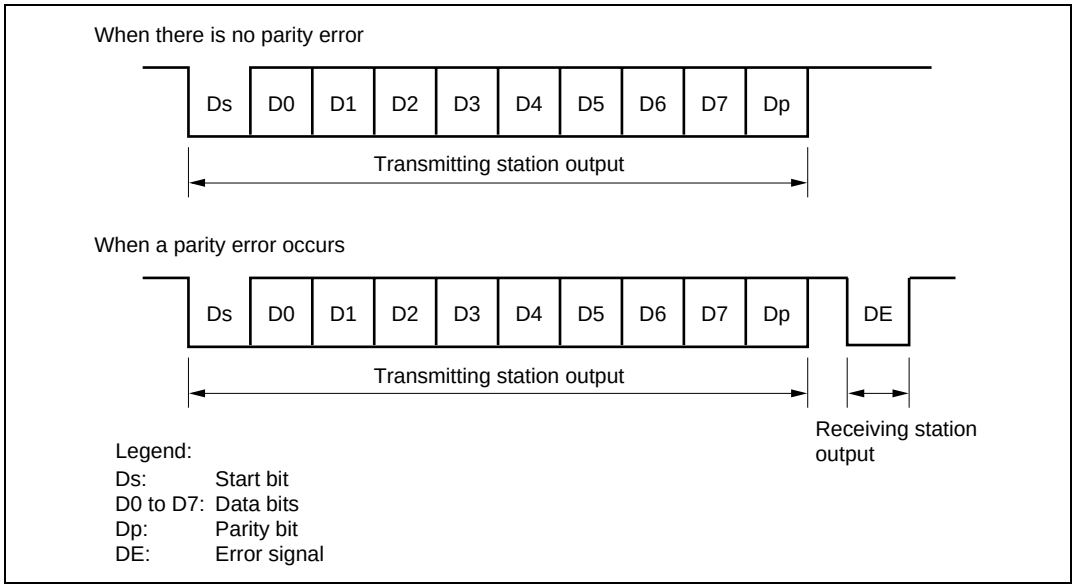


Figure 13.22 Normal Smart Card Interface Data Format

Data transfer with the types of IC cards (direct convention and inverse convention) are performed as described in the following.

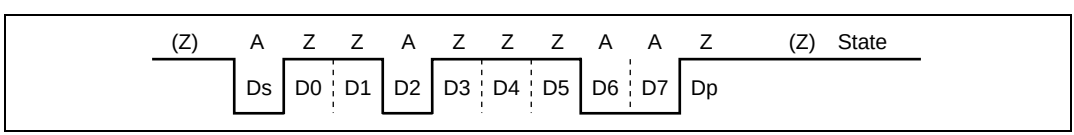


Figure 13.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

As in the above sample start character, with the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. 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 the Smart Card regulations, clear the O/$\overline{E}$ bit in SMR to 0 to select even parity mode.

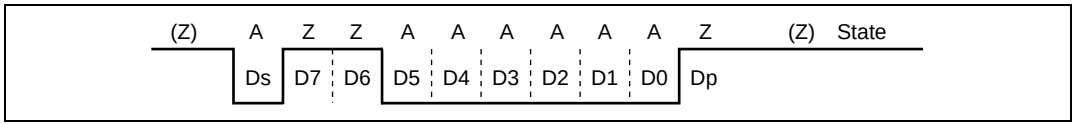


Figure 13.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

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 above is H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. According to the 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 D7 to D0. 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 normal Smart Card interface, 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

Only the internal clock generated by the on-chip baud rate generator is used as transmit/receive clock in Smart Card interface. In Smart Card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the bit rate (fixed at 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 cycle (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.

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

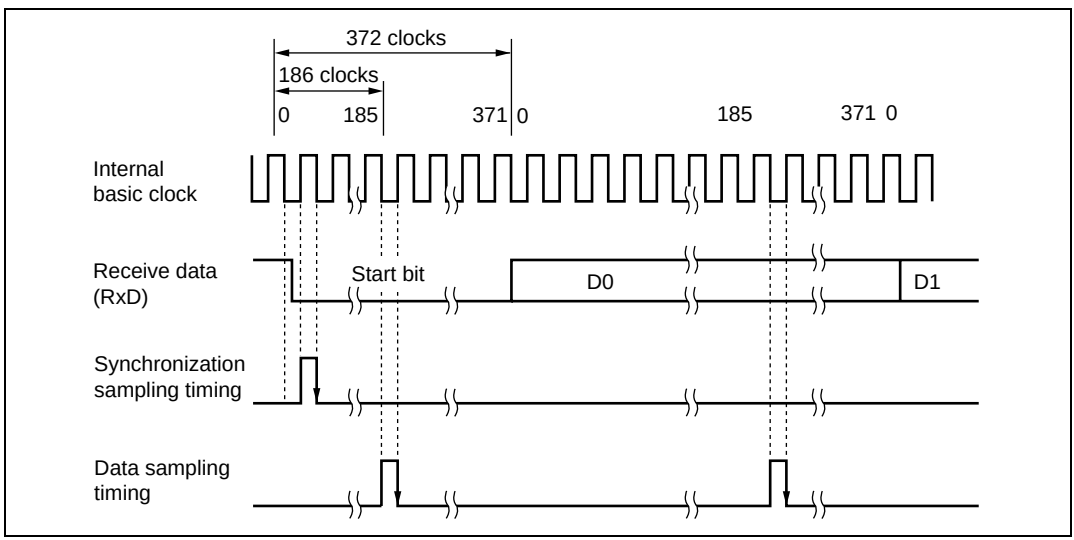


Figure 13.25 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Bit Rate)

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/$\overline{E}$, BCP1, BCP0, CKS1, and CKS0 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 clear RE to 0 and set TE to 1. Whether SCI has finished reception can be checked with the RDRF, PER, or ORER flag. To switch from transmit mode to receive mode, after checking that the SCI has finished transmission, initialize the SCI, and clear TE to 0 and set RE to 1. Whether SCI has finished transmission can be checked with the TEND flag.

13.7.6 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 sampled from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The ERS bit in SSR should be cleared to 0 before the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame for which an error signal 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 set 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. The 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 also set to 1 at the same time as the TEND flag in SSR, 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 transfer is performed 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 7, Data Transfer Controller (DTC).

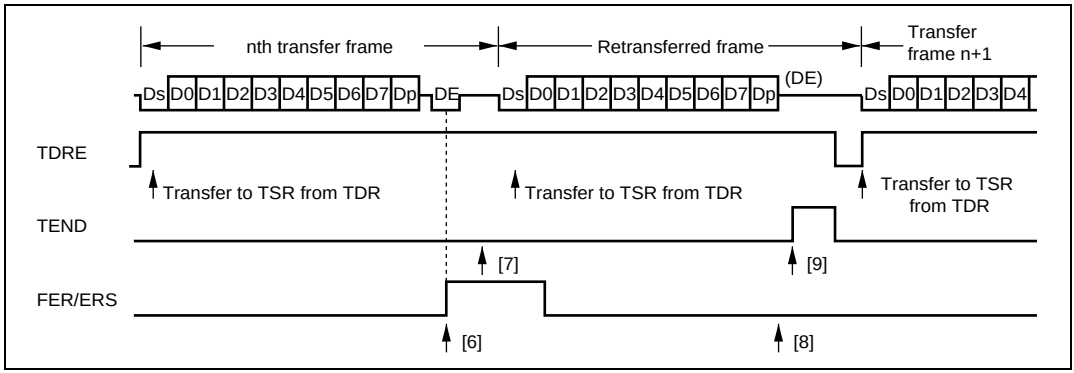


Figure 13.26 Retransfer Operation in SCI Transmit Mode

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag generation timing is shown in figure 13.27.

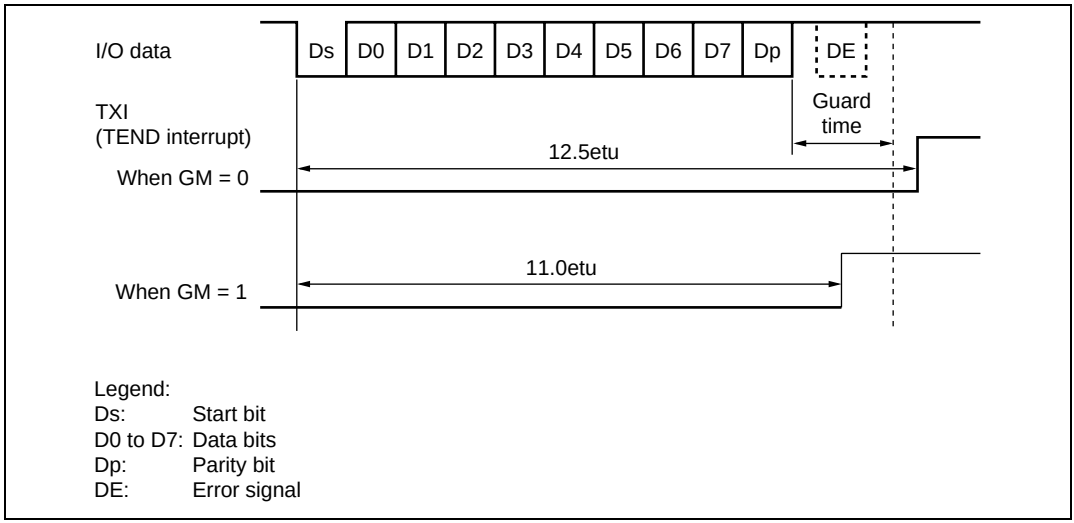


Figure 13.27 TEND Flag Generation Timing in Transmission Operation

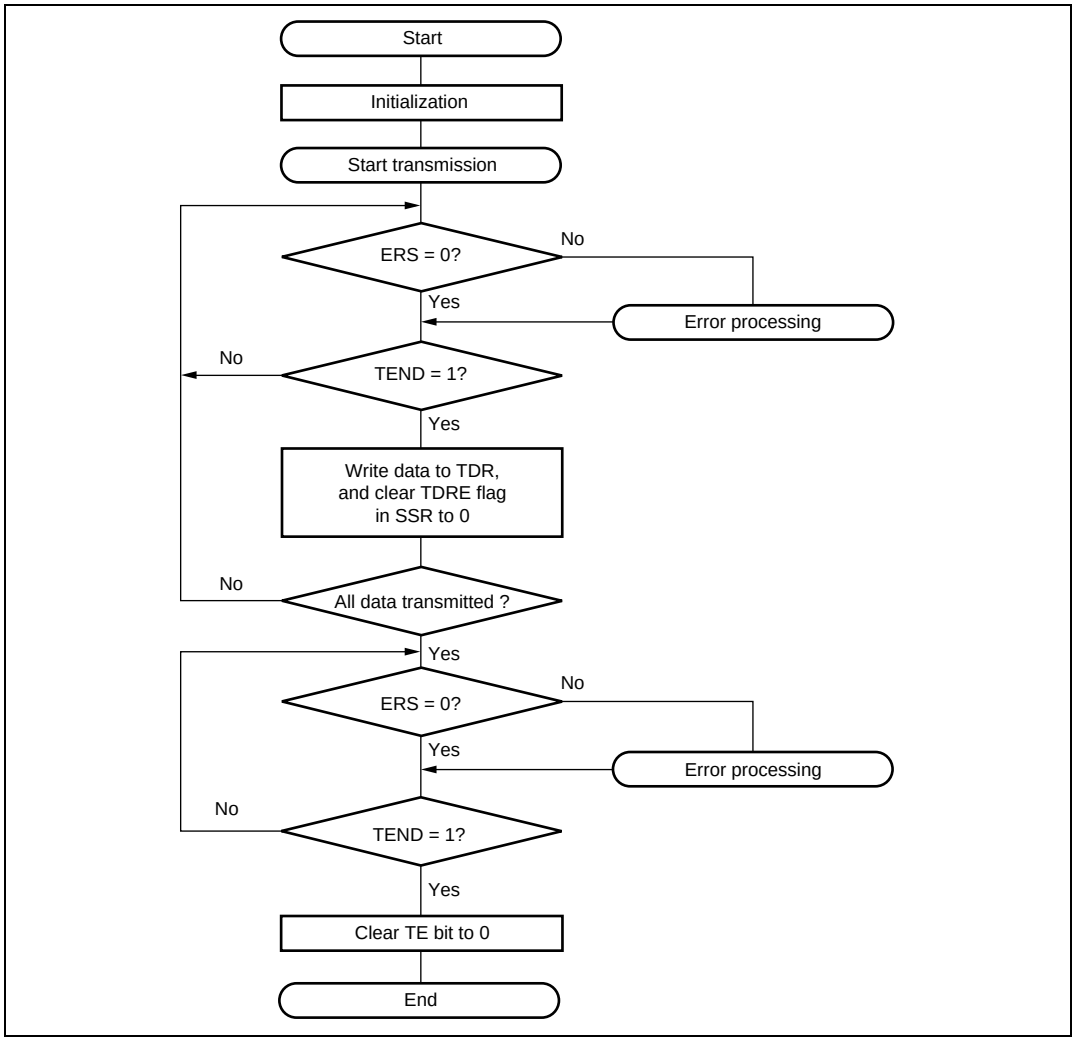


Figure 13.28 Example of Transmission Processing Flow

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 cleared to 0 before 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 set at this time, an RXI interrupt request is generated.

Figure 13.30 shows a flowchart for reception. The 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 transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed 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, and 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.

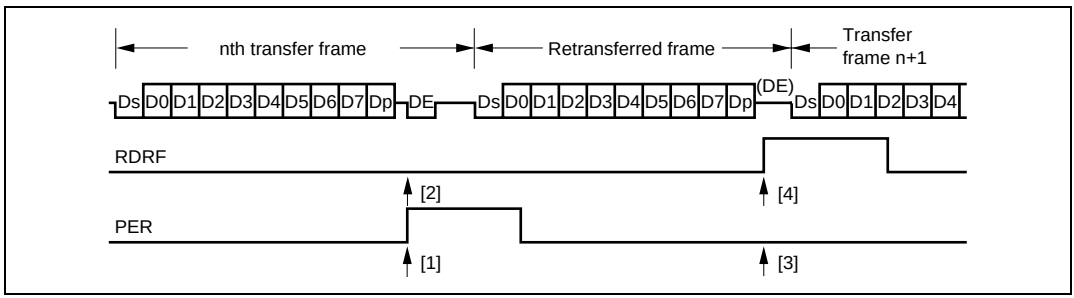


Figure 13.29 Retransfer Operation in SCI Receive Mode

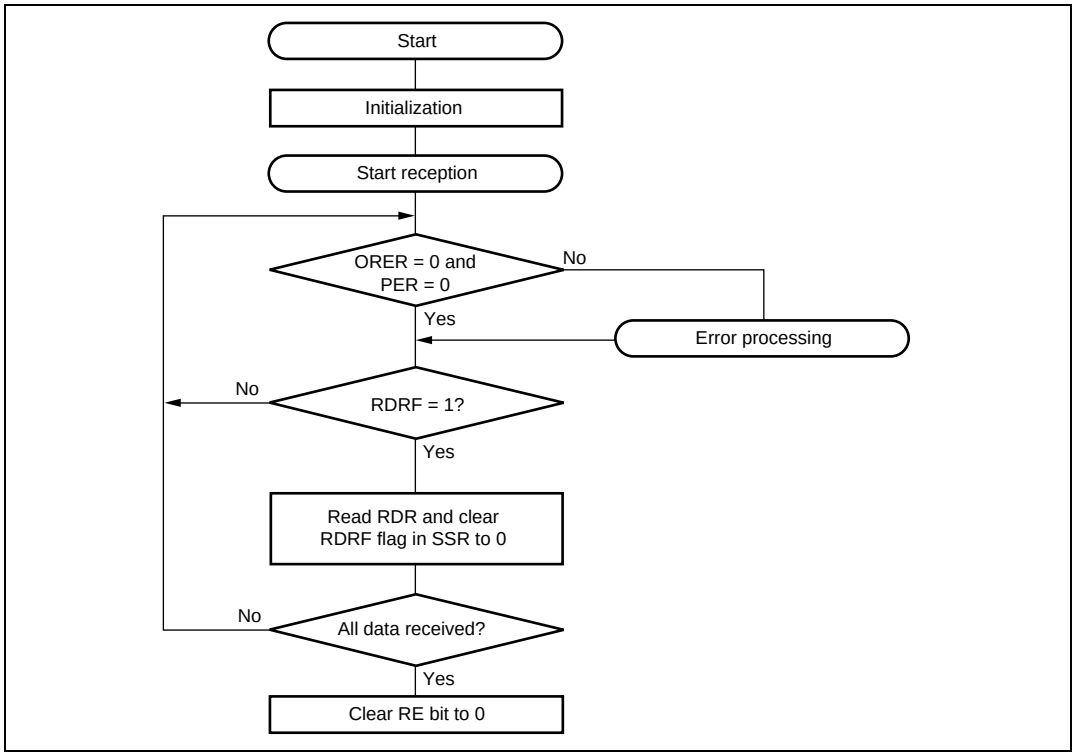


Figure 13.30 Example of Reception Processing Flow

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 CKE1 and CKE0 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.

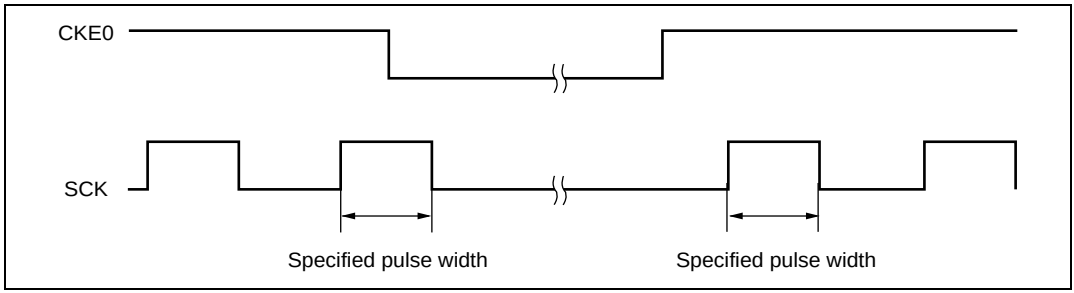


Figure 13.31 Timing for Fixing Clock Output Level

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 cycle.

Powering On: To secure the clock duty cycle 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 cycle 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 cycle.

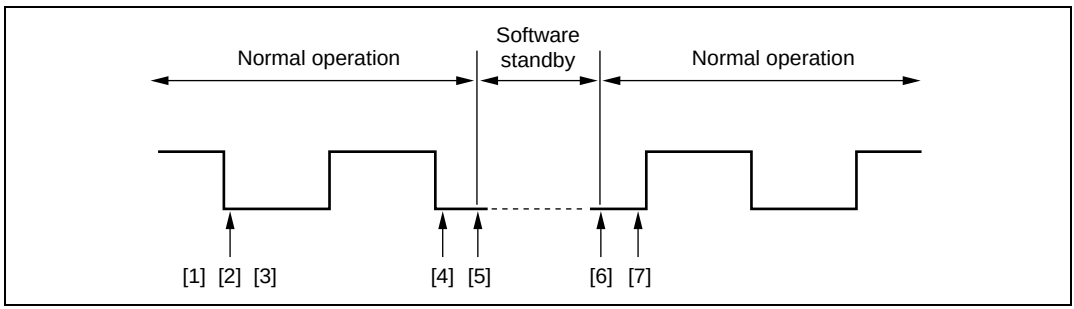


Figure 13.32 Clock Halt and Restart Procedure

13.8 IrDA Operation

When the IrDA function is enabled with bit IrE in IrCR, the SCI_0 TxD0 and RxD0 signals are subjected to waveform encoding/decoding conforming to IrDA specification version 1.0 (IrTxD and IrRxD pins). By connecting these pins to an infrared transceiver/receiver, it is possible to implement infrared transmission/reception conforming to the IrDA specification version 1.0 system.

In the IrDA specification version 1.0 system, communication is started at a transfer rate of 9600 bps, and subsequently the transfer rate can be varied as necessary. As the IrDA interface in this

LSI does not include a function for varying the transfer rate automatically, the transfer rate setting must be changed by software.

Figure 13.33 shows a block diagram of the IrDA function.

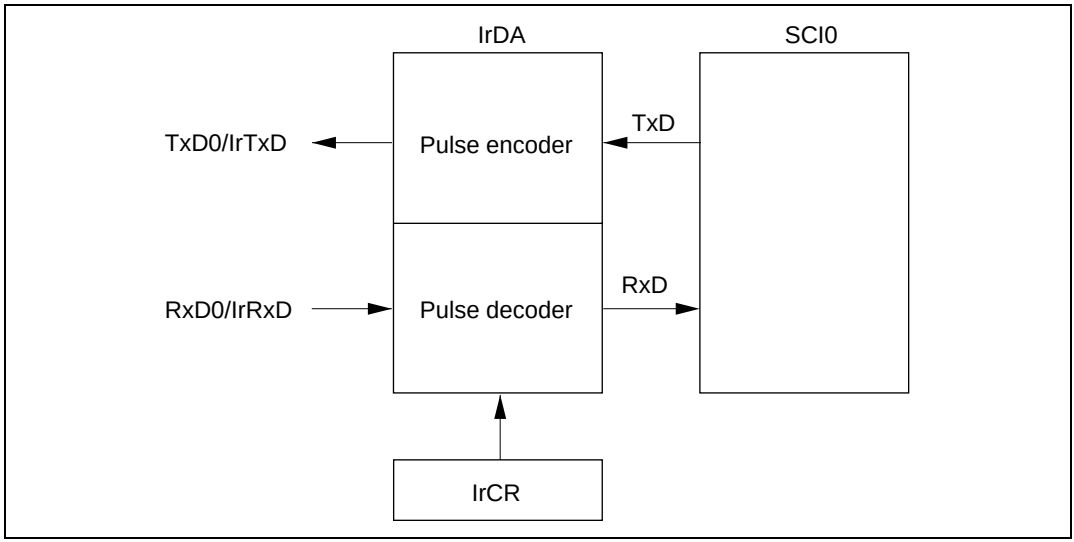


Figure 13.33 Block Diagram of IrDA

Transmission: In transmission, the output signal (UART frame) from the SCI is converted to an IR frame by the IrDA interface (see figure 13.34).

When the serial data is 0, a high pulse of 3/16 the bit rate (interval equivalent to the width of one bit) is output (initial value). The high-level pulse can be varied according to the setting of bits IrCKS2 to IrCKS0 in IrCR.

In the specification, the high pulse width is fixed at a minimum of 1.41 µs, and a maximum of (3/16 + 2.5%) × bit rate or (3/16 × bit rate) + 1.08 µs. When system clock ϕ is 20 MHz, 1.6 µs can be set for a high pulse width with a minimum value of 1.41 µs.

When the serial data is 1, no pulse is output.

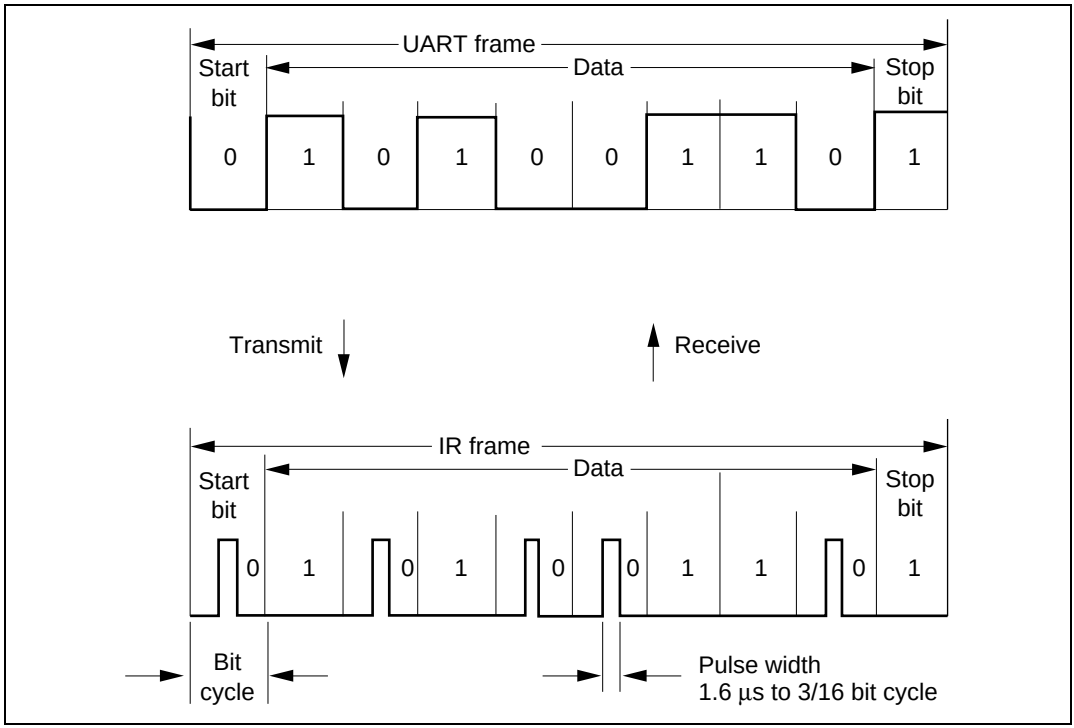


Figure 13.34 IrDA Transmit/Receive Operations

Reception: In reception, IR frame data is converted to a UART frame by the IrDA interface, and input to the SCI.

When a high pulse is detected, 0 data is output, and if there is no pulse during a one-bit interval, 1 data is output. Note that a pulse shorter than the minimum pulse width of 1.41 µs will be identified as a 0 signal.

High Pulse Width Selection: Table 13.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and operating frequencies of this LSI and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

Table 13.12 Settings of Bits IrCKS2 to IrCKS0

Operating Frequency ϕ (MHz)	Bit Rate (bps) (Above) /Bit Period × 3/16 (µs) (Below)					
	2400	9600	19200	38400	57600	115200
	78.13	19.53	9.77	4.88	3.26	1.63
2	010	010	010	010	010	—
2.097152	010	010	010	010	010	—
2.4576	010	010	010	010	010	—
3	011	011	011	011	011	—
3.6864	011	011	011	011	011	011
4.9152	011	011	011	011	011	011
5	011	011	011	011	011	011
6	100	100	100	100	100	100
6.144	100	100	100	100	100	100
7.3728	100	100	100	100	100	100
8	100	100	100	100	100	100
9.8304	100	100	100	100	100	100
10	100	100	100	100	100	100
12	101	101	101	101	101	101
12.288	101	101	101	101	101	101
14	101	101	101	101	101	101
14.7456	101	101	101	101	101	101
16	101	101	101	101	101	101
16.9344	101	101	101	101	101	101
17.2032	101	101	101	101	101	101
18	101	101	101	101	101	101
19.6608	101	101	101	101	101	101
20	101	101	101	101	101	101
25	110	110	110	110	110	110

Legend:

—: A bit rate setting cannot be made on the SCI side.

13.9 Interrupt Sources

13.9.1 Interrupts in Normal Serial Communication Interface Mode

Table 13.13 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 transfer is performed 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 perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC.

A TEI interrupt is generated when the TEND flag is set to 1 while the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are generated simultaneously, the TXI interrupt has priority for acceptance. However, note that 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 13.13 SCI Interrupt Sources

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	Low

13.9.2 Interrupts in Smart Card Interface Mode

Table 13.14 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

Table 13.14 SCI Interrupt Sources

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	Low

In Smart Card interface mode, as in normal serial communication interface mode, transfer can be carried out using the DTC. In transmit operations, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt is generated. 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 transfer is performed 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 7, Data Transfer Controller (DTC).

In receive operations, 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 transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC. If an error occurs, an

error flag is set but the RDRF flag is not. Consequently, the DTC is not activated, but instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

13.10 Usage Notes

13.10.1 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 19, Power-Down Modes.

13.10.2 Break Detection and Processing

When framing error 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, and so the FER flag is set, and the PER flag may also be set. Note that, since 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.

13.10.3 Mark State and Break Sending

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by DR and DDR. This can be used to set the TxD pin to mark state or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both PCR and PDR to 1. Since 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 PCR to 1 and clear PDR 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.

13.10.4 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.

13.10.5 Relation between Writes to TDR and the TDRE Flag

The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

13.10.6 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 13.35)
- When RDR is read by the DTC, be sure to set the activation source to the relevant SCI receive-data-full interrupt (RXI).

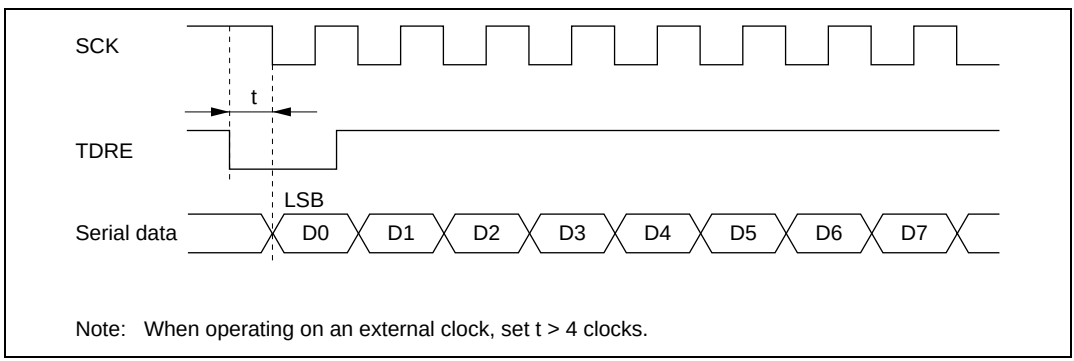


Figure 13.35 Example of Synchronous Transmission Using DTC

13.10.7 Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode or software standby mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode or software standby 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 13.36 shows a sample flowchart for mode transition during transmission. Port pin states during mode transition are shown in figures 13.37 and 13.38.

  Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before making a transition from transmission by DTC transfer to module stop mode or software standby mode transition. To perform transmission with the DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI flag and start DTC transmission.

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode or software standby mode transition. RSR, RDR, and SSR are reset. If a transition is made during reception, 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 13.39 shows a sample flowchart for mode transition during reception.

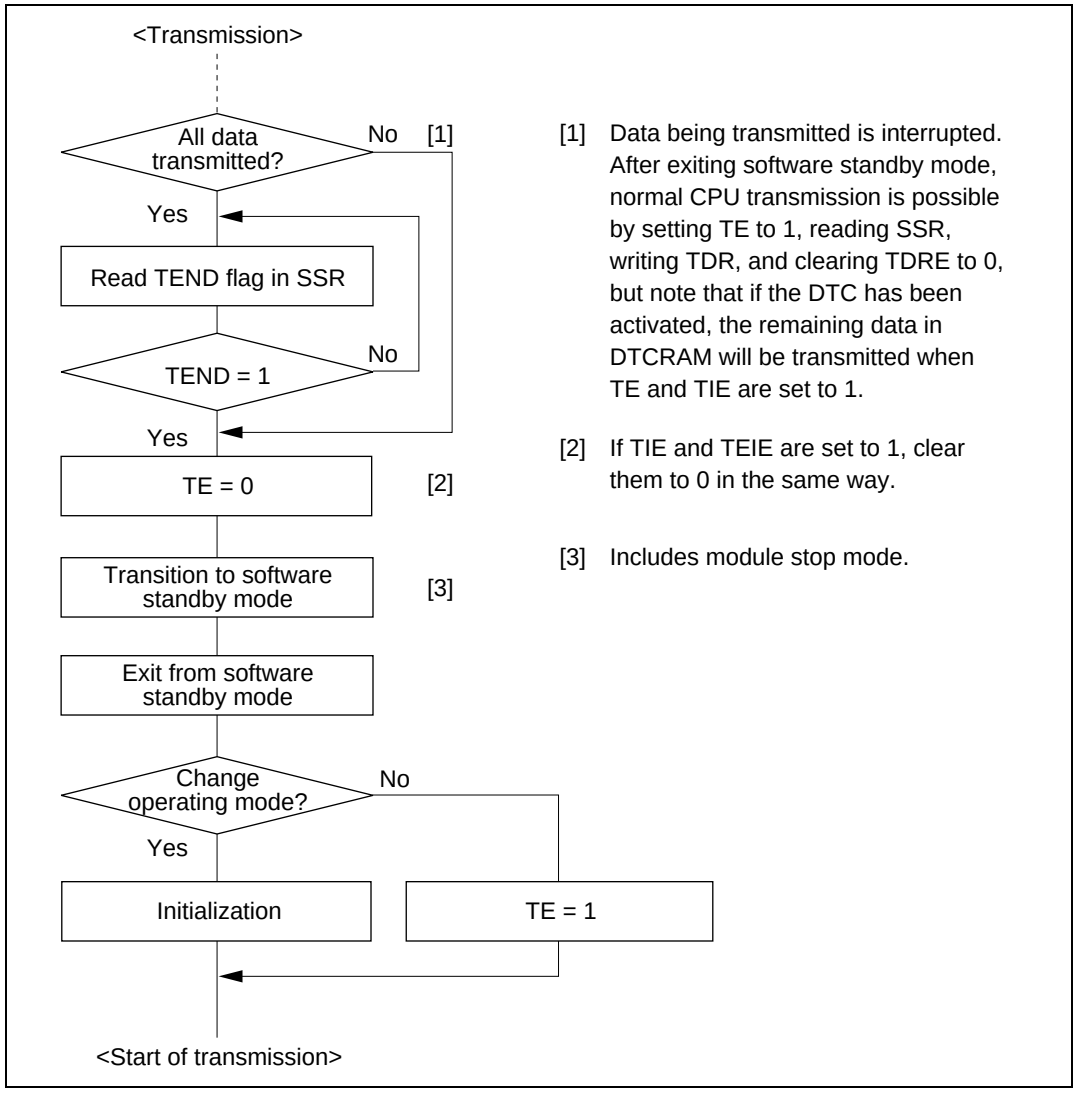


Figure 13.36 Sample Flowchart for Mode Transition during Transmission

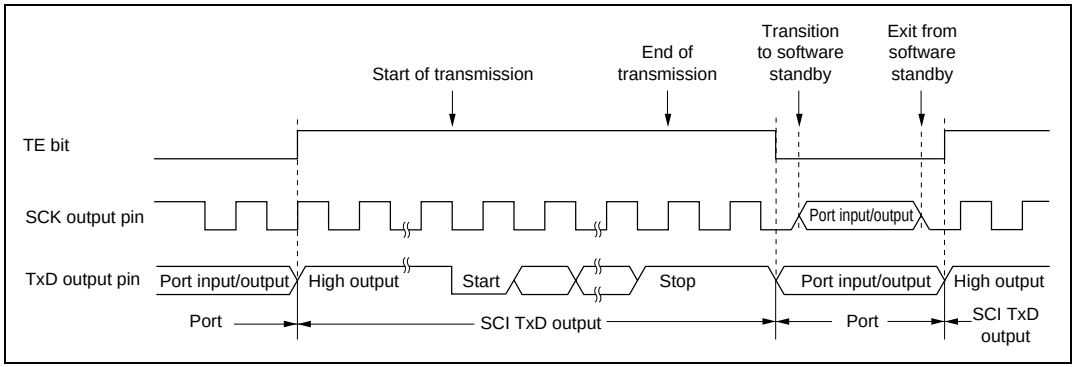


Figure 13.37 Port Pin States during Mode Transition (Internal Clock, Asynchronous Transmission)

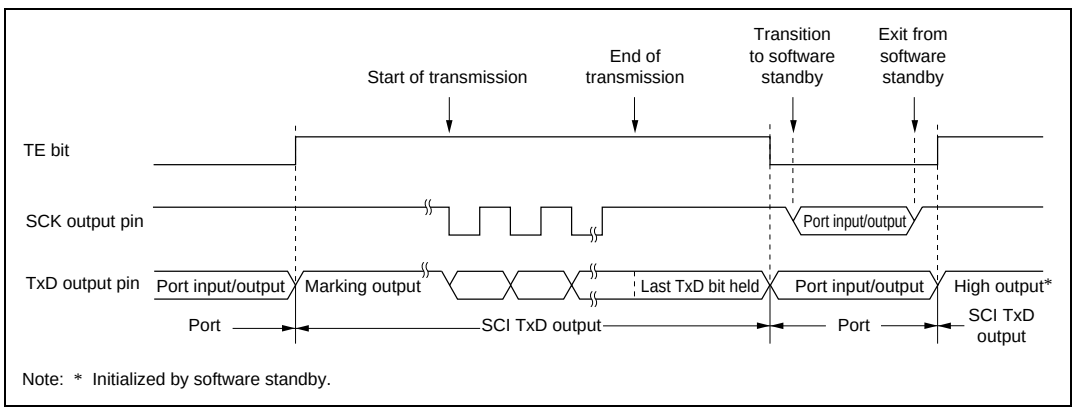


Figure 13.38 Port Pin States during Mode Transition (Internal Clock, Synchronous Transmission)

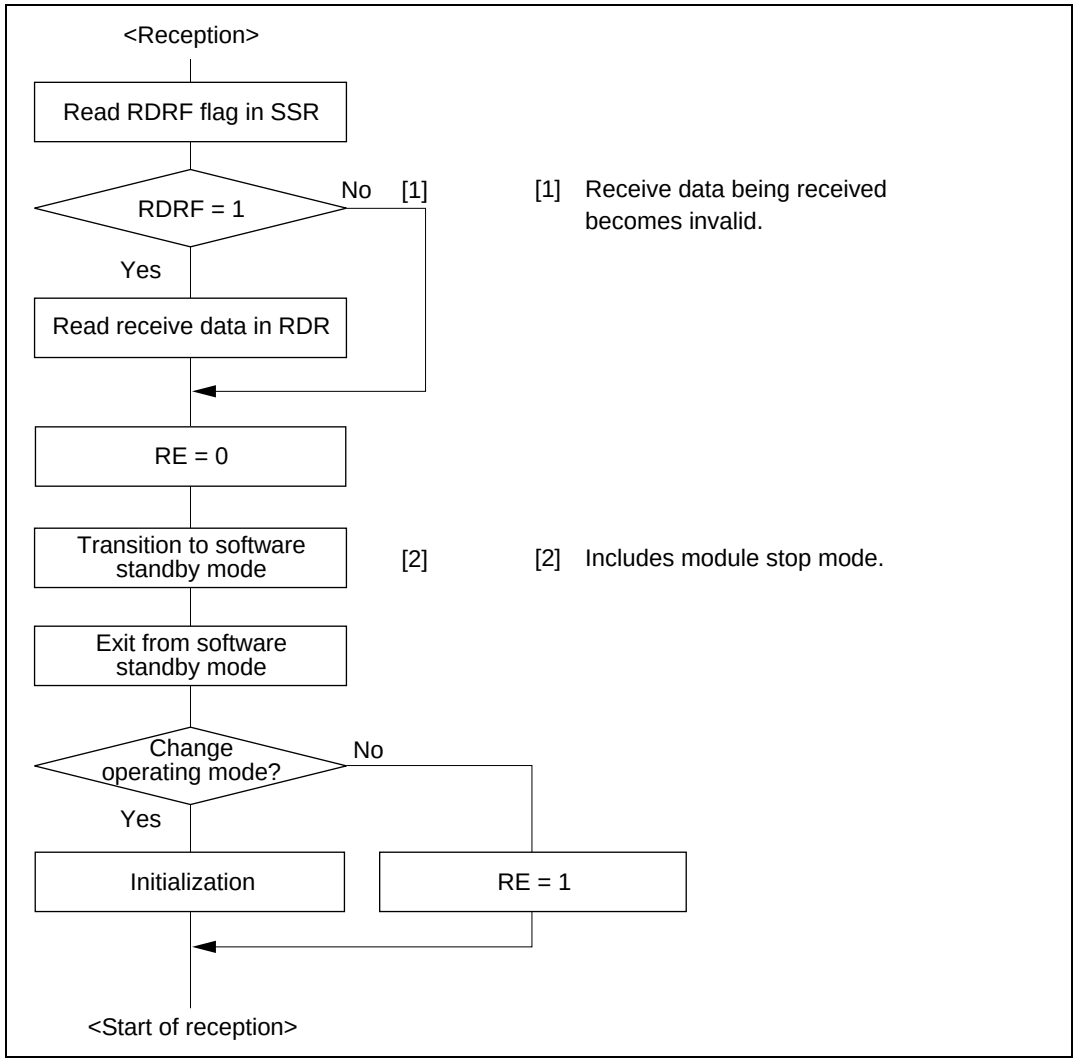


Figure 13.39 Sample Flowchart for Mode Transition during Reception

Section 14 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to twelve analog input channels to be selected. The block diagram of A/D converter is shown in figure 14.1.

14.1 Features

- 10-bit resolution
- Twelve input channels
- Conversion time: 6.7 µs per channel (at 20 MHz operation)
- Two kinds of operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels, or 1 to 8 channels
- Eight data registers
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - Conversion can be started by software, 16-bit timer pulse unit (TPU), conversion start trigger by 8-bit timer (TMR), or external trigger signal.
- Interrupt request
  - A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set

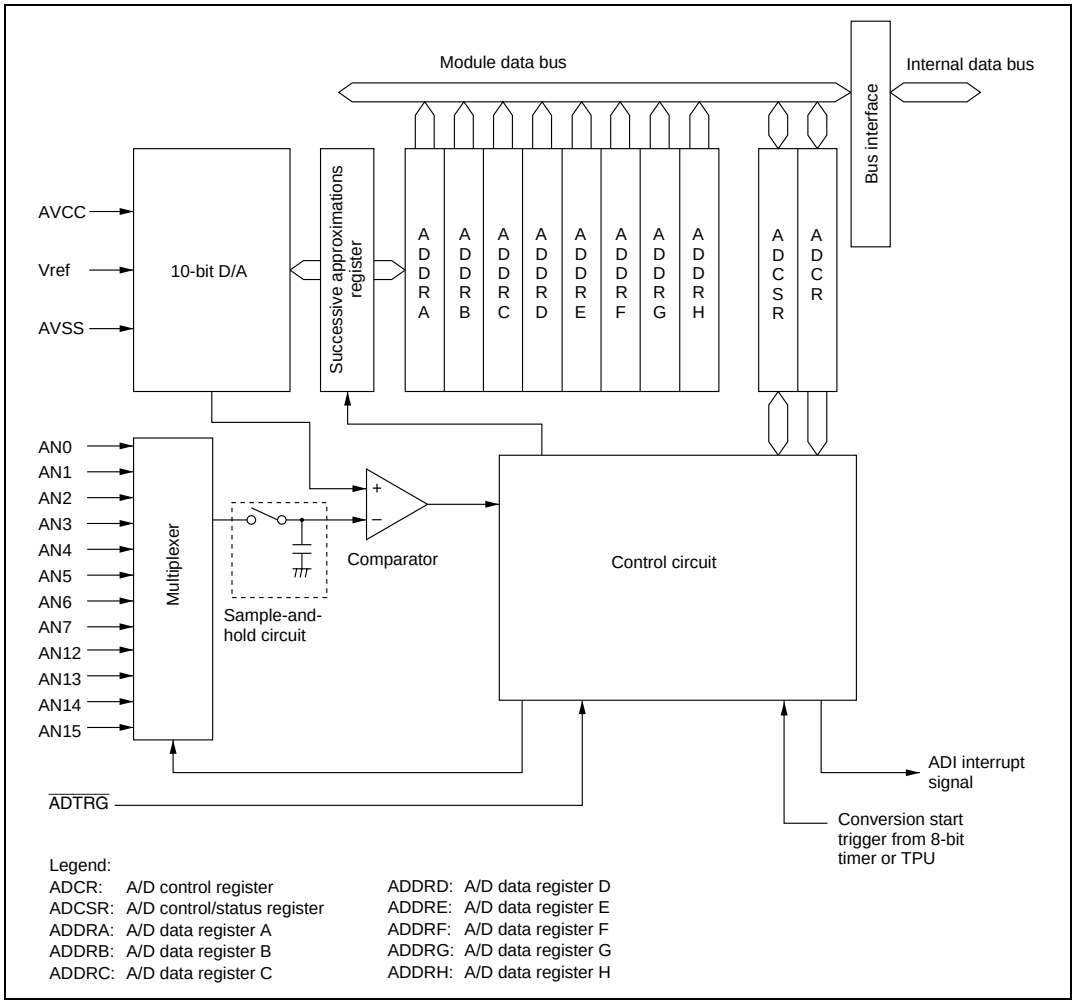


Figure 14.1 Block Diagram of A/D Converter

14.2 Input/Output Pins

Table 14.1 shows the pin configuration of the A/D converter.

The twelve analog input pins are divided into two channel sets: channel set 0 (AN0 to AN7) and channel set 1 (AN12 to AN15).

The AV_{CC} and AV_{SS} 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 14.1 A/D Converter 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
Reference voltage pin	Vref	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Channel set 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 12	AN12	Input	Channel set 1 analog inputs
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

14.3 Register Descriptions

The A/D converter has the following registers.

- 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 data register E (ADDRE)
- A/D data register F (ADDRF)
- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

14.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

There are eight 16-bit read-only ADDR registers, ADDRA to ADDRH, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each channel, are shown in table 14.2.

The converted 10-bit data is stored to bits 15 to 6. The lower 6-bit data is always read as 0. ADDR must not be accessed in 8-bit units and must be accessed in 16-bit units.

The data bus between the CPU and the A/D converter is 16-bit width. The data can be read directly from the CPU.

Table 14.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel		A/D Data Register which stores conversion result
Channel Set 0 (CH3 = 0)	**Channel Set 1 (CH3 = 1)**	
AN0	Nothing	ADDRA
AN1	Nothing	ADDRB
AN2	Nothing	ADDRC
AN3	Nothing	ADDRD
AN4	AN12	ADDRE
AN5	AN13	ADDRF
AN6	AN14	ADDRG
AN7	AN15	ADDRH

14.3.2 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 wait state. When this bit is set to 1 by software, TPU (trigger), TMR (trigger), or the $\overline{\text{ADTRG}}$ pin, A/D conversion starts. This bit remains set to 1 during A/D conversion. In single mode, cleared to 0 automatically when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by a reset, or a transition to hardware standby mode or software.
4	—	0	—	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	CH3	0	R/W	Channel select 3 to 0
2	CH2	0	R/W	Selects analog input together with bits SCANE and SCANS in ADCR.
1	CH1	0	R/W	
0	CH0	0	R/W	When SCANE = 0 and SCANS = X
				0000: AN0 1000: Setting prohibited
				0001: AN1 1001: Setting prohibited
				0010: AN2 1010: Setting prohibited
				0011: AN3 1011: Setting prohibited
				0100: AN4 1100: AN12
				0101: AN5 1101: AN13
				0110: AN6 1110: AN14
				0111: AN7 1111: AN15
				When SCANE = 1 and SCANS = 0
				0000: AN0 1000: Setting prohibited
				0001: AN0 and AN1 1001: Setting prohibited
				0010: AN0 to AN2 1010: Setting prohibited
				0011: AN0 to AN3 1011: Setting prohibited
				0100: AN4 1100: AN12
				0101: AN4 and AN5 1101: AN12 and AN13
				0110: AN4 to AN6 1110: AN12 to AN14
				0111: AN4 to AN7 1111: AN12 to AN15
				When SCANE = 1 and SCANS = 1
				0000: AN0 1000: Setting prohibited
				0001: AN0 and AN1 1001: Setting prohibited
				0010: AN0 to AN2 1010: Setting prohibited
				0011: AN0 to AN3 1011: Setting prohibited
				0100: AN0 to AN4 1100: Setting prohibited
				0101: AN0 to AN5 1101: Setting prohibited
				0110: AN0 to AN6 1110: Setting prohibited
				0111: AN0 to AN7 1111: Setting prohibited

Note: * Only 0 can be written in bit 7, to clear the flag.

Legend: X: Don't care.

14.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion start by an external trigger input.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	These bits select enabling or disabling of the start of A/D conversion by a trigger signal. 00: A/D conversion start by external trigger is disabled 01: A/D conversion start by external trigger (TPU) is enabled 10: A/D conversion start by external trigger (TMR) is enabled 11: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Selects single mode or scan mode as the A/D conversion operating mode. 0x: Single mode 10: Scan mode. A/D conversion is performed continuously for channels 1 to 4 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.
3	CKS1	0	R/W	Clock Select 1 to 0
2	CKS0	0	R/W	Sets the A/D conversion time. Only set bits CKS1 and CKS0 while conversion is stopped (ADST = 0). 00: A/D conversion time = 530 states (max) 01: A/D conversion time = 266 states (max) 10: A/D conversion time = 134 states (max) 11: A/D conversion time = 68 states (max)
1, 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

Legend: X: Don’t care.

14.4 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, to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR to halt A/D conversion. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

14.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. Operations are as follows.

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to the software 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, and is automatically cleared to 0 when conversion ends. When the ADST bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters wait state.

14.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels: maximum four channels or maximum eight channels. Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by a software, TPU or external trigger input, A/D conversion starts on the first channel in the group.

   The consecutive A/D conversion on maximum four channels (SCANE and SCANS = 10) or on maximum eight channels (SCANE and SCANS = 11) can be selected. When the consecutive A/D conversion is performed on the four channels, the A/D conversion starts on AN0 when CH3 and CH2 =00, AN4 when CH3 and CH2 = 01, or AN12 when CH3 and CH2 = 11. When the consecutive A/D conversion is performed on the eight channels, the A/D conversion starts on AN0 when SH3 and SH2 =00.
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the corresponding A/D data register to each channel.

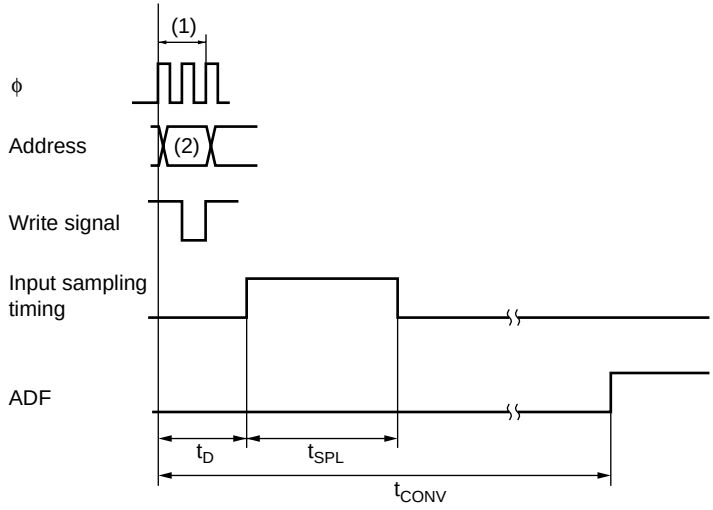
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. Conversion of the first channel in the group starts again.
4. The ADST bit is not cleared automatically, and steps [2] to [3] are repeated 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 wait state. If the ADST bit is later set to 1, A/D conversion starts again from the first channel in the group.

14.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when A/D conversion start delay time (t_D) passes after the ADST bit is set to 1, then starts conversion. Figure 14.2 shows the A/D conversion timing. Table 14.3 indicates the A/D conversion time.

As indicated in figure 14.2, 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 tables 14.3.

In scan mode, the values given in tables 14.3 apply to the first conversion time. The values given in tables 14.4 apply to the second and subsequent conversions. The conversion time must be within the ranges indicated in the descriptions, A/D Conversion Characteristics in section 21, Electrical Characteristics. Therefore the CKS1 and CKS0 bits must be set to satisfy this condition.



Legend:
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 14.2 A/D Conversion Timing

Table 14.3 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: Values in the table are the number of states.

Table 14.4 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)

14.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to 11 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 14.3 shows the timing.

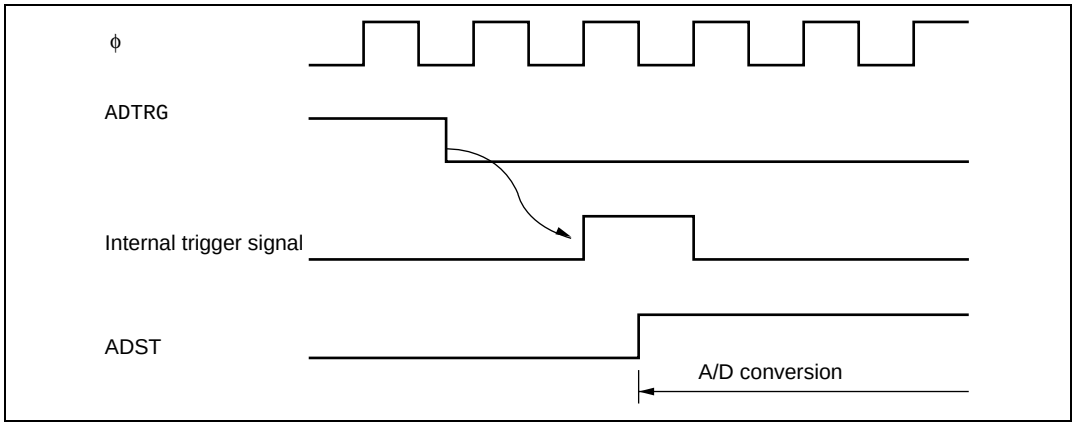


Figure 14.3 External Trigger Input Timing

14.5 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 an ADI interrupt requests while the bit ADF in ADCSR is set to 1 after A/D conversion is completed. The 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 to be achieved without imposing a load on software.

Table 14.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
ADI	End of conversion	ADF	Possible

14.6 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 14.4).
- 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 14.5).
- 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 14.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 14.5).
- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

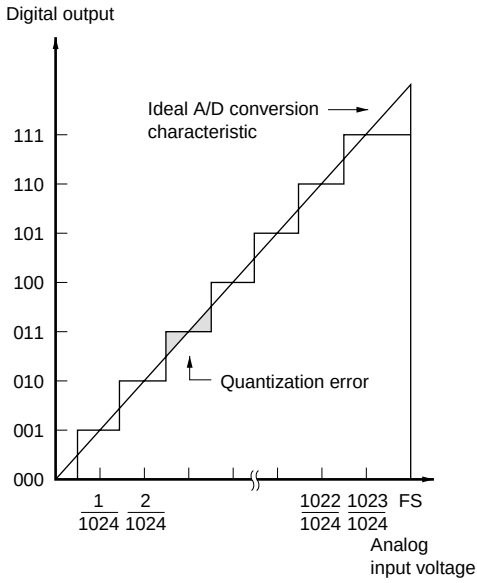


Figure 14.4 A/D Conversion Accuracy Definitions

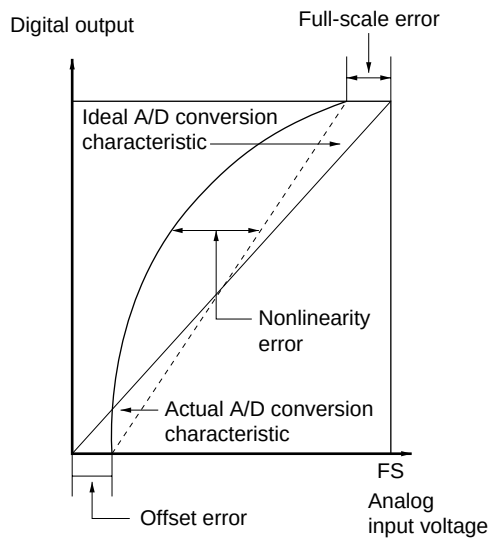


Figure 14.5 A/D Conversion Accuracy Definitions

14.7 Usage Notes

14.7.1 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 19, Power-Down Modes.

14.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 10 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 10 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion accuracy. However, if a large capacitance is provided externally for conversion in single mode, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance becomes unnecessary. However, since 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 14.6). When converting a high-speed analog signal or conversion in scan mode, a low-impedance buffer should be inserted.

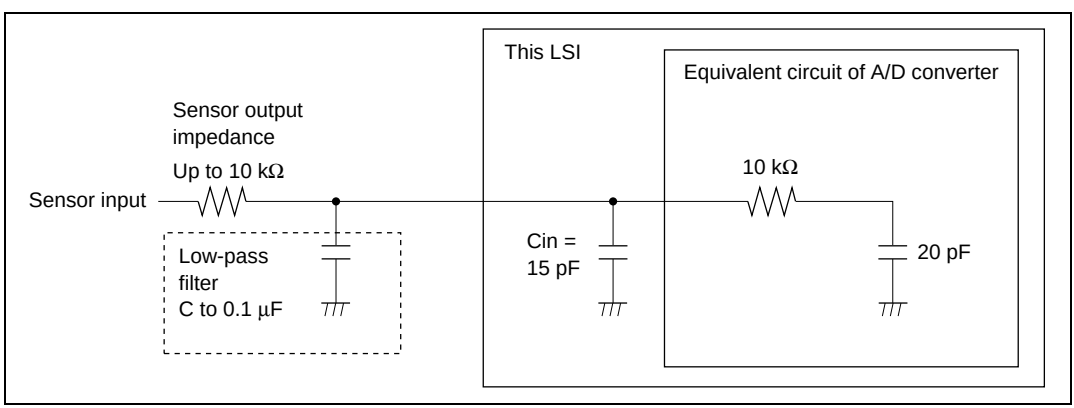


Figure 14.6 Example of Analog Input Circuit

14.7.3 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. 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, so acting as antennas.

14.7.4 Setting Range of Analog Power Supply and Other Pins

If conditions shown 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 ≤ V_{AN} ≤ Vref.
- Relation between AVcc, AVss and Vcc, Vss

  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc ≥ Vcc and AVss = Vss. If the A/D converter is not used, the AVcc and AVss pins must not be left open.
- Vref setting range

  The reference voltage at the Vref pin should be set in the range Vref ≤ AVcc.

14.7.5 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 AN7 and AN12 to AN15), analog reference power supply (Vref), 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.

14.7.6 Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN7 and AN12 to AN15) should be connected between AVcc and AVss as shown in figure 14.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN7 and AN12 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN7 and AN12 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 the circuit constants.

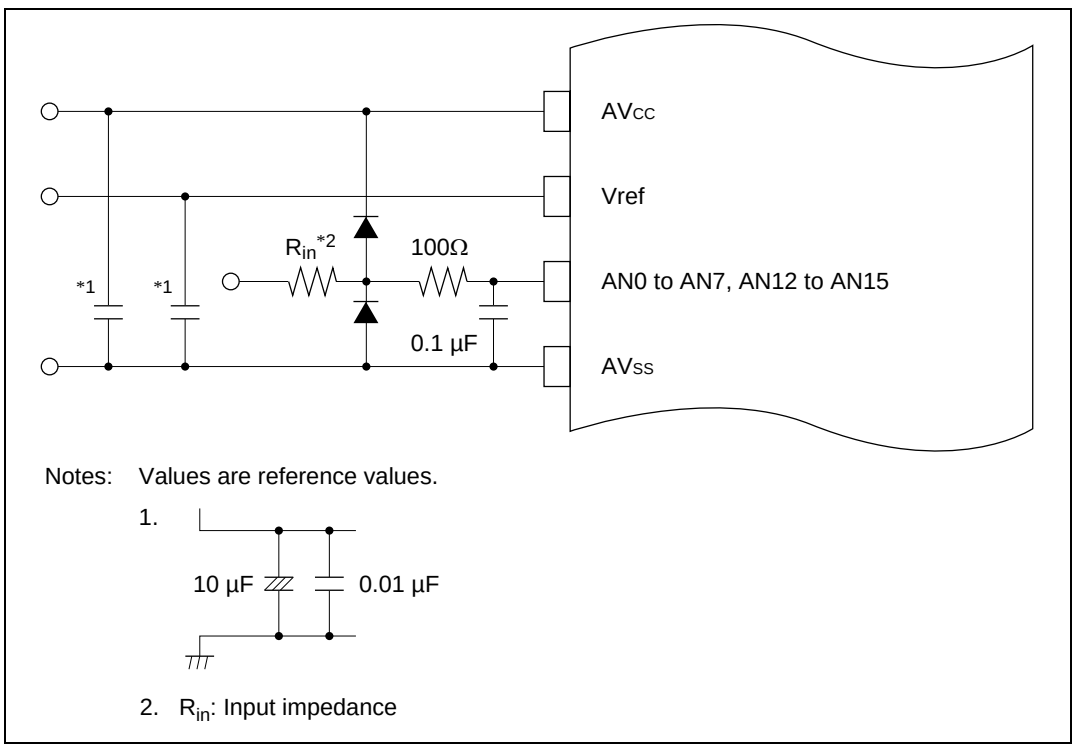
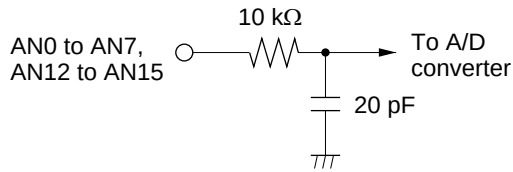


Figure 14.7 Example of Analog Input Protection Circuit

Table 14.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	10	kΩ



Note: Values are reference values.

Figure 14.8 Analog Input Pin Equivalent Circuit

Section 15 D/A Converter

15.1 Features

D/A converter features are listed below.

- 8-bit resolution
- Four output channels
- Maximum conversion time of 10 µs (with 20 pF load)
- Output voltage of 0 V to Vref
- D/A output hold function in software standby mode
- Setting the module stop mode

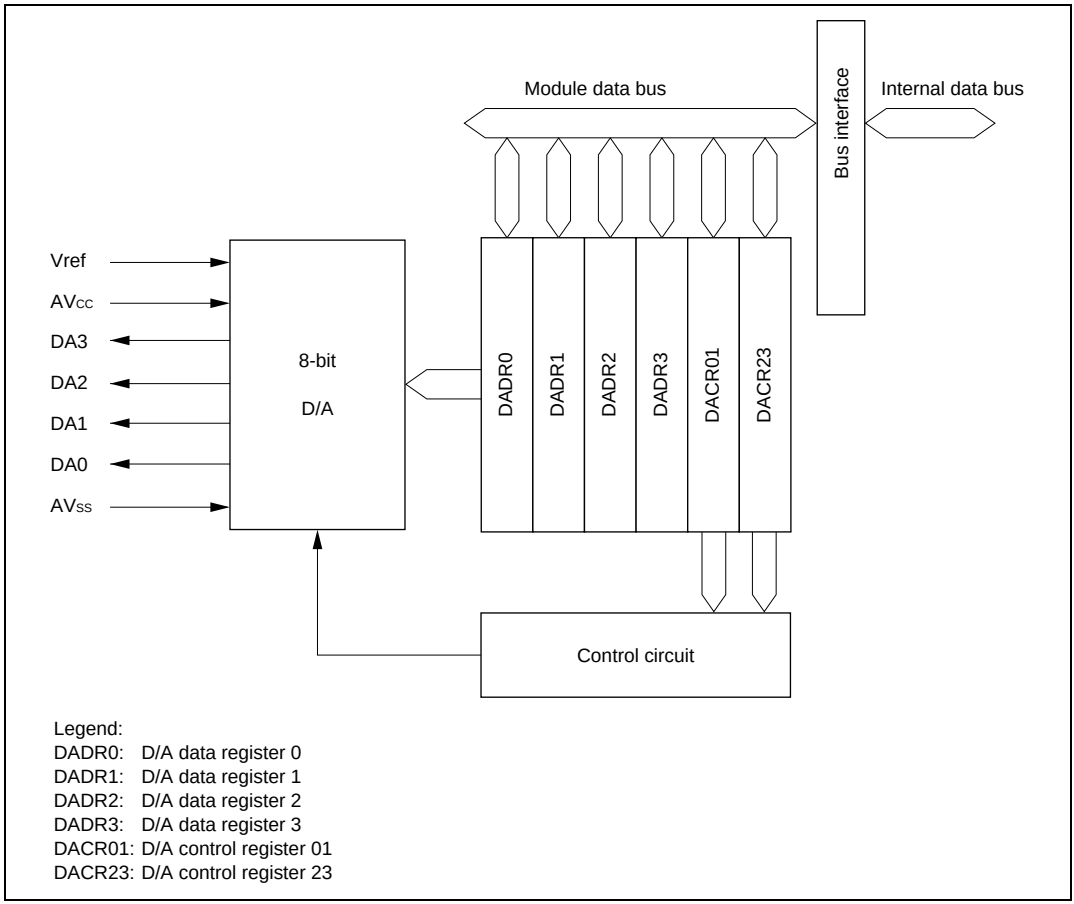


Figure 15.1 Block Diagram of D/A Converter

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the D/A converter.

Table 15.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power pin	AV_{CC}	Input	Analog power
Analog ground pin	AV_{SS}	Input	Analog ground
Reference voltage pin	Vref	Input	Reference voltage of D/A converter
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output
Analog output pin 2	DA2	Output	Channel 2 analog output
Analog output pin 3	DA3	Output	Channel 3 analog output

15.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A data register 2 (DADR2)
- D/A data register 3 (DADR3)
- D/A control register 01 (DACR01)
- D/A control register 23 (DACR23)

15.3.1 D/A Data Registers 0 to 3 (DADR0 to DADR3)

DADR0 to DADR3 are 8-bit readable/writable registers that store data for conversion.

Whenever output is enabled, the values in DADR are converted and output to the analog output pins.

15.3.2 D/A Control Registers 01 and 23 (DACR01, DACR23)

DACR01 and DACR23 control the operation of the D/A converter.

DACR01

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: Channel 1 D/A conversion is enabled; analog output (DA1) is 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: Channel 0 D/A conversion is enabled; analog output (DA0) is enabled
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 0 and 1 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 0 and 1 D/A conversions are controlled together. Output of conversion results is always controlled independently by the DAOE0 and DAOE1 bits. For details, see table 15.2 Control of D/A Conversion.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 15.2 Control of D/A Conversion

Bit 5 DAE	Bit 7 DAOE1	Bit 6 DAOE0	Description
0	0	0	D/A conversion disabled
		1	Channel 0 D/A conversion enabled, channel1 D/A conversion disabled
	1	0	Channel 1 D/A conversion enabled, channel0 D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
1	0	0	D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
	1	0	
		1	

DACR23

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE3	0	R/W	D/A Output Enable 3 Controls D/A conversion and analog output. 0: Analog output (DA3) is disabled 1: Channel 3 D/A conversion is enabled; analog output (DA3) is enabled
6	DAOE2	0	R/W	D/A Output Enable 2 Controls D/A conversion and analog output. 0: Analog output (DA2) is disabled 1: Channel 2 D/A conversion is enabled; analog output (DA2) is enabled
5	DAE	0	R/W	D/A Enable Used together with the DAOE2 and DAOE3 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 2 and 3 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 2 and 3 D/A conversions are controlled together. Output of conversion results is always controlled independently by the DAOE2 and DAOE3 bits. For details, see table 15.3 Control of D/A Conversion.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 15.3 Control of D/A Conversion

Bit 5 DAE	Bit 7 DAOE3	Bit 6 DAOE2	Description
0	0	0	D/A conversion disabled
		1	Channel 2 D/A conversion enabled, channel3 D/A conversion disabled
	1	0	Channel 3 D/A conversion enabled, channel2 D/A conversion disabled
		1	Channel 2 and 3 D/A conversions enabled
1	0	0	D/A conversion disabled
		1	Channel 2 and 3 D/A conversions enabled
	1	0	
		1	

15.4 Operation

The D/A converter includes D/A conversion circuits for four channels, each of which can operate independently.

When DAOE bit in DACR01 or DACR23 is set to 1, D/A conversion is enabled and the conversion result is output.

The operation example concerns D/A conversion on channel 0. Figure 15.2 shows the timing of this operation.

[1] Write the conversion data to DADR0.

[2] Set the DAOE0 bit in DACR01 to 1. D/A conversion is started. The conversion result is output from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The conversion result is continued to output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times \text{Vref}$$

[3] If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time t_{DCONV} has elapsed.

[4] If the DAOE0 bit is cleared to 0, analog output is disabled.

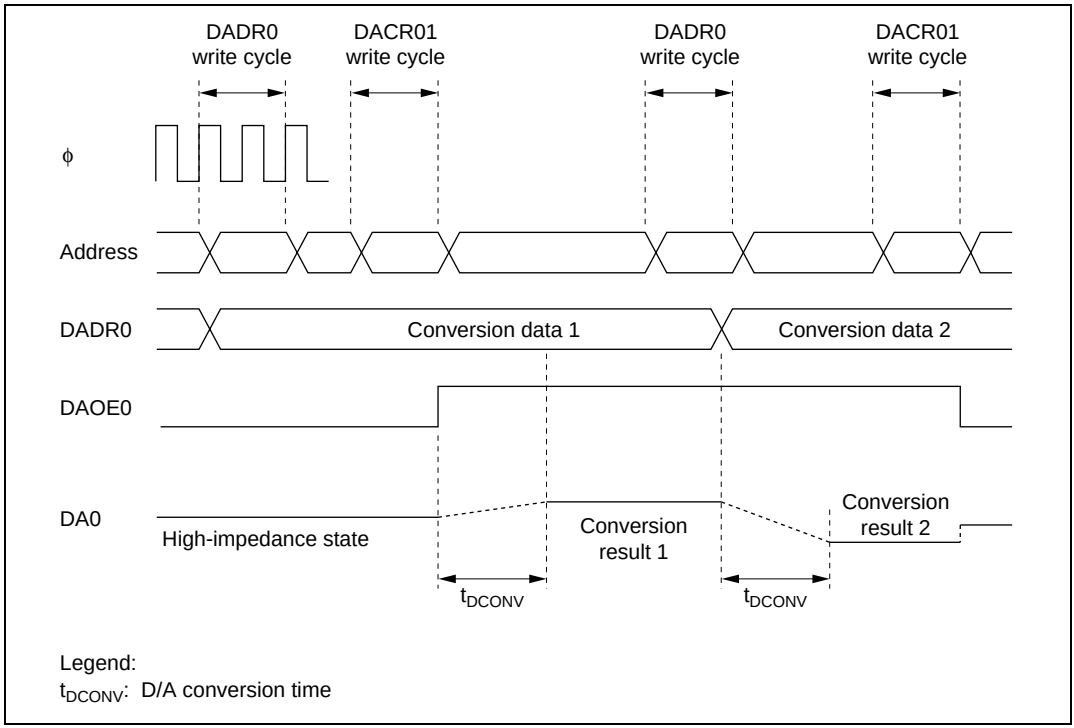


Figure 15.2 Example of D/A Converter Operation

15.5 Usage Notes

15.5.1 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 details, see section 19, Power-Down Modes.

15.5.2 D/A Output Hold Function in Software Standby Mode

If D/A conversion is enabled and this LSI enters software standby mode, D/A output is held and analog power supply current remains at the same level during D/A conversion. When the analog power supply current is required to go low in software standby mode, bits DAOE0 to DAOE3 and DAE should be cleared to 0, and D/A output should be disabled.

Section 16 RAM

This LSI has an on-chip 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 on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on the system control register (SYSCR), refer to section 3.2.2, System Control Register (SYSCR).

Product Type		ROM Type	RAM Capacitance	RAM Address
H8S/2668 Group	HD64F2667	Flash memory version	16 kbytes	H'FF8000 to H'FFBFFF

Section 17 Flash Memory (F-ZTAT Version)

The features of the flash memory included in the flash memory version are summarized below. The block diagram of the flash memory is shown in figure 17.1.

17.1 Features

- Size

Product Classification		ROM Size	ROM Address
H8S/2668 Group	HD64F2667	384 kbytes	H'000000 to H'05FFFF (Modes 3, 4, and 7) H'100000 to H'15FFFF (Modes 5 and 6)

- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory of 384 kbytes is configured as follows: 64 kbytes × 5 blocks, 32 kbytes × 1 block, and 4 kbytes × 8 block. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- Two on-board programming modes

  Boot mode

  User program mode

  On-board programming/erasing can be done in boot mode in which the on-chip boot program is started for erase or programming of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment

  With data transfer in boot mode, the bit rate of this LSI can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation by RAM

  Part of the RAM area can be overlapped onto flash memory, to emulate flash memory updates in real time.
- Programming/erasing protection

  There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase operations.

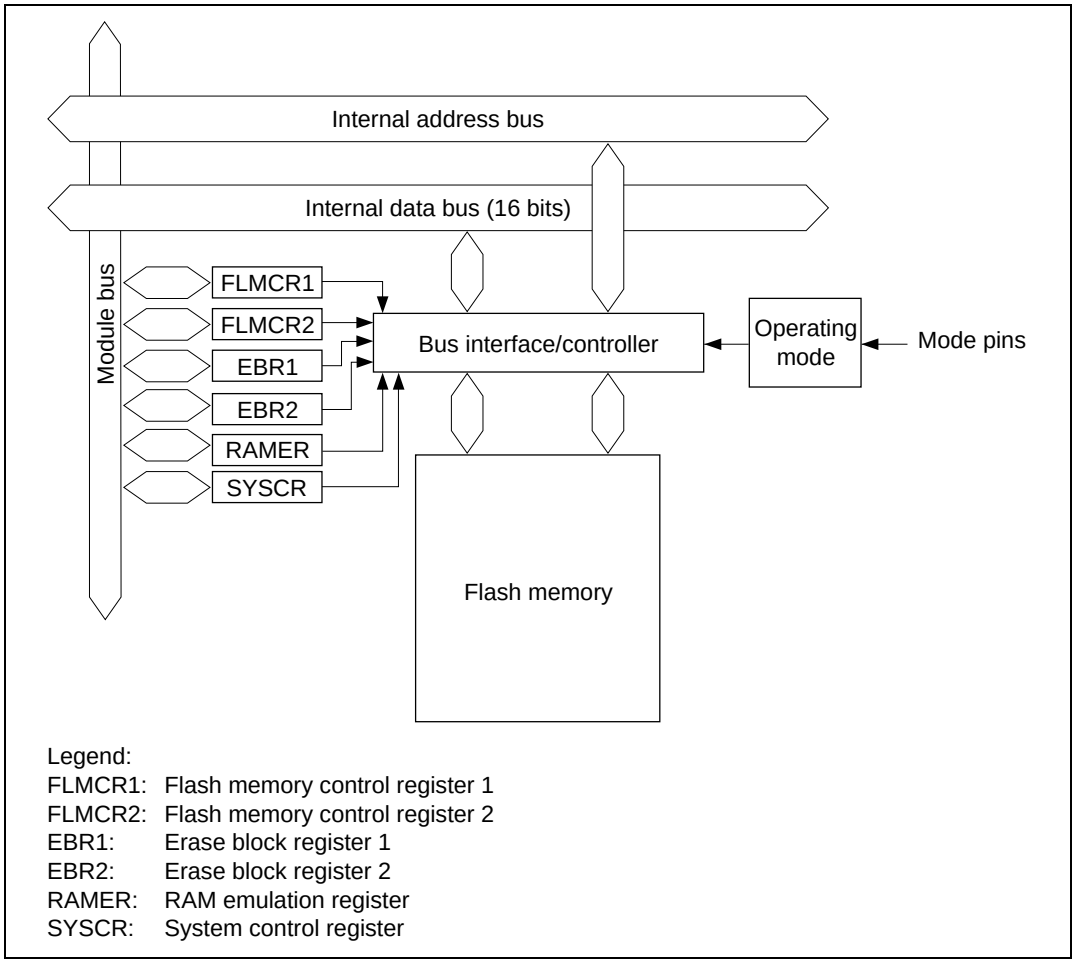


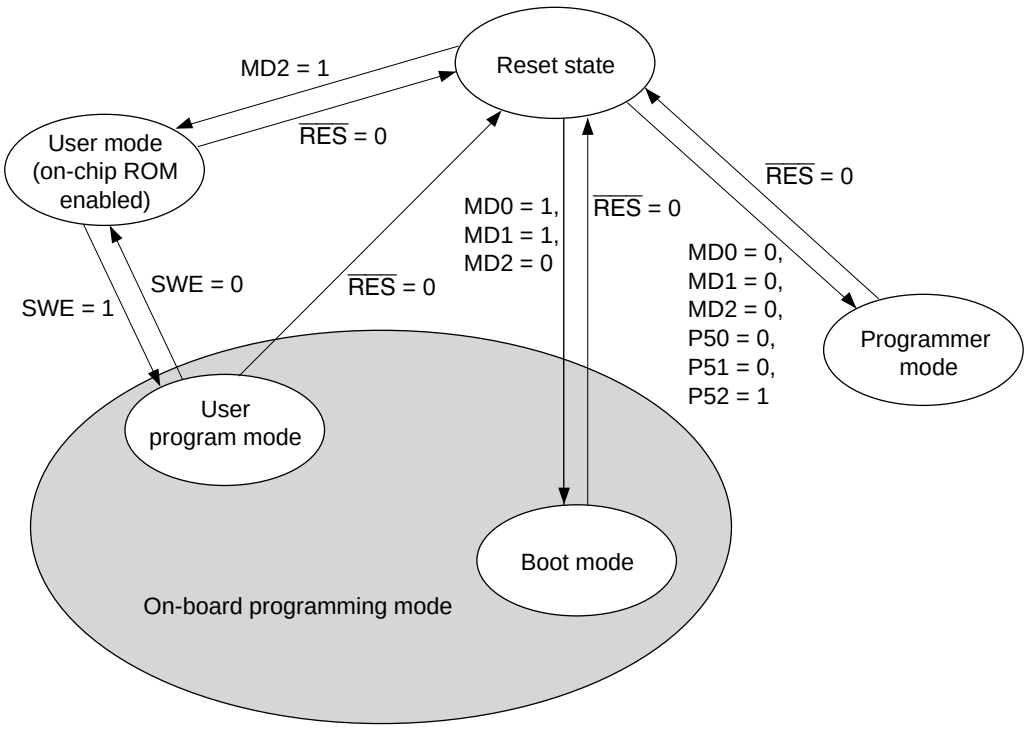
Figure 17.1 Block Diagram of Flash Memory

17.2 Mode Transitions

When the mode pins are set in the reset state and a reset-start is executed, this LSI enters an operating mode as shown in figure 17.2. In user mode, flash memory can be read but not programmed or erased.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.

The differences between boot mode and user program mode are shown in table 17.1. Figure 17.3 shows boot mode. Figure 17.4 shows user program mode.



Note: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

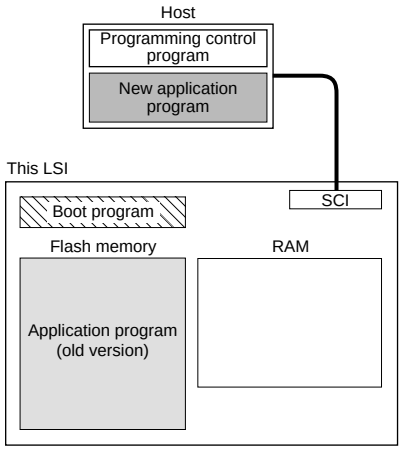
Figure 17.2 Flash Memory State Transitions

Table 17.1 Differences between Boot Mode and User Program Mode

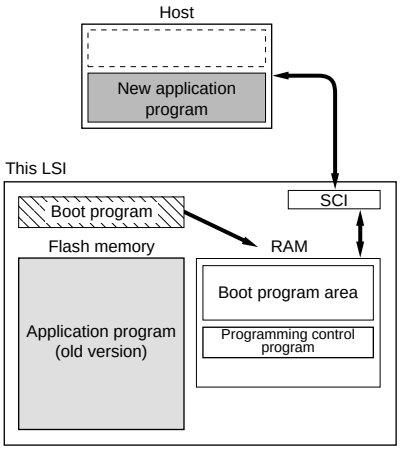
	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify/program/ program-verify emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

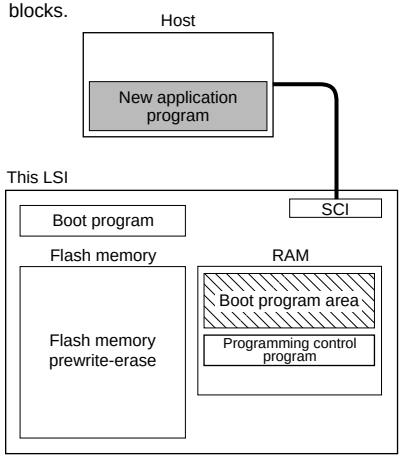
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



2. Programming control program transfer
   When boot mode is entered, the boot program in the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, entire flash memory erasure is performed, without regard to blocks.



4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

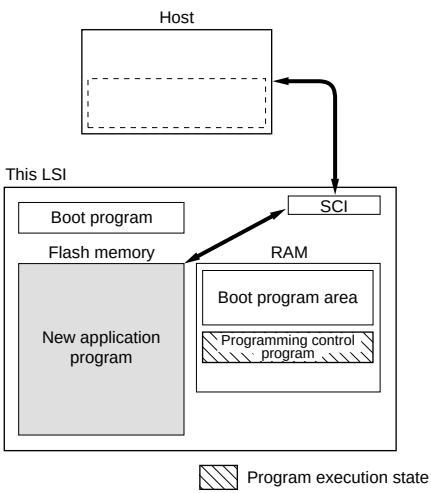
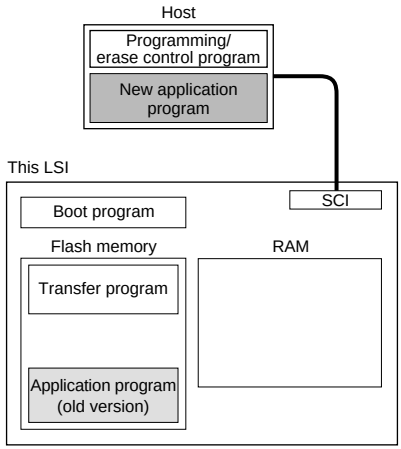
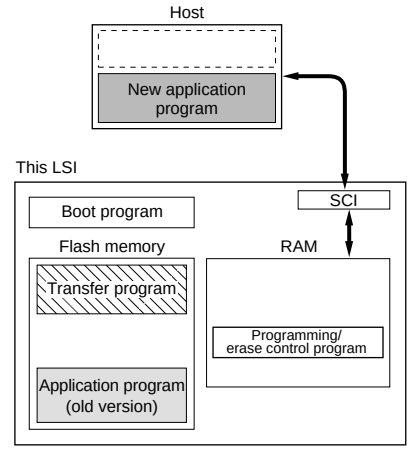


Figure 17.3 Boot Mode

1. Initial state
   (1) The program that will transfer the programming/ erase control program to on-chip RAM should be written into the flash memory by the user beforehand. (2) The programming/erase control program should be prepared in the host or in the flash memory.

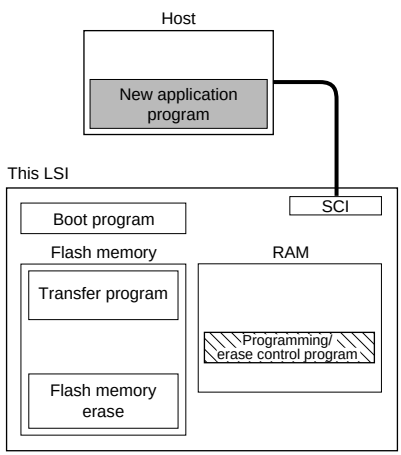
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.





3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.

4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



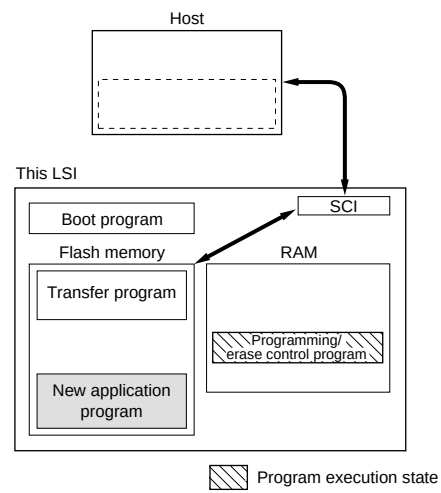
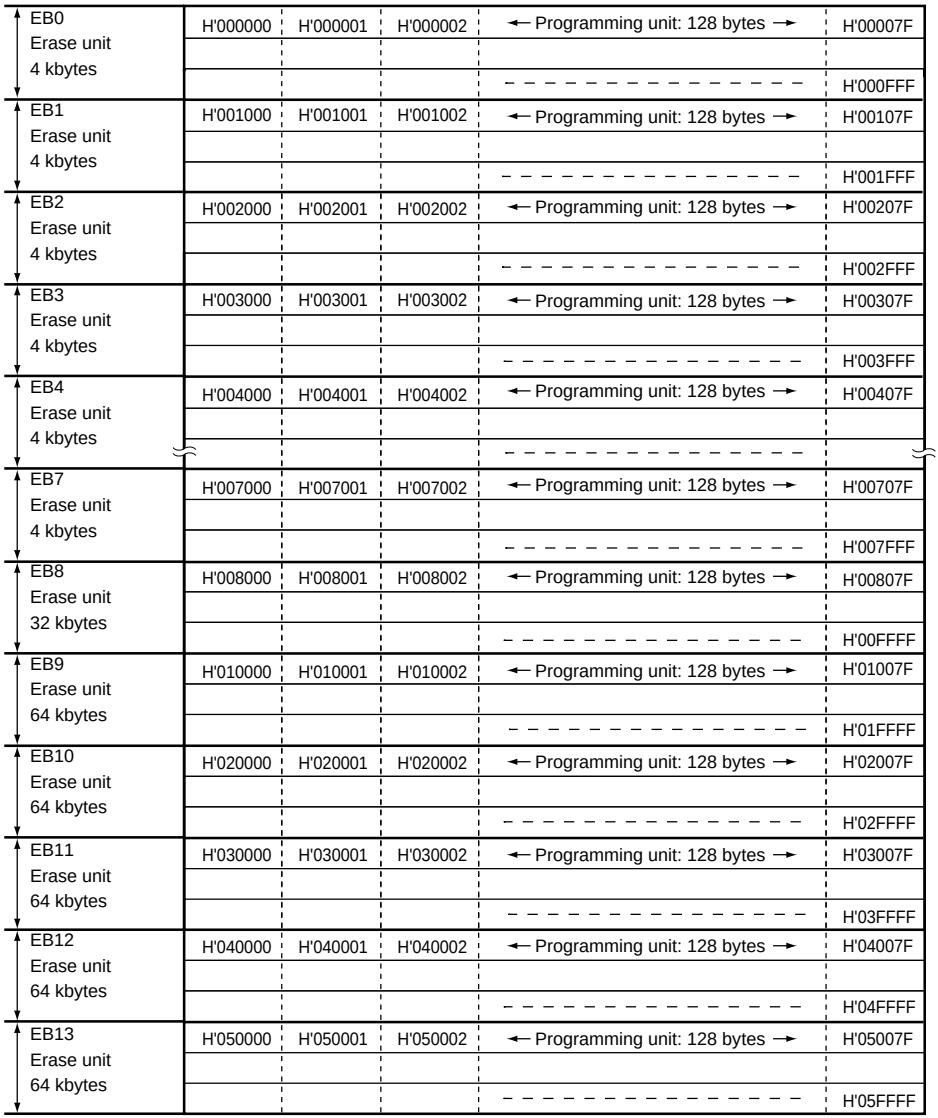


Figure 17.4 User Program Mode

17.3 Block Configuration

Figure 17.5 shows the block configuration of 384-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The 384-kbyte flash memory is divided into 64 kbytes (5 blocks), 32 kbytes (1 block), and 4 kbytes (8 blocks). Erasing is performed in these divided units. Programming is performed in 128-byte units starting from an address whose lower eight bits are H'00 or H'80.



Note: Addresses H'100000 to H'15FFFF are allocated in modes 5 and 6.

Figure 17.5 384-Kbyte Flash Memory Block Configuration (Modes 3, 4, and 7)

17.4 Input/Output Pins

Table 17.2 shows the pin configuration of the flash memory.

Table 17.2 Pin Configuration

Pin Name	I/O	Function
$\overline{\text{RES}}$	Input	Reset
MD2	Input	Sets this LSI's operating mode
MD1	Input	Sets this LSI's operating mode
MD0	Input	Sets this LSI's operating mode
P52	Input	Sets operating mode in programmer mode
P51	Input	Sets operating mode in programmer mode
P50	Input	Sets operating mode in programmer mode
TxD1	Output	Serial transmit data output
RxD1	Input	Serial receive data input

17.5 Register Descriptions

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

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)
- RAM emulation register (RAMER)

17.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory transit to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, refer to section 17.8, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0/1	R	This bit is reserved. This bit is always read as 0 in modes 1 and 2. This bit is always read as 1 in modes 3 to 7. The initial value should not be changed.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 and EBR2 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1 while SWE = 1, the flash memory transits to the erase setup state. When it is cleared to 0, the erase setup state is cancelled.
4	PSU	0	R/W	Program Setup When this bit is set to 1 while SWE = 1, the flash memory transits to the program setup state. When it is cleared to 0, the program setup state is cancelled.
3	EV	0	R/W	Erase-Verify When this bit is set to 1 while SWE = 1, the flash memory transits to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1 while SWE = 1, the flash memory transits to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1 while SWE = 1, and ESU = 1, the flash memory transits to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P	0	R/W	Program When this bit is set to 1 while SWE = 1, and PSU = 1, the flash memory transits to program mode. When it is cleared to 0, program mode is cancelled.

17.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLMCR2 is a read-only register, and should not be written to. When the on-chip flash memory is disabled, the contents of FLMCR2 are always read as H'00.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See 17.9.3 Error Protection, for details.
6 to 0	—	All 0	R	Reserved These bits are always read 0.

17.5.3 Erase Block Register 1 (EBR1)

EBR1 specifies the flash memory erase area block. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Set only one bit in EBR1 and EBR2 together (do not set more than one bit at the same time). Setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0. For details, see table 17.3, Erase Blocks.

Bit	Bit Name	Initial Value	R/W	Description
7	EB7	0	R/W	When this bit is set to 1, 4 kbytes of EB7 are to be erased.
6	EB6	0	R/W	When this bit is set to 1, 4 kbytes of EB6 are to be erased.
5	EB5	0	R/W	When this bit is set to 1, 4 kbytes of EB5 are to be erased.
4	EB4	0	R/W	When this bit is set to 1, 4 kbytes of EB4 are to be erased.
3	EB3	0	R/W	When this bit is set to 1, 4 kbytes of EB3 is to be erased.
2	EB2	0	R/W	When this bit is set to 1, 4 kbytes of EB2 is to be erased.
1	EB1	0	R/W	When this bit is set to 1, 4 kbytes of EB1 is to be erased.
0	EB0	0	R/W	When this bit is set to 1, 4 kbytes of EB0 is to be erased.

17.5.4 Erase Block Register 2 (EBR2)

EBR2 specifies the flash memory erase area block. EBR2 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Set only one bit in EBR2 and EBR1 together (do not set more than one bit at the same time). Setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0. For details, see table 17.3, Erase Blocks.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	EB13	0	R/W	When this bit is set to 1, 64 kbytes of EB13 are to be erased.
4	EB12	0	R/W	When this bit is set to 1, 64 kbytes of EB12 are to be erased.
3	EB11	0	R/W	When this bit is set to 1, 64 kbytes of EB11 are to be erased.
2	EB10	0	R/W	When this bit is set to 1, 64 kbytes of EB10 are to be erased.
1	EB9	0	R/W	When this bit is set to 1, 64 kbytes of EB9 are to be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 kbytes of EB8 are to be erased.

Table 17.3 Erase Blocks

	Address	
Block (Size)	**Modes 3, 4, and 7**	**Modes 5 and 6**
EB0 (4 kbytes)	H'000000 to H'000FFF	H'100000 to H'100FFF
EB1 (4 kbytes)	H'001000 to H'001FFF	H'101000 to H'101FFF
EB2 (4 kbytes)	H'002000 to H'002FFF	H'102000 to H'102FFF
EB3 (4 kbytes)	H'003000 to H'003FFF	H'103000 to H'103FFF
EB4 (4 kbytes)	H'004000 to H'004FFF	H'104000 to H'104FFF
EB5 (4 kbytes)	H'005000 to H'005FFF	H'105000 to H'105FFF
EB6 (4 kbytes)	H'006000 to H'006FFF	H'106000 to H'106FFF
EB7 (4 kbytes)	H'007000 to H'007FFF	H'107000 to H'107FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF	H'108000 to H'10FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF	H'110000 to H'11FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF	H'120000 to H'12FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF	H'130000 to H'13FFFF
EB12 (64 kbytes)	H'040000 to H'04FFFF	H'140000 to H'14FFFF
EB13 (64 kbytes)	H'050000 to H'05FFFF	H'150000 to H'15FFFF

17.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER settings should be made in user mode or user program mode. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits always read 0.
4	—	0	R/W	Reserved The initial value should not be changed.
3	RAMS	0	R/W	RAM Select Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, the flash memory is overlapped with part of RAM, and all flash memory block are in the program/erase-protect state. When this bit is cleared to 0, the RAM emulation function is invalid.
2	RAM2	0	R/W	Flash Memory Area Selection
1	RAM1	0	R/W	When the RAMS bit is set to 1, selects one of the following flash memory areas to overlap the RAM area. The areas correspond with 4-kbyte erase blocks.
0	RAM0	0	R/W	Modes 3, 4, and 7 000: H'000000 to H'000FFF (EB0) 001: H'001000 to H'001FFF (EB1) 010: H'002000 to H'002FFF (EB2) 011: H'003000 to H'003FFF (EB3) 100: H'004000 to H'004FFF (EB4) 101: H'005000 to H'005FFF (EB5) 110: H'006000 to H'006FFF (EB6) 111: H'007000 to H'007FFF (EB7) Modes 5 and 6 000: H'100000 to H'100FFF (EB0) 001: H'101000 to H'101FFF (EB1) 010: H'102000 to H'102FFF (EB2) 011: H'103000 to H'103FFF (EB3) 100: H'104000 to H'104FFF (EB4) 101: H'105000 to H'105FFF (EB5) 110: H'106000 to H'106FFF (EB6) 111: H'107000 to H'107FFF (EB7)

17.6 On-Board Programming Modes

In an on-board programming mode, programming, erasing, and verification for the on-chip flash memory can be performed. There are two on-board programming modes: boot mode and user program mode. Table 17.4 shows how to select boot mode. User program mode can be selected by setting the control bits by software. For a diagram that shows mode transitions of flash memory, see figure 17.2.

Table 17.4 Setting On-Board Programming Modes

Mode Setting		MD2	MD1	MD0
Boot mode	Single-chip activation expanded mode with on-chip ROM enabled	0	1	1

17.6.1 Boot Mode

When this LSI enters boot mode, the embedded boot program is started. The boot program transfers the programming control program from the externally connected host to the on-chip RAM via the SCI_1. When the flash memory is all erased, the programming control program is executed.

Table 17.5 shows the boot mode operations between reset end and branching to the programming control program.

1. When the boot program is initiated, the SCI_1 should be set to asynchronous mode, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI_1 bit rate to match that of the host. The transfer format is 8-bit data, 1 stop bit, and no parity. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period.
2. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 17.6.

3. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 17.8, Flash Memory Programming/Erasing.
4. Before branching to the programming control program, the chip terminates transfer operations by the SCI_1 (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of program data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, since the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
5. In boot mode, if flash memory contains data (all data is not 1), all blocks of flash memory are erased. Boot mode is used for the initial programming in the on-board state or for a forcible return when a program that is to be initiated in user program mode was accidentally erased and could not be executed in user program mode.

Notes: 1. In boot mode, a part of the on-chip RAM area (H'FF8000 to H'FF87FF) is used by the boot program. Addresses H'FF8800 to H'FFBFFF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
2. Boot mode can be cleared by a reset. Release the reset by setting the MD pins, after waiting at least 20 states since driving the reset pin low. Boot mode is also cleared when the WDT overflow reset occurs.
3. Do not change the MD pin input levels in boot mode.
4. All interrupts are disabled during programming or erasing of the flash memory.

Table 17.5 Boot Mode Operation

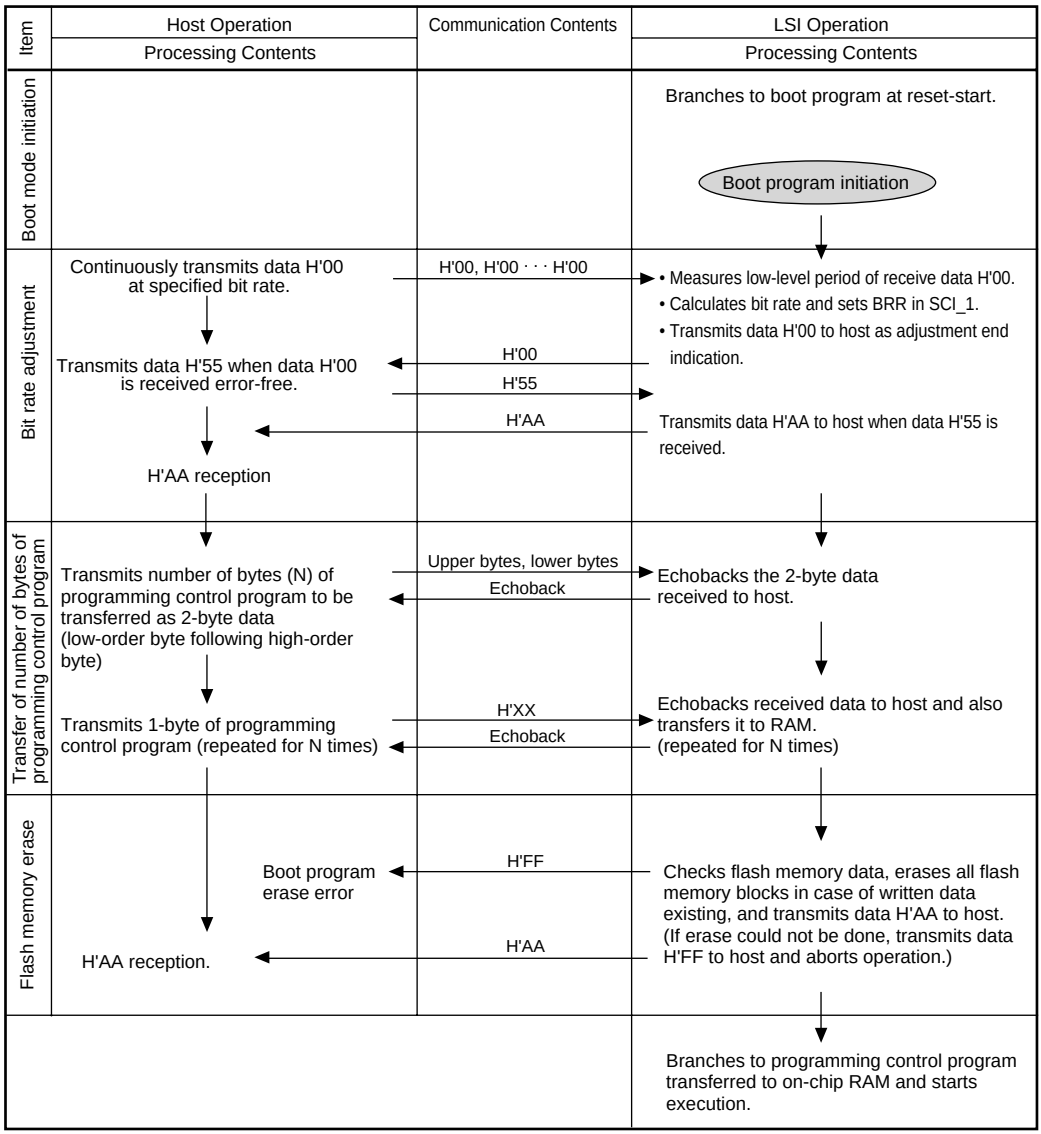


Table 17.6 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate Is Possible

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	8 to 25 MHz
9,600 bps	8 to 25 MHz

17.6.2 User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the program/erase program or a program which provides the program/erase program from external memory. Because the flash memory itself cannot be read during programming/erasing, transfer the program/erase program to on-chip RAM, as like in boot mode. Figure 17.6 shows a sample procedure for programming/erasing in user program mode. Prepare a program/erase program in accordance with the description in section 17.8, Flash Memory Programming/Erasing.

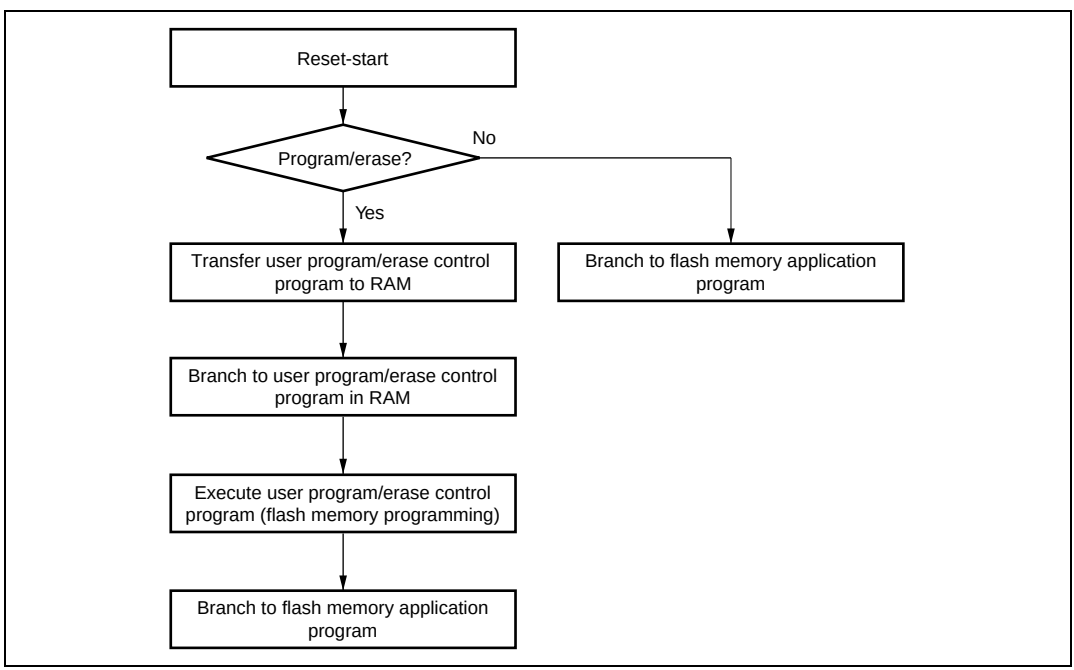


Figure 17.6 Programming/Erasing Flowchart Example in User Program Mode

17.7 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables RAM to be overlapped onto the part of flash memory area so that data to be programmed to flash memory can be emulated in the on-chip RAM in real time. Emulation can be performed in user mode or user program mode. Figure 17.7 shows an example of emulation of real-time flash memory programming.

1. Set RAMER to overlap RAM onto the area for which real-time programming is required.
2. Emulation is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, thus releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

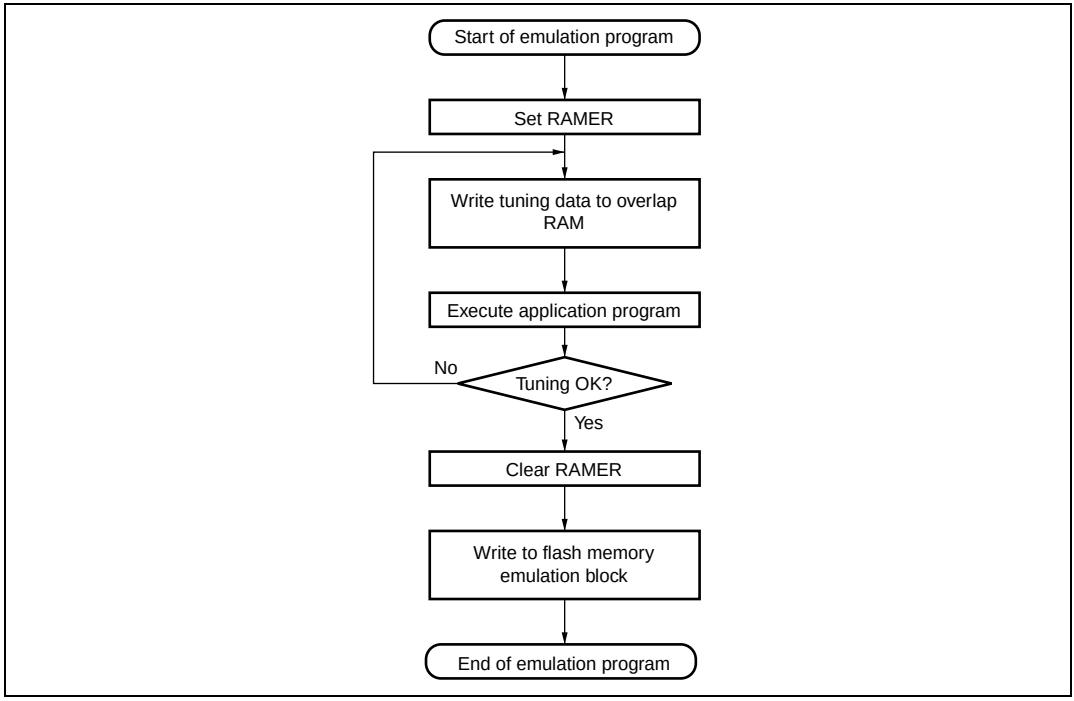


Figure 17.7 Flowchart for Flash Memory Emulation in RAM

Example in which flash memory block is overlapped is shown in figure 17.8.

1. The RAM area to be overlapped is fixed at a 4-kbyte area in the range of H'FFA000 to H'FFAFFF.
2. The flash memory area to overlap is selected by RAMER from a 4-kbyte area among one of the EB0 to EB7 blocks.
3. The overlapped RAM area can be accessed from both the flash memory addresses and RAM addresses.

Note: 1. When the RAMS bit in RAMER is set to 1, program/erase protection is enabled for all flash memory blocks (emulation protection). In this state, setting the P or E bit in FLMCR1 to 1 does not cause a transition to program mode or erase mode.
2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm.
3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

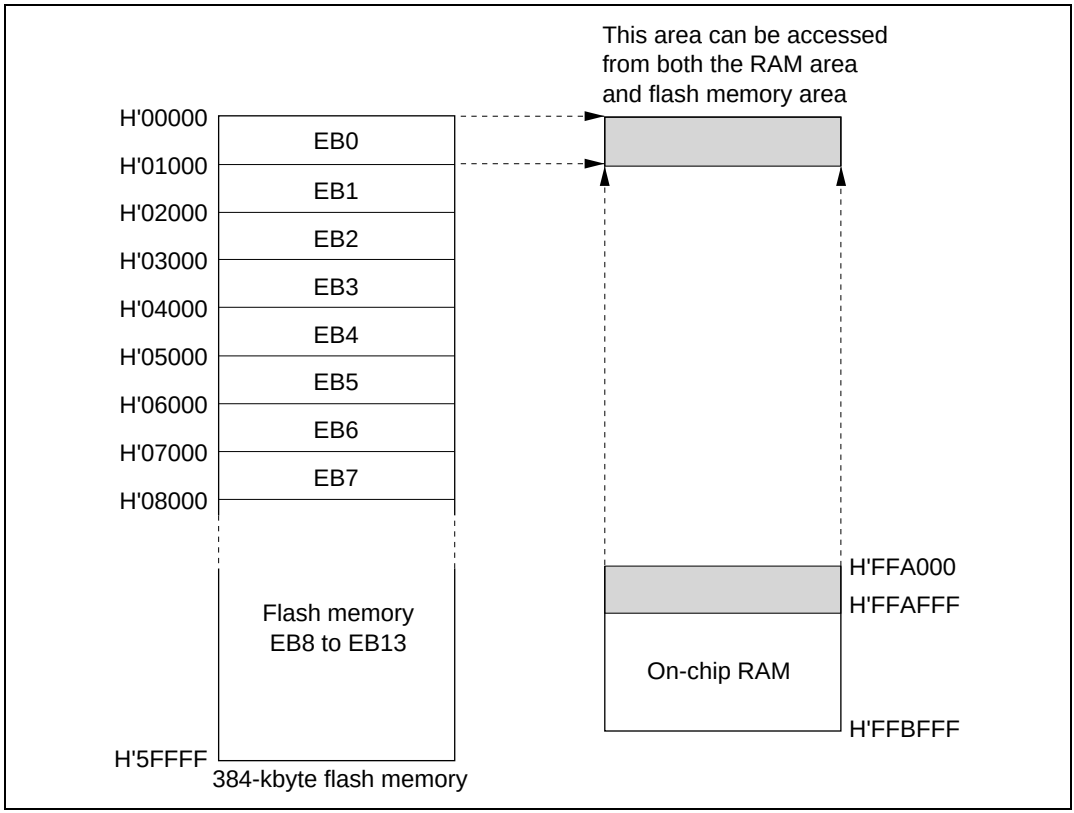


Figure 17.8 Example of RAM Overlap Operation

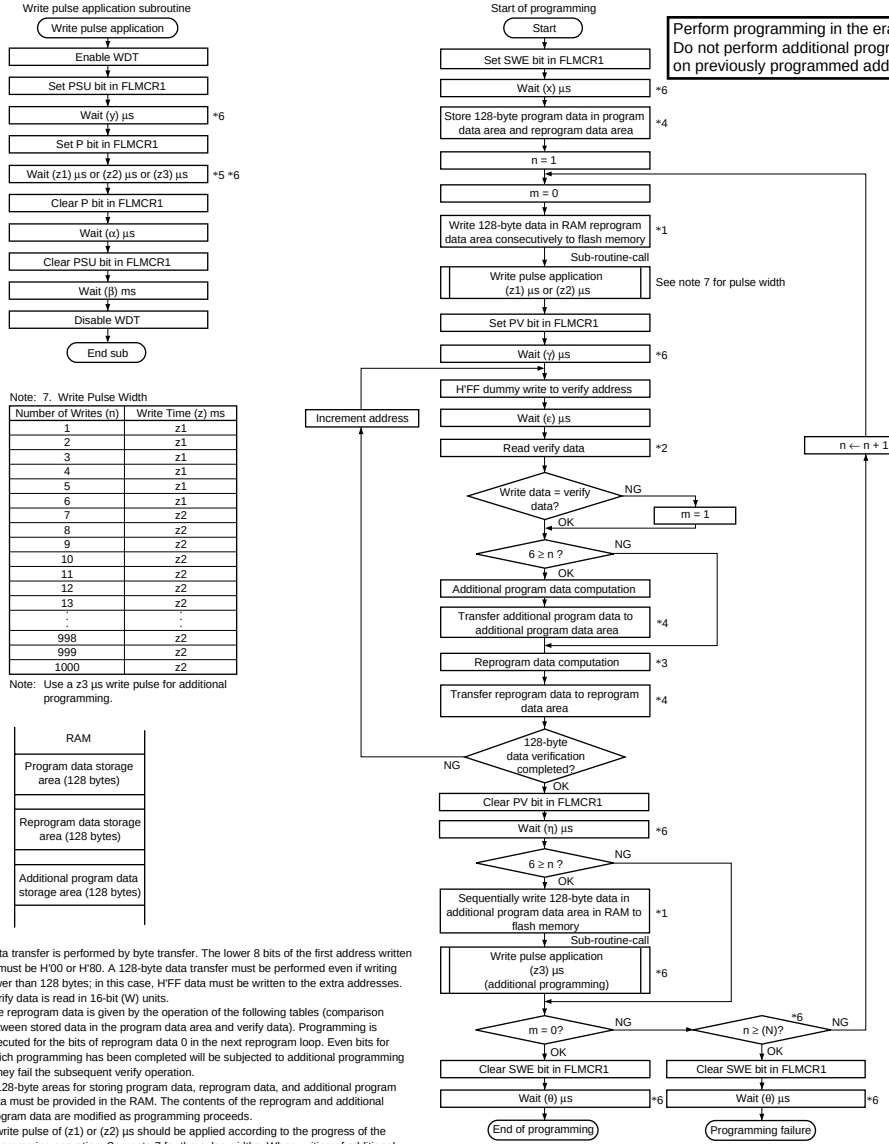
17.8 Flash Memory Programming/Erasing

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 and FLMCR2 setting, the flash memory operates in one of the following four modes: program mode, erase mode, program-verify mode, and erase-verify mode. The programming control program in boot mode and the user program/erase program in user mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 17.8.1, Program/Program-Verify and section 17.8.2, Erase/Erase-Verify, respectively.

17.8.1 Program/Program-Verify

When programming data or programs to the flash memory, the program/program-verify flowchart shown in figure 17.9 should be followed. Performing programming operations according to this flowchart will enable data or programs to be programmed to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: a 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation and additional programming data computation according to figure 17.9.
4. Consecutively transfer 128 bytes of data in byte units from the programming data area, reprogramming data area, or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Figure 17.9 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than ($y + z2 + \alpha + \beta$) µs as the WDT overflow period.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B’00. Verify data can be read in words from the address to which a dummy write was performed.
8. The maximum number of repetitions of the program/program-verify sequence to the same bit (N) must not be exceeded.



Note: 7. Write Pulse Width

Number of Writes (n)	Write Time (z) ms
1	z1
2	z1
3	z1
4	z1
5	z1
6	z1
7	z2
8	z2
9	z2
10	z2
11	z2
12	z2
13	z2
:	:
998	z2
999	z2
1000	z2

Note: Use a z3 µs write pulse for additional programming.

Notes:
1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
2. Verify data is read in 16-bit (W) units.
3. The reprogram data is given by the operation of the following tables (comparison between stored data in the program data area and verify data). Programming is executed for the bits of reprogram data 0 in the next reprogram loop. Even bits for which programming has been completed will be subjected to additional programming if they fail the subsequent verify operation.
4. A 128-byte areas for storing program data, reprogram data, and additional program data must be provided in the RAM. The contents of the reprogram and additional program data are modified as programming proceeds.
5. A write pulse of (z1) or (z2) µs should be applied according to the progress of the programming operation. See note 7 for the pulse widths. When writing of additional-programming data is executed, a (z3) µs write pulse should be applied.
   Reprogram data X' means reprogram data when the write pulse is applied.
6. For the values of x, y, z1, z2, z3, α, β, γ, ε, η, θ, and N, see section 21.1.6, Flash Memory Characteristics.

Program Data Operation Chart

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
	1	0	Programming incomplete; reprogram
1	0	1	
	1		Still in erased state; no action

Additional Program Data Operation Chart

Reprogram Data (X')	Verify Data (V)	Additional Program Data (Y)	Comments
0	0	0	Additional programming executed
	1	1	Additional programming not executed
1	0		Additional programming not executed
	1		Additional programming not executed

Figure 17.9 Program/Program-Verify Flowchart

17.8.2 Erase/Erase-Verify

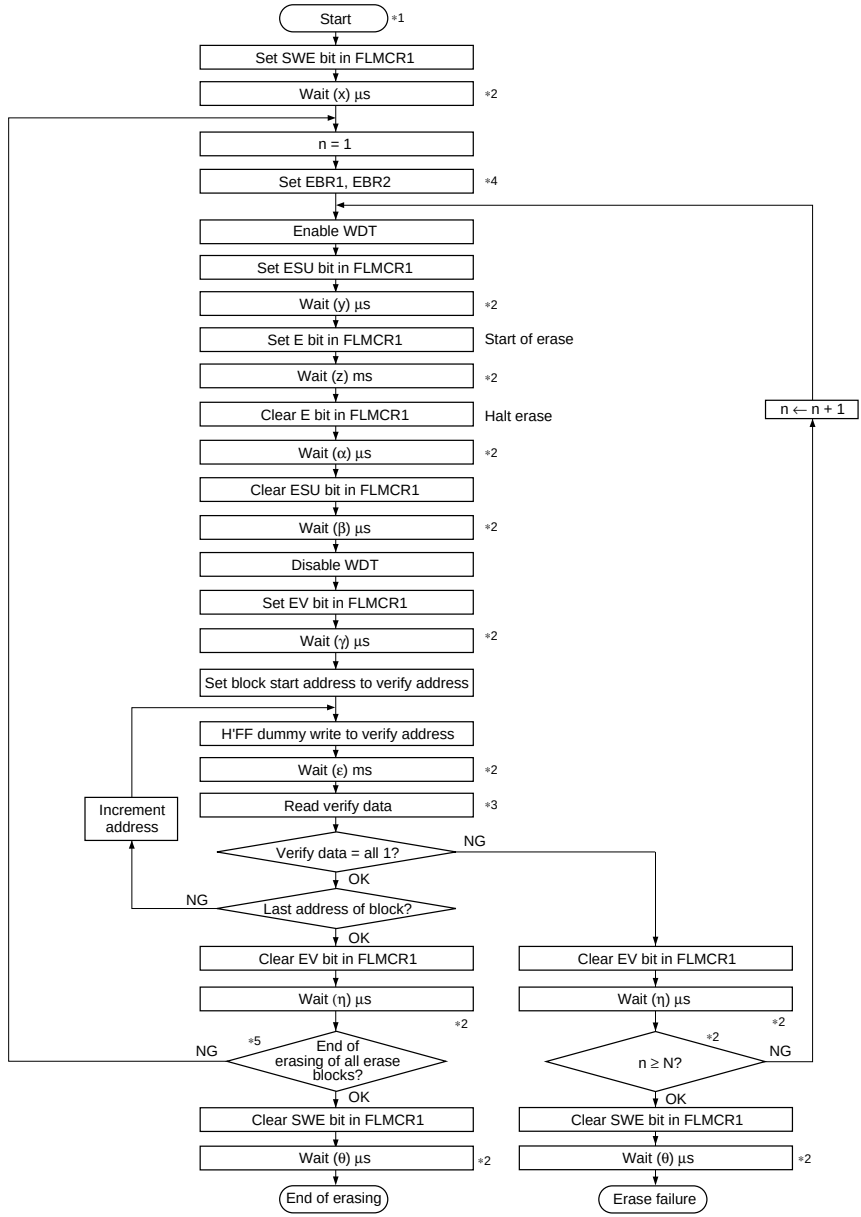
When erasing flash memory, the erase/erase-verify flowchart shown in figure 17.10 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block registers (EBR1 and EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than ($y + z + \alpha + \beta$) ms as the WDT overflow period.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B’00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is unerased, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence (N) must not be exceeded.

17.8.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI input, are disabled when flash memory is being programmed or erased, and while the boot program is executing in boot mode. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. If the interrupt exception handling is started when the vector address has not been programmed yet or the flash memory is being programmed or erased, the vector would not be read correctly, possibly resulting in CPU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.



Notes: 1. Prewriting (setting erase block data to all 0) is not necessary.
2. The values of x, y, z, α, β, γ, ε, η, θ, and N are shown in section 21.1.6, Flash Memory Characteristics.
3. Verify data is read in 16-bit (W) units.
4. Set only one bit in EBR1or EBR2. More than one bit cannot be set.
5. Erasing is performed in block units. To erase a number of blocks, the individual blocks must be erased sequentially.

Figure 17.10 Erase/Erase-Verify Flowchart

17.9 Program/Erase Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

17.9.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset (including an overflow reset by the WDT) or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2) are initialized. In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.

17.9.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1 (this operation must be executed in the on-chip RAM or external memory). When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1) and erase block register 2 (EBR2), erase protection can be set for individual blocks. When EBR1 and EBR2 are set to H'00, erase protection is set for all blocks.

17.9.3 Error Protection

In error protection, an error is detected when the CPU's runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is forcibly aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the FLER bit in FLMCR2 is set to 1, and the error protection state is entered.

- When flash memory is read during programming/erasing (including a vector read or instruction fetch)
- When an exception handling (excluding a reset) is started during programming/erasing
- When a SLEEP instruction is executed during programming/erasing
- When the CPU releases the bus mastership during programming/erasing

The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is forcibly aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode. The error protection state can be canceled by a power-on reset or in hardware standby mode.

17.10 Programmer Mode

In programmer mode, a PROM programmer can perform programming/erasing via a socket adapter, just like for a discrete flash memory. Use a PROM programmer which supports the Renesas Technology 512-kbyte flash memory on-chip MCU device type (FZTAT512V3A). A 12-MHz input clock is needed.

17.11 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode
  The flash memory can be read.
- Standby mode
  All flash memory circuits are halted.

Table 17.7 shows the correspondence between the operating modes of this LSI and the flash memory. When the flash memory returns to normal operation from a standby state, a power supply circuit stabilization period is needed. When the flash memory returns to its normal operating state, bits STS3 to STS0 in SBYCR must be set to provide a wait time of at least 100 µs, even when the external clock is being used.

Table 17.7 Flash Memory Operating States

Operating Mode	Flash Memory Operating State
Active mode	Normal operating state
Sleep mode	Normal operating state
Standby mode	Standby state

17.12 Usage Notes

Precautions concerning the use of on-board programming mode, the RAM emulation function, and programmer mode are summarized below.

1. Use the specified voltages and timing for programming and erasing.

   Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Renesas Technology microcomputer device type with 512-kbyte on-chip flash memory (FZTAT512V3A).

   Do not select the HN27C4096 setting for the PROM programmer, and only use the specified socket adapter.

2. Reset the flash memory before turning on/off the power.

   When applying or disconnecting Vcc power, fix the $\overline{\text{RES}}$ pin low and place the flash memory in the hardware protection state. The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

3. Use the recommended algorithm when programming and erasing flash memory.

   The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

4. Do not set or clear the SWE bit during execution of a program in flash memory.

   Wait for at least 100 µs after clearing the SWE bit before executing a program or reading data in flash memory.

   When the SWE bit is set, data in flash memory can be rewritten. When the SWE bit is set to 1, data in flash memory can be read only in program-verify/erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying. Similarly, when using the RAM emulation function, the SWE bit must be cleared before executing a program or reading data in flash memory.

   However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

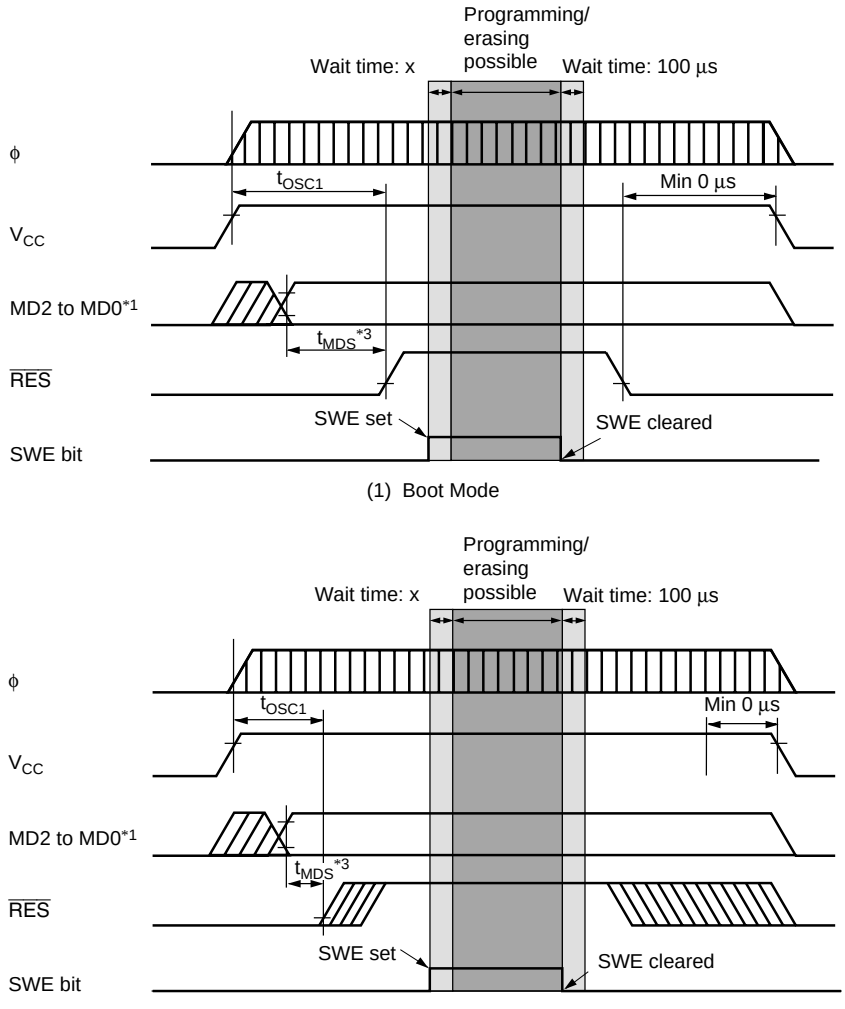
5. Do not use interrupts while flash memory is being programmed or erased.

   All interrupt requests, including NMI, should be disabled during programming/erasing the flash memory to give priority to program/erase operations.

6. Do not perform additional programming. Erase the memory before reprogramming.

   In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

7. Before programming, check that the chip is correctly mounted in the PROM programmer.
   Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.
8. Do not touch the socket adapter or chip during programming.
   Touching either of these can cause contact faults and write errors.
9. Apply the reset signal after the SWE, bit is cleared during its operation.
   The reset signal is applied at least 100 µs after the SWE bit has been cleared.

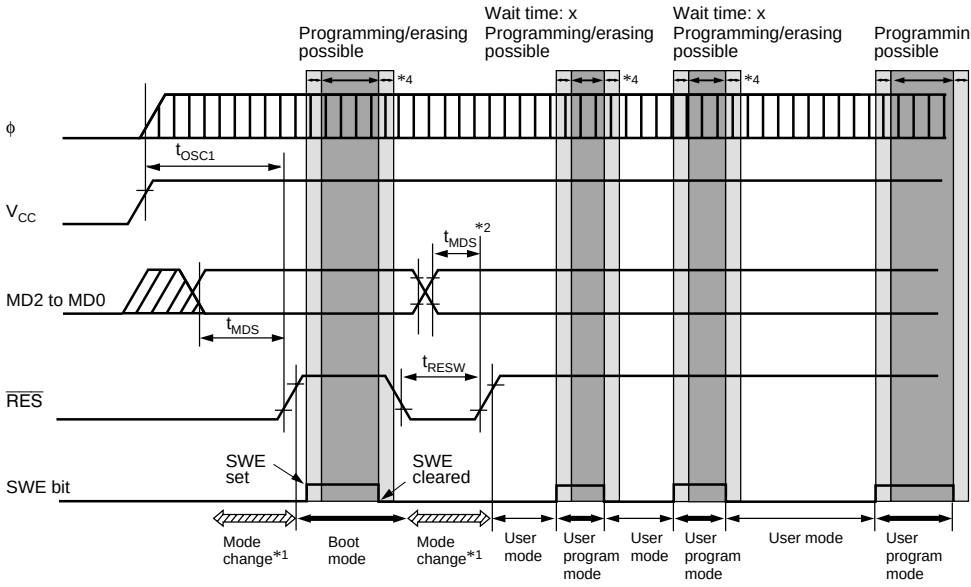


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
2. See section 21.1.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns

Figure 17.11 Power-On/Off Timing



Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of RES input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), and therefore these pins should not be used as output signals during this time.
2. When making a transition from boot mode to another mode, a mode programming setup time t_{MDS} (min) of 200 ns is necessary with respect to $\overline{RES}$ clearance timing.
3. See section 21.1.6, Flash Memory Characteristics.
4. Wait time: 100 ms

Figure 17.12 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

Section 18 Clock Pulse Generator

This LSI has an on-chip clock pulse generator (CPG) that generates the system clock (ϕ) and internal clocks.

The clock pulse generator consists of an oscillator circuit, PLL circuit, and divider.

Figure 18.1 shows a block diagram of the clock pulse generator.

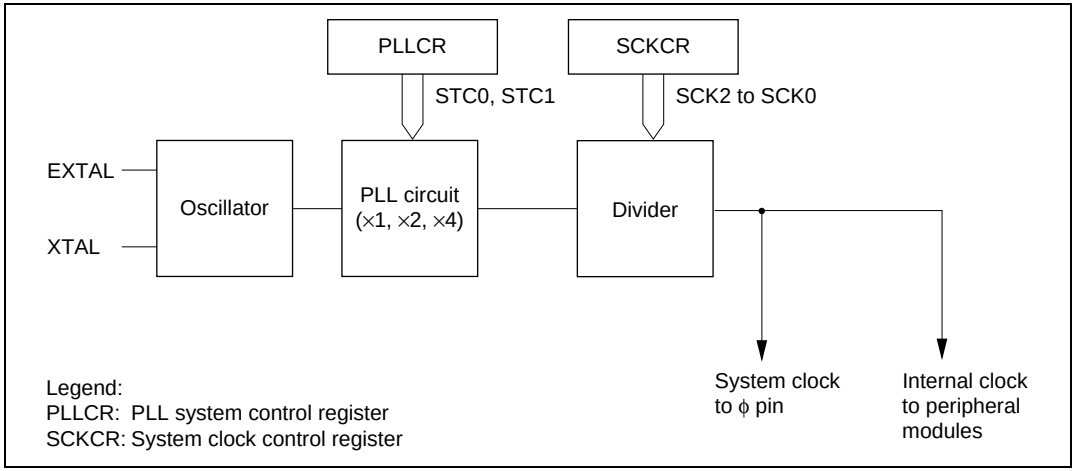


Figure 18.1 Block Diagram of Clock Pulse Generator

The frequency can be changed by means of the PLL circuit. Frequency changes are made by software by means of settings in the PLL control register (PLLCR) and the system clock control register (SCKCR).

18.1 Register Descriptions

The clock pulse generator has the following registers.

- System clock control register (SCKCR)
- PLL control register (PLLCR)

18.1.1 System Clock Control Register (SCKCR)

SCKCR controls φ clock output and selects operation when the frequency multiplication factor used by the PLL circuit is changed, and the division ratio used by the divider.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	φ Clock Output Disable Controls φ output. Normal Operation 0: φ output 1: Fixed high Sleep Mode 0: φ output 1: Fixed high Software Standby Mode 0: Fixed high 1: Fixed high Hardware Standby Mode 0: High impedance 1: High impedance All module clock stop mode 0: φ output 1: Fixed high
6	—	0	R/W	Reserved This bit can be read from or written to. However, 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	Frequency Multiplication Factor Switching Mode Select Selects the operation when the PLL circuit frequency multiplication factor is changed. 0: Specified multiplication factor is valid after transition to software standby mode 1: Specified multiplication factor is valid immediately after STC1 and STC0 bits are rewritten

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 division ratio.
0	SCK0	0	R/W	000: 1/1
				001: 1/2
				010: 1/4
				011: 1/8
				100: 1/16
				101: 1/32
				11X: Setting prohibited

X: Don't care

18.1.2 PLL Control Register (PLLCR)

PLLCR sets the frequency multiplication factor used by the PLL circuit.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	—	0	R/W	Reserved This bit can be read from or written to. However, the write value should always be 0.
2	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
1	STC1	0	R/W	Frequency Multiplication Factor
0	STC0	0	R/W	The STC bits specify the frequency multiplication factor used by the PLL circuit.
				00: × 1
				01: × 2
				10: × 4
				11: Setting prohibited

18.2 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

18.2.1 Connecting a Crystal resonator

A crystal resonator can be connected as shown in the example in figure 18.2. Select the damping resistance R_d according to table 18.1. An AT-cut parallel-resonance type should be used.

Figure 18.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 18.2.

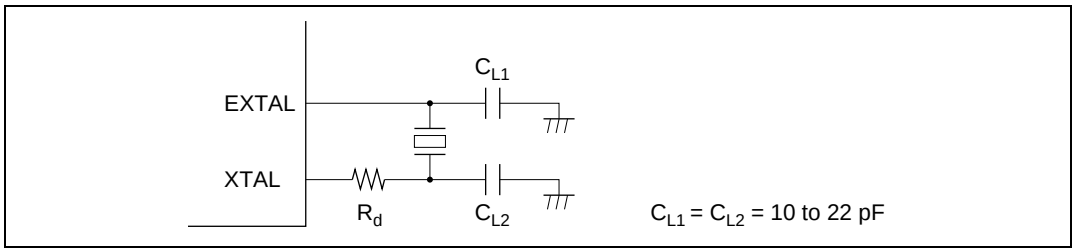


Figure 18.2 Connection of Crystal Resonator (Example)

Table 18.1 Damping Resistance Value

Frequency (MHz)	8	12	16	20	25
R_d (Ω)	200	0	0	0	0

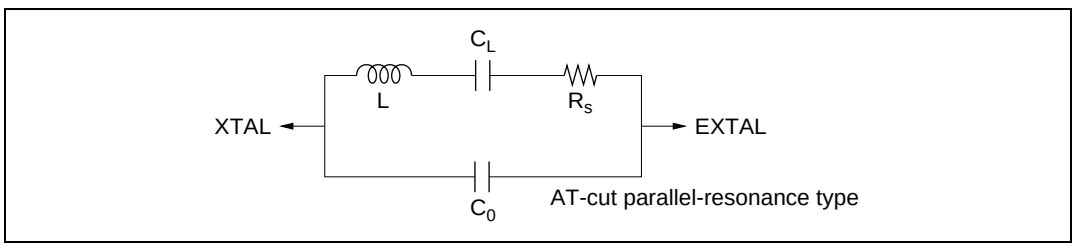


Figure 18.3 Crystal Resonator Equivalent Circuit

Table 18.2 Crystal Resonator Characteristics

Frequency (MHz)	8	12	16	20	25
R_S max (Ω)	80	60	50	40	40
C_0 max (pF)	7	7	7	7	7

18.2.2 External Clock Input

An external clock signal can be input as shown in the examples in figure 18.4. If the XTAL pin is left open, make sure that parasitic capacitance is no more than 10 pF. When the counter clock is input to the XTAL pin, make sure that the external clock is held high in standby mode.

Table 18.3 shows the input conditions for the external clock

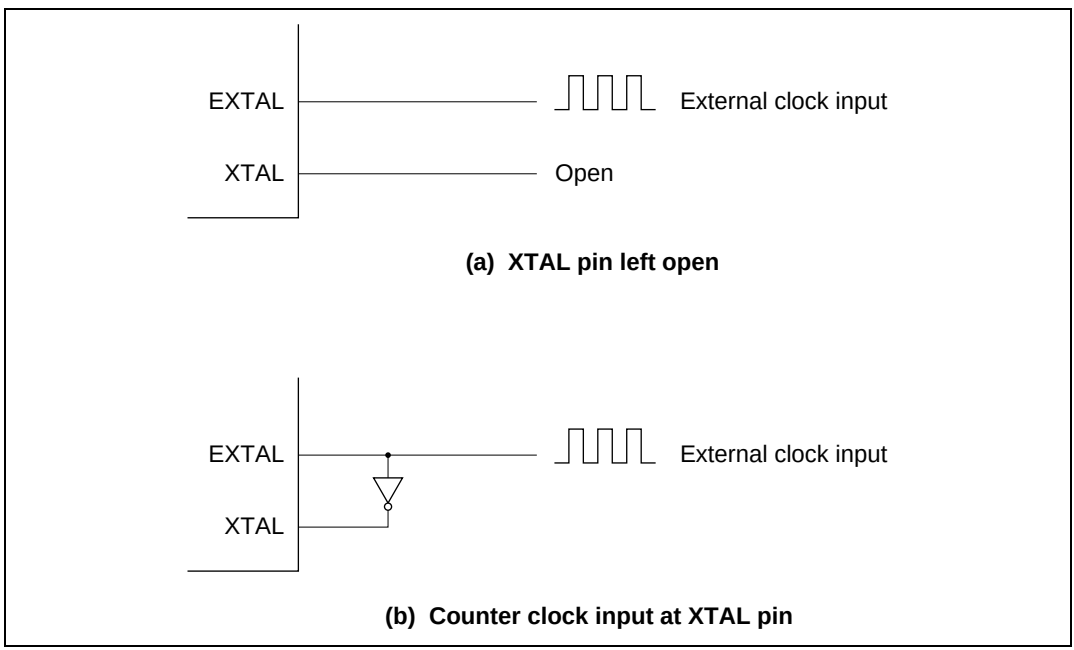


Figure 18.4 External Clock Input (Examples)

Table 18.3 External Clock Input Conditions

		V_{cc} = 3.0 V to 3.6 V			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
External clock input low pulse width	t_{EXL}	15	—	ns	Figure 18.5
External clock input high pulse width	t_{EXH}	15	—	ns	
External clock rise time	t_{EXr}	—	5	ns	
External clock fall time	t_{EXf}	—	5	ns	
Clock low pulse width	t_{CL}	0.4	0.6	t_{cyc}	
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	

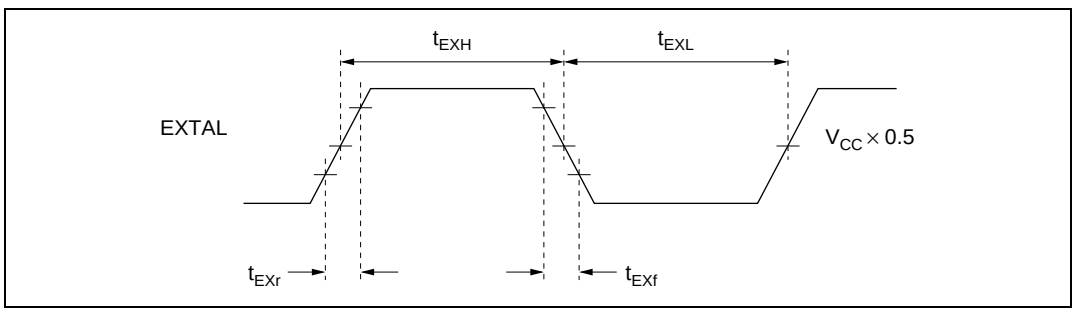


Figure 18.5 External Clock Input Timing

18.3 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set with the STC1 and the STC0 bits in PLLCR. The phase of the rising edge of the internal clock is controlled so as to match that of the rising edge of the EXTAL pin.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0, the setting becomes valid after a transition to software standby mode. The transition time count is performed in accordance with the setting of bits STS3 to STS0 in SBYCR. For details on SBYCR, refer to section 19.1.1, Standby Control Register (SBYCR).

1. The initial PLL circuit multiplication factor is 1.
2. A value is set in bits STS3 to STS0 to give the specified transition time.
3. The target value is set in bits STC1 and STC0, and a transition is made to software standby mode.
4. The clock pulse generator stops and the value set in STC1 and STC0 becomes valid.
5. Software standby mode is cleared, and a transition time is secured in accordance with the setting in STS3 to STS0.
6. After the set transition time has elapsed, this LSI resumes operation using the target multiplication factor.

When STCS = 1, this LSI operates using the new multiplication factor immediately after bits STC1 and STC0 are rewritten.

18.4 Frequency Divider

The frequency divider divides the PLL output clock to generate a 1/2, 1/4, 1/8, 1/16, or 1/32 clock.

18.5 Usage Notes

18.5.1 Notes on Clock Pulse Generator

1. The following points should be noted since the frequency of ϕ changes according to the setting of SCKCR and PLLCR.

   Select the clock division ratio that is within the operation guaranteed range of clock cycle time tcyc shown in the AC timing of Electrical Characteristics. In other words, the range of ϕ must be specified from 8 MHz (min) to 33 MHz (max); outside of this range must be prevented.
2. All the on-chip peripheral modules operate on the ϕ. Therefore, note that the time processing of modules such as a timer and SCI differ before and after changing the clock division ratio. In addition, wait time for clearing software standby mode differs by changing the clock division ratio. See the description, Setting Oscillation Stabilization Time after Clearing Software Standby Mode in section 19.2.3, Software Standby Mode, for details.
3. Note that the frequency of ϕ will be changed when setting SCKCR or PLLCR while executing the external bus cycle with the write-data-buffer function.

18.5.2 Notes on Resonator

Since various characteristics related to the 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 parameters for the oscillation circuit will depend on the floating capacitance of the resonator and the user board, the parameters 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 resonator pin.

18.5.3 Notes on Board Designs

When using the crystal resonator, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 18.6.

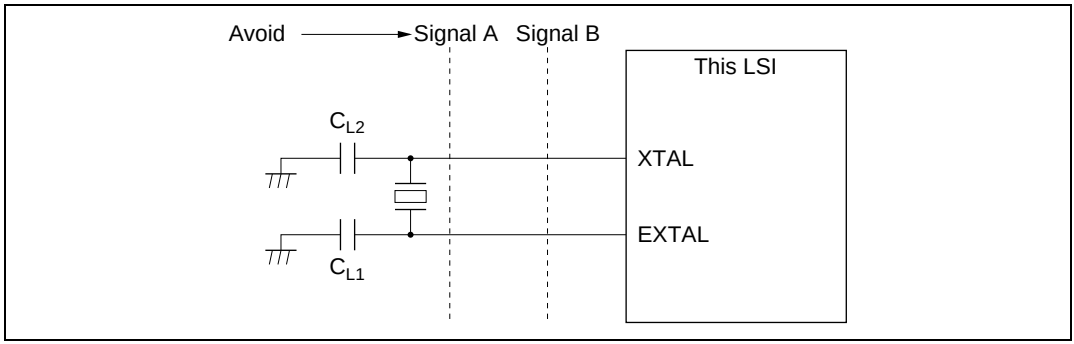


Figure 18.6 Note on Oscillator Board Design

Figure 18.7 shows the external circuitry recommended for the PLL circuit. Separate PLLVcc and PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure to insert bypass capacitors CPB and CB close to the pins.

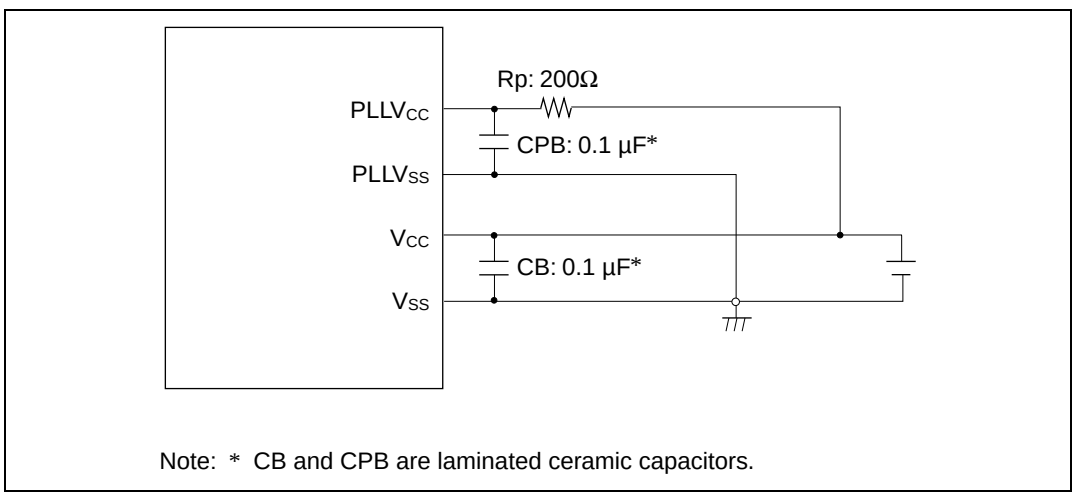


Figure 18.7 Recommended External Circuitry for PLL Circuit

Section 19 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 dissipation 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 high-speed mode and six power down modes:

- Clock division mode
- Sleep mode
- Module stop mode
- All module clock stop mode
- Software standby mode
- Hardware standby mode

Sleep mode is a CPU state, clock division mode is a CPU and bus master state, and module stop mode is an on-chip peripheral function (including bus masters other than the CPU) state. A combination of these modes can be set.

After a reset, this LSI is in high-speed mode.

Table 19.1 shows the internal states of this LSI in each mode. Figure 19.1 shows the mode transition diagram.

Table 19.1 Operating Modes

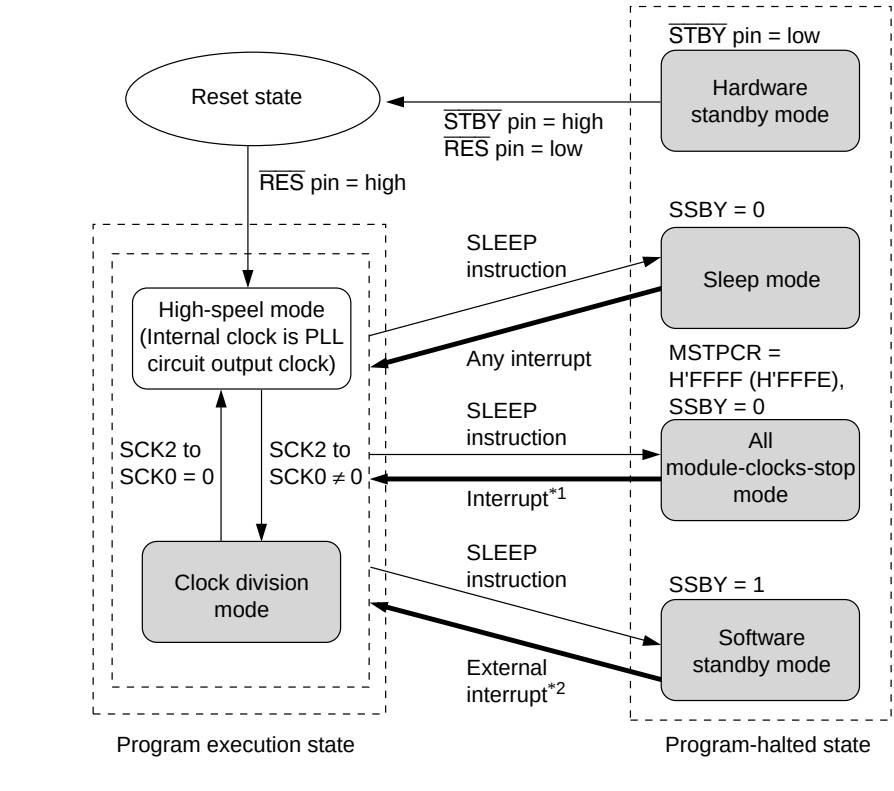
Operating State		High Speed Mode	Clock Division Mode	Sleep Mode	Module Stop Mode	All Module Clock Stop Mode	Software Standby Mode	Hardware Standby Mode
Clock pulse generator		Functions	Functions	Functions	Functions	Functions	Halted	Halted
CPU	Instruction execution	Functions	Functions	Halted	Functions	Halted	Halted	Halted
	Register			Retained			Retained	Undefined
External interrupts	NMI	Functions	Functions	Functions	Functions	Functions	Functions	Halted
	IRQ0 to 15							
Peripheral functions	WDT	Functions	Functions	Functions	Functions	Functions	Halted (Retained)	Halted (Reset)
	TMR	Functions	Functions	Functions	Halted (Retained)	Functions/ Halted (Retained)*	Halted (Retained)	Halted (Reset)
	DTC	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	TPU	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	PPG	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	D/A	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	A/D	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Reset)	Halted (Reset)
	SCI	Functions	Functions	Functions	Halted (Reset)	Halted (Reset)	Halted (Reset)	Halted (Reset)
	RAM	Functions	Functions	Functions	Functions	Functions	Retained	Retained
	I/O	Functions	Functions	Functions	Functions	Retained	Retained	High impedance

Notes: "Halted (Retained)" in the table means that internal register values are retained and internal operations are suspended.

"Halted (Reset)" in the table 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).

* The active or halted state can be selected by means of the MSTP0 bit in MSTPCR.



Notes: 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, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ is driven low.
From any state except hardware standby mode, a transition to the reset state occurs when $\overline{\text{RES}}$ is driven low.

1. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$, 8-bit timer interrupts, watchdog timer interrupts.
   (8-bit timer interrupts are valid when MSTP0 = 0.)
2. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$
   (IRQ0 to IRQ7 are valid when the corresponding bit in SSIER is 1.)

Figure 19.1 Mode Transitions

19.1 Register Descriptions

The registers relating to the power-down mode are shown below. For details on the system clock control register (SCKCR), refer to section 18.1.1, System Clock Control Register (SCKCR).

- System clock control register (SCKCR)
- Standby control register (SBYCR)
- Module stop control register H (MSTPCRH)
- Module stop control register L (MSTPCRL)

19.1.1 Standby Control Register (SBYCR)

SBYCR performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit specifies the transition mode after executing the SLEEP instruction 0: Shifts to sleep mode after the SLEEP instruction is executed 1: Shifts to software standby mode after the SLEEP instruction is executed This bit does not change when clearing the software standby mode by using external interrupts and shifting to normal operation. This bit should be written 0 when clearing.
6	OPE	1	R/W	Output Port Enable Specifies whether the output of the address bus and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{UCAS}$, $\overline{LCAS}$) is retained or set to the high-impedance state in software standby mode. 0: In software standby mode, address bus and bus control signals are high-impedance 1: In software standby mode, address bus and bus control signals retain output state

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	All 0	—	Reserved These bits are always read as 0. The initial value should not be changed.
3	STS3	1	R/W	Standby Timer Select 3 to 0 These bits select the time the MCU waits for the clock to stabilize when software standby mode is cleared by an external interrupt. With crystal oscillation, refer to table 19.2 and make a selection according to the operating frequency so that the standby time is at least the oscillation stabilization time. With an external clock, a PLL circuit stabilization time is necessary. Refer to table 19.2 to set the wait time. 0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: Setting prohibited 0100: Setting prohibited 0101: Standby time = 64 states 0110: Standby time = 512 states 0111: Standby time = 1024 states 1000: Standby time = 2048 states 1001: Standby time = 4096 states 1010: Standby time = 16384 states 1011: Standby time = 32768 states 1100: Standby time = 65536 states 1101: Standby time = 131072 states 1110: Standby time = 262144 states 1111: Standby time = 524288 states
2	STS2	1	R/W	
1	STS1	1	R/W	
0	STS0	1	R/W	

19.1.2 Module Stop Control Registers H and L (MSTPCRH, MSTPCRL)

MSTPCR performs module stop mode control.

Setting a bit to 1, the corresponding module enters module stop mode, while clearing the bit to 0 clears the module stop mode.

MSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15	ACSE	0	R/W	All-Module-Clocks-Stop Mode Enable Enables or disables all-module-clocks-stop mode, in which, when the CPU executes a SLEEP instruction after module stop mode has been set for all the on-chip peripheral functions controlled by MSTPCR or the on-chip peripheral functions except the TMR. 0: All-module-clocks-stop mode disabled 1: All-module-clocks-stop mode enabled
14	MSTP14	0	R/W	—
13	MSTP13	0	R/W	—
12	MSTP12	0	R/W	Data transfer controller (DTC)
11	MSTP11	1	R/W	16-bit timer-pulse unit (TPU)
10	MSTP10	1	R/W	Programmable pulse generator (PPG)
9	MSTP9	1	R/W	D/A converter (channels 0 and 1)
8	MSTP8	1	R/W	D/A converter (channels 2 and 3)

MSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP7	1	R/W	—
6	MSTP6	1	R/W	A/D converter
5	MSTP5	1	R/W	—
4	MSTP4	1	R/W	—
3	MSTP3	1	R/W	Serial communication interface 2 (SCI_2)
2	MSTP2	1	R/W	Serial communication interface 1 (SCI_1)
1	MSTP1	1	R/W	Serial communication interface 0 (SCI_0)
0	MSTP0	1	R/W	8-bit timer (TMR)

19.2 Operation

19.2.1 Clock Division Mode

When bits SCK2 to SCK0 in SCKCR are set to a value from 001 to 101, a transition is made to clock division mode at the end of the bus cycle. In clock division mode, the CPU, bus masters, and on-chip peripheral functions all operate on the operating clock (1/2, 1/4, 1/8, 1/16, or 1/32) specified by bits SCK2 to SCK0.

Clock division mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode at the end of the bus cycle, and clock division mode is cleared.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is cleared to 0, the chip enters sleep mode. When sleep mode is cleared by an interrupt, clock division mode is restored.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the chip enters software standby mode. When software standby mode is cleared by an external interrupt, clock division mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, the reset state is entered and clock division mode is cleared. The same applies to a reset caused by watchdog timer overflow.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

19.2.2 Sleep Mode

Transition to Sleep Mode: When the SLEEP instruction is executed when the SSBY bit is 0 in SBYCR, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral functions do not stop.

Exiting Sleep Mode: Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pins.

- Exiting Sleep Mode by Interrupts:
  When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.
- Exiting Sleep Mode by $\overline{\text{RES}}$ pin:
  Setting the $\overline{\text{RES}}$ pin level low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ pin high starts the CPU performing reset exception processing.
- Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin:
  When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

19.2.3 Software Standby Mode

Transition to Software Standby Mode: If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, software standby mode is entered. In this mode, the CPU, on-chip peripheral functions, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip peripheral functions other than the SCI and A/D converter, and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state or retain the output state can be specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

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 or $\overline{\text{STBY}}$ pin. Setting the SSI bit in SSIER to 1 enables $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$ to be used as software standby mode clearing sources.

- 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 LSI, 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 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

  When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ 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.

Setting Oscillation Stabilization Time after Clearing Software Standby Mode: Bits STS3 to STS0 in SBYCR should be set as described below.

- Using a Crystal Oscillator

  Set bits STS3 to STS0 so that the standby time is more than the oscillation stabilization time.

  Table 19.2 shows the standby times for operating frequencies and settings of bits STS3 to STS0.
- Using an External Clock

  A PLL circuit stabilization time is necessary. Refer to table 19.2 to set the wait time.

Table 19.2 Oscillation Stabilization Time Settings

					ϕ^* [MHz]						
STS3	STS2	STS1	STS0	Standby Time	33	25	20	13	10	8	Unit
0	0	0	0	Reserved	—	—	—	—	—	—	µs
			1	Reserved	—	—	—	—	—	—	
		1	0	Reserved	—	—	—	—	—	—	
			1	Reserved	—	—	—	—	—	—	
	1	0	0	Reserved	—	—	—	—	—	—	
			1	64	1.9	2.6	3.2	4.9	6.4	8.0	
		1	0	512	15.5	20.5	25.6	39.4	51.2	64.0	
			1	1024	31.0	41.0	51.2	78.8	102.4	128.0	
1	0	0	0	2048	62.1	81.9	102.4	157.5	204.8	256.0	
			1	4096	0.12	0.16	0.20	0.32	0.41	0.51	ms
		1	0	16384	0.50	0.66	0.82	1.26	1.64	2.05	
			1	32765	0.99	1.31	1.64	2.52	3.28	4.10	
	1	0	0	65536	1.99	2.62	3.28	5.04	6.55	8.19	
			1	131072	3.97	5.24	6.55	10.08	13.11	16.38	
		1	0	262144	7.94	10.49	13.11	20.16	26.21	32.77	
			1	524288	15.89	20.97	26.21	40.33	52.43	65.54	

☐ : Recommended time setting

Note: * ϕ is the frequency divider output.

Software Standby Mode Application Example: Figure 19.2 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 INTCR 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.

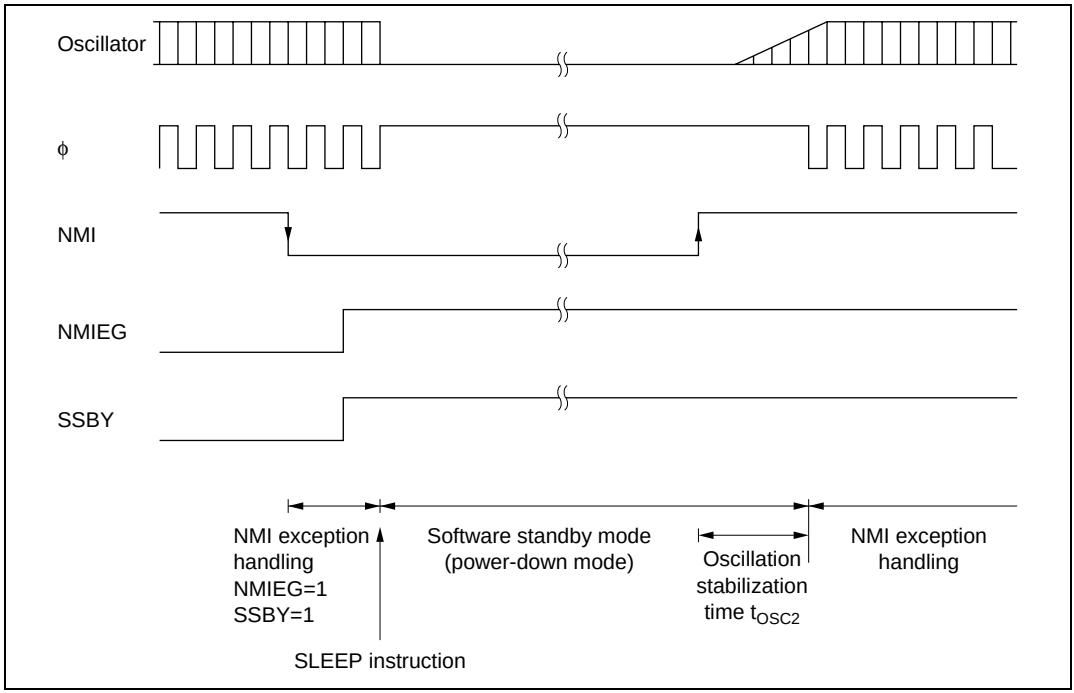


Figure 19.2 Software Standby Mode Application Example

19.2.4 Hardware Standby Mode

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 dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low. Do not change the state of the mode pins (MD2 to MD0) while this LSI is in hardware standby mode.

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 stabilizes (for details on the oscillation stabilization time, refer to table 19.2). 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.

Hardware Standby Mode Timing: Figure 19.3 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 stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

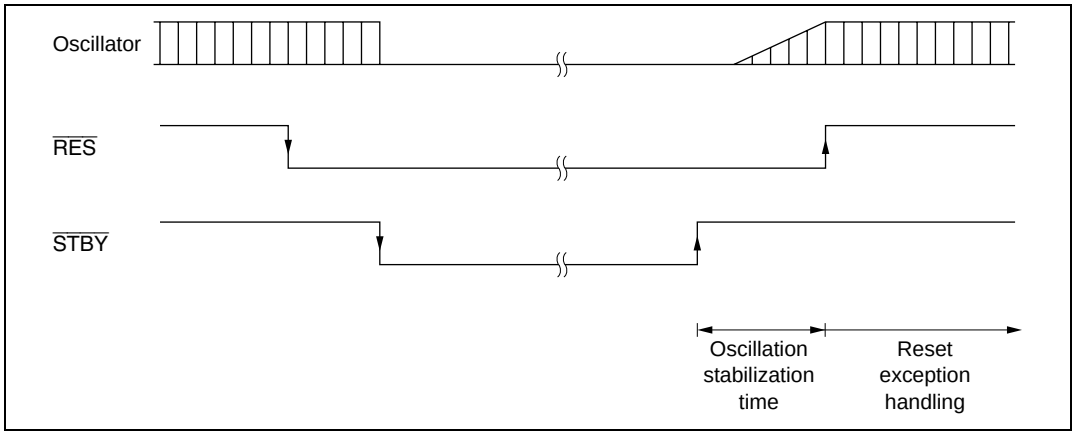


Figure 19.3 Hardware Standby Mode Timing

19.2.5 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 SCI are retained.

After reset clearance, all modules other than the DTC are in module stop mode.

The module registers which are set in module stop mode cannot be read or written to.

19.2.6 All-Module-Clocks-Stop Mode

When the ACSE bit in MSTPCRH is set to 1 and module stop mode is set for all the on-chip peripheral functions controlled by MSTPCR (MSTPCR = H'FFFF), or for all the on-chip peripheral functions except the 8-bit timer (MSTPCR = H'FFFE), executing a SLEEP instruction while the SSBY bit in SBYCR is cleared to 0 will cause all the on-chip peripheral functions (except the 8-bit timer and watchdog timer), the bus controller, and the I/O ports to stop operating, and a transition to be made to all-module-clocks-stop mode, at the end of the bus cycle.

Operation or halting of the 8-bit timer can be selected by means of the MSTP0 bit.

All-module-clocks-stop mode is cleared by an external interrupt (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$ pins), $\overline{\text{RES}}$ pin input, or an internal interrupt (8-bit timer, watchdog timer), and the CPU returns to the normal program execution state via the exception handling state. All-module-clocks-stop mode is not cleared if interrupts are disabled, if interrupts other than NMI are masked by the CPU, or if the relevant interrupt is designated as a DTC activation source.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

19.3 φ Clock Output Control

Output of the φ clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the φ clock stops at the end of the bus cycle, and φ output goes high. φ clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, φ clock output is disabled and input port mode is set. Table 19.3 shows the state of the φ pin in each processing state.

Table 19.3 φ Pin State in Each Processing State

Register Setting						
DDR	PSTOP	Normal Operating State	Sleep Mode	Software Standby Mode	Hardware Standby Mode	All-Module-Clocks-Stop Mode
0	X	High impedance	High impedance	High impedance	High impedance	High impedance
1	0	φ output	φ output	Fixed high	High impedance	φ output
1	1	Fixed high	Fixed high	Fixed high	High impedance	Fixed high

19.4 Usage Notes

19.4.1 I/O Port Status

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

19.4.2 Current Dissipation during Oscillation Stabilization Standby Period

Current dissipation increases during the oscillation stabilization standby period.

19.4.3 DTC Module Stop

Depending on the operating status of the DTC, the MSTP14 to MSTP12 bits 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 7, Data Transfer Controller (DTC).

19.4.4 On-Chip Peripheral Module Interrupts

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.

19.4.5 Writing to MSTPCR

MSTPCR should only be written to by the CPU.

Section 20 List of Registers

The address 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.
- Reserved addresses are indicated by — in the register name column.
  Do not access the reserved addresses.
- For 16-bit and 32-bit addresses, the MSB address is shown in the table.
- The access size is indicated.

2. Register bits
- Bit configurations of the registers are described in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
- For 16-bit and 32-bit registers, the bits are aligned from the MSB.

3. Register states in each operating mode
- Register states are described in the same order as the register addresses.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

20.1 Register Addresses (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	Bit No.	Address	Module	Data Width	Access States
DTC mode register A	MRA	8	H'BC00 to H'BFFF	DTC	16/32	2
DTC source address register	SAR	24		DTC	16/32	2
DTC mode register B	MRB	8		DTC	16/32	2
DTC destination address register	DAR	24		DTC	16/32	2
DTC transfer count register A	CRA	16		DTC	16/32	2
DTC transfer count register B	CRB	16		DTC	16/32	2
Serial expansion mode register	SEMR	8	H'FDA8	SCI_2	8	2
Interrupt priority register A	IPRA	16	H'FE00	INT	16	2
Interrupt priority register B	IPRB	16	H'FE02	INT	16	2
Interrupt priority register C	IPRC	16	H'FE04	INT	16	2
Interrupt priority register D	IPRD	16	H'FE06	INT	16	2
Interrupt priority register E	IPRE	16	H'FE08	INT	16	2
Interrupt priority register F	IPRF	16	H'FE0A	INT	16	2
Interrupt priority register G	IPRG	16	H'FE0C	INT	16	2
Interrupt priority register H	IPRH	16	H'FE0E	INT	16	2
Interrupt priority register I	IPRI	16	H'FE10	INT	16	2
Interrupt priority register J	IPRJ	16	H'FE12	INT	16	2
Interrupt priority register K	IPRK	16	H'FE14	INT	16	2
IRQ pin select register	ITSR	16	H'FE16	INT	16	2
Software standby release IRQ enable register	SSIER	16	H'FE18	INT	16	2
IRQ sense control register	ISCR	16	H'FE1C	INT	16	2
IrDA control register_0	IrCR_0	8	H'FE1E	IrDA_0	8	2
Port 1 data direction register	P1DDR	8	H'FE20	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FE21	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FE22	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FE24	PORT	8	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Port 6 data direction register	P6DDR	8	H'FE25	PORT	8	2
Port 7 data direction register	P7DDR	8	H'FE26	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FE27	PORT	8	2
Port A data direction register	PADDR	8	H'FE29	PORT	8	2
Port B data direction register	PBDDR	8	H'FE2A	PORT	8	2
Port C data direction register	PCDDR	8	H'FE2B	PORT	8	2
Port D data direction register	PDDDR	8	H'FE2C	PORT	8	2
Port E data direction register	PEDDR	8	H'FE2D	PORT	8	2
Port F data direction register	PFDDR	8	H'FE2E	PORT	8	2
Port G data direction register	PGDDR	8	H'FE2F	PORT	8	2
Port function control register 0	PFCR0	8	H'FE32	PORT	8	2
Port function control register 1	PFCR1	8	H'FE33	PORT	8	2
Port function control register 2	PFCR2	8	H'FE34	PORT	8	2
Port A MOS pull-up control register	PAPCR	8	H'FE36	PORT	8	2
Port B MOS pull-up control register	PBPCR	8	H'FE37	PORT	8	2
Port C MOS pull-up control register	PCPCR	8	H'FE38	PORT	8	2
Port D MOS pull-up control register	PDPCR	8	H'FE39	PORT	8	2
Port E MOS pull-up control register	PEPCR	8	H'FE3A	PORT	8	2
Port 3 open drain control register	P3ODR	8	H'FE3C	PORT	8	2
Port A open drain control register	PAODR	8	H'FE3D	PORT	8	2
Timer control register_3	TCR_3	8	H'FE80	TPU_3	16	2
Timer mode register_3	TMDR_3	8	H'FE81	TPU_3	16	2
Timer I/O control register H_3	TIORH_3	8	H'FE82	TPU_3	16	2
Timer I/O control register L_3	TIORL_3	8	H'FE83	TPU_3	16	2
Timer interrupt enable register_3	TIER_3	8	H'FE84	TPU_3	16	2
Timer status register_3	TSR_3	8	H'FE85	TPU_3	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

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Timer control register_4	TCR_4	8	H'FE90	TPU_4	16	2
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	16	2
Timer I/O control register_4	TIOR_4	8	H'FE92	TPU_4	16	2
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FE95	TPU_4	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	16	2
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	16	2
Timer I/O control register_5	TIOR_5	8	H'FEA2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	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
Bus width control register	ABWCR	8	H'FEC0	BSC	16	2
Access state control register	ASTCR	8	H'FEC1	BSC	16	2
Wait control register AH	WTCRAH	8	H'FEC2	BSC	16	2
Wait control register AL	WTCRAL	8	H'FEC3	BSC	16	2
Wait control register BH	WTCRBH	8	H'FEC4	BSC	16	2
Wait control register BL	WTCRBL	8	H'FEC5	BSC	16	2
Read strobe timing control register	RDNCR	8	H'FEC6	BSC	16	2
Chip select assertion period control registers H	CSACRH	8	H'FEC8	BSC	16	2
Chip select assertion period control register L	CSACRL	8	H'FEC9	BSC	16	2
Bus control register	BCR	16	H'FECC	BSC	16	2
RAM emulation register*1	RAMER	8	H'FECE	FLASH	16	2
DTC enable register A	DTCERA	8	H'FF28	DTC	16	2
DTC enable register B	DTCERB	8	H'FF29	DTC	16	2
DTC enable register C	DTCERC	8	H'FF2A	DTC	16	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
DTC enable register D	DTCERD	8	H'FF2B	DTC	16	2
DTC enable register E	DTCERE	8	H'FF2C	DTC	16	2
DTC enable register F	DTCERF	8	H'FF2D	DTC	16	2
DTC enable register G	DTCERG	8	H'FF2E	DTC	16	2
DTC vector register	DTVECR	8	H'FF30	DTC	16	2
Interrupt control register	INTCR	8	H'FF31	INT	16	2
IRQ enable register	IER	16	H'FF32	INT	16	2
IRQ status register	ISR	16	H'FF34	INT	16	2
Standby control register	SBYCR	8	H'FF3A	SYSTEM	8	2
System clock control register	SCKCR	8	H'FF3B	SYSTEM	8	2
System control register	SYSCR	8	H'FF3D	SYSTEM	8	2
Mode control register	MDCR	8	H'FF3E	SYSTEM	8	2
Module stop control register H	MSTPCRH	8	H'FF40	SYSTEM	8	2
Module stop control register L	MSTPCRL	8	H'FF41	SYSTEM	8	2
PLL control register	PLLCR	8	H'FF45	SYSTEM	8	2
PPG output control register	PCR	8	H'FF46	PPG	8	2
PPG output mode register	PMR	8	H'FF47	PPG	8	2
Next data enable register H	NDERH	8	H'FF48	PPG	8	2
Next data enable register L	NDERL	8	H'FF49	PPG	8	2
Output data register H	PODRH	8	H'FF4A	PPG	8	2
Output data register L	PODRL	8	H'FF4B	PPG	8	2
Next data register H*2	NDRH	8	H'FF4C	PPG	8	2
Next data register L*2	NDRL	8	H'FF4D	PPG	8	2
Next data register H*2	NDRH	8	H'FF4E	PPG	8	2
Next data register L*2	NDRL	8	H'FF4F	PPG	8	2
Port 1 register	PORT1	8	H'FF50	PORT	8	2
Port 2 register	PORT2	8	H'FF51	PORT	8	2
Port 3 register	PORT3	8	H'FF52	PORT	8	2
Port 4 register	PORT4	8	H'FF53	PORT	8	2
Port 5 register	PORT5	8	H'FF54	PORT	8	2
Port 6 register	PORT6	8	H'FF55	PORT	8	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Port 7 register	PORT7	8	H'FF56	PORT	8	2
Port 8 register	PORT8	8	H'FF57	PORT	8	2
Port A register	PORTA	8	H'FF59	PORT	8	2
Port B register	PORTB	8	H'FF5A	PORT	8	2
Port C register	PORTC	8	H'FF5B	PORT	8	2
Port D register	PORTD	8	H'FF5C	PORT	8	2
Port E register	PORTE	8	H'FF5D	PORT	8	2
Port F register	PORTF	8	H'FF5E	PORT	8	2
Port G register	PORTG	8	H'FF5F	PORT	8	2
Port 1 data register	P1DR	8	H'FF60	PORT	8	2
Port 2 data register	P2DR	8	H'FF61	PORT	8	2
Port 3 data register	P3DR	8	H'FF62	PORT	8	2
Port 5 data register	P5DR	8	H'FF64	PORT	8	2
Port 6 data register	P6DR	8	H'FF65	PORT	8	2
Port 7 data register	P7DR	8	H'FF66	PORT	8	2
Port 8 data register	P8DR	8	H'FF67	PORT	8	2
Port A data register	PADR	8	H'FF69	PORT	8	2
Port B data register	PBDR	8	H'FF6A	PORT	8	2
Port C data register	PCDR	8	H'FF6B	PORT	8	2
Port D data register	PDDR	8	H'FF6C	PORT	8	2
Port E data register	PEDR	8	H'FF6D	PORT	8	2
Port F data register	PFDR	8	H'FF6E	PORT	8	2
Port G data register	PGDR	8	H'FF6F	PORT	8	2
Port H register	PORTH	8	H'FF70	PORT	8	2
Port H data register	PHDR	8	H'FF72	PORT	8	2
Port H data direction register	PHDDR	8	H'FF74	PORT	8	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

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
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
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 A	ADDRA	16	H'FF90	A/D	16	2
A/D data register B	ADDRB	16	H'FF92	A/D	16	2
A/D data register C	ADDRC	16	H'FF94	A/D	16	2
A/D data register D	ADDRD	16	H'FF96	A/D	16	2
A/D data register E	ADDRE	16	H'FF98	A/D	16	2
A/D data register F	ADDRF	16	H'FF9A	A/D	16	2
A/D data register G	ADDRG	16	H'FF9C	A/D	16	2
A/D data register H	ADDRH	16	H'FF9E	A/D	16	2
A/D control/status register	ADCSR	8	H'FFA0	A/D	16	2
A/D control register	ADCR	8	H'FFA1	A/D	16	2
D/A data register 0	DADR0	8	H'FFA4	D/A	8	2
D/A data register 1	DADR1	8	H'FFA5	D/A	8	2
D/A control register 01	DACR01	8	H'FFA6	D/A	8	2
D/A data register 2	DADR2	8	H'FFA8	D/A	8	2
D/A data register 3	DADR3	8	H'FFA9	D/A	8	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
D/A control register 23	DACR23	8	H'FFAA	D/A	8	2
Timer control register 0	TCR_0	8	H'FFB0	TMR_0	16	2
Timer control register 1	TCR_1	8	H'FFB1	TMR_1	16	2
Timer control/status register 0	TCSR_0	8	H'FFB2	TMR_0	16	2
Timer control/status register 1	TCSR_1	8	H'FFB3	TMR_1	16	2
Time constant register A0	TCORA_0	8	H'FFB4	TMR_0	16	2
Time constant register A1	TCORA_1	8	H'FFB5	TMR_1	16	2
Time constant register B0	TCORB_0	8	H'FFB6	TMR_0	16	2
Time constant register B1	TCORB_1	8	H'FFB7	TMR_1	16	2
Timer counter 0	TCNT_0	8	H'FFB8	TMR_0	16	2
Timer counter 1	TCNT_1	8	H'FFB9	TMR_1	16	2
Timer control/status register	TCSR	8	H'FFBC*3 (Write) H'FFBC (Read)	WDT	16	2
Timer counter	TCNT	8	H'FFBC*3 (Write) H'FFBD (Read)	WDT	16	2
Reset control/status register	RSTCSR	8	H'FFBE*3 (Write) H'FFBF (Read)	WDT	16	2
Timer start register	TSTR	8	H'FFC0	TPU	16	2
Timer synchronous register	TSYR	8	H'FFC1	TPU	16	2
Flash memory control register 1*1	FLMCR1	8	H'FFC8	FLASH	8	2
Flash memory control register 2*1	FLMCR2	8	H'FFC9	FLASH	8	2
Erase block register 1*1	EBR1	8	H'FFCA	FLASH	8	2
Erase block register 2*1	EBR2	8	H'FFCB	FLASH	8	2
Timer control register_0	TCR_0	8	H'FFD0	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FFD1	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FFD2	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FFD3	TPU_0	16	2

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
Timer interrupt enable register_0	TIER_0	8	H'FFD4	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FFD5	TPU_0	16	2
Timer counter_0	TCNT_0	16	H'FFD6	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FFD8	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FFDA	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FFDC	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FFDE	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FFE0	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FFE1	TPU_1	16	2
Timer I/O control register_1	TIOR_1	8	H'FFE2	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FFE4	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FFE5	TPU_1	16	2
Timer counter_1	TCNT_1	16	H'FFE6	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FFE8	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FFEA	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FFF0	TPU_2	16	2
Timer mode register_2	TMDR_2	8	H'FFF1	TPU_2	16	2
Timer I/O control register_2	TIOR_2	8	H'FFF2	TPU_2	16	2
Timer interrupt enable register_2	TIER_2	8	H'FFF4	TPU_2	16	2
Timer status rgister_2	TSR_2	8	H'FFF5	TPU_2	16	2
Timer counter_2	TCNT_2	16	H'FFF6	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FFF8	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FFFA	TPU_2	16	2

Notes: 1. Register of the flash memory version. Not available in the masked ROM version and ROM-less version.

2. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.

3. For writing, refer to section 12.6.1, Notes on register access.

20.2 Register Bits

Register bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16- or 32-bit registers are shown as 2 or 4 lines.

Register Name	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*7
SAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
MRB	CHNE	DISEL	CHNS	—	—	—	—	—	
DAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRB	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SEMR*7	—	—	—	—	ABCS	ACS2	ACS1	ACS0	SCI_2 Smart card interface 2
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8	INT
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8	
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8	
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0	
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8	
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0	
IPRE	—	IPRE14	IPRE13	IPRE12	—	IPRE10	IPRE9	IPRE8	
	—	IPRE6	IPRE5	IPRE4	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF14	IPRF13	IPRF12	—	IPRF10	IPRF9	IPRF8	
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG14	IPRG13	IPRG12	—	IPRG10	IPRG9	IPRG8	
	—	IPRG6	IPRG5	IPRG4	—	IPRG2	IPRG1	IPRG0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
IPRH	—	IPRH14	IPRH13	IPRH12	—	IPRH10	IPRH9	IPRH8	INT
	—	IPRH6	IPRH5	IPRH4	—	IPRH2	IPRH1	IPRH0	
IPRI	—	IPRI14	IPRI13	IPRI12	—	IPRI10	IPRI9	IPRI8	
	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0	
IPRJ	—	IPRJ14	IPRJ13	IPRJ12	—	IPRJ10	IPRJ9	IPRJ8	
	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRK	—	IPRK14	IPRK13	IPRK12	—	IPRK10	IPRK9	IPRK8	
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0	
ITSR	—	—	—	—	—	—	—	—	
	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0	
SSIER	—	—	—	—	—	—	—	—	
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0	
ISCR	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IrCR_0	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—	IrDA_0
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P5DDR	—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR	
P6DDR	—	—	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
P7DDR	—	—	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR	
P8DDR	—	—	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR	
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	PF2DDR	PF1DDR	PF0DDR	
PGDDR	—	PG6DDR	PG5DDR	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E	
PFCR1	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PFCR2	—	—	—	—	ASOE	LWROE	—	—	PORT
PAPCR	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	—	—	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	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TGRA_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_4
	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	
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	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WTCRAH	—	W72	W71	W70	—	W62	W61	W60	
WTCRAL	—	W52	W51	W50	—	W42	W41	W40	
WTCRBH	—	W32	W31	W30	—	W22	W21	W20	
WTCRBL	—	W12	W11	W10	—	W02	W01	W00	
RDNCR	RDN7	RDN6	RDN5	RDN4	RDN3	RDN2	RDN1	RDN0	
CSACRH	CSXH7	CSXH6	CSXH5	CSXH4	CSXH3	CSXH2	CSXH1	CSXH0	
CSACRL	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1	CSXT0	
BCR	BRLE	BREQ0E	—	IDLC	ICIS1	ICIS0	WDBE	WAITE	
	—	—	—	—	—	ICIS2	—	—	
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	FLASH (F-ZTAT version)

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	—	—	—	—	—	—	—	—	
DTCERC	—	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	—	—	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	—	—	—	—	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	DTCEG7	DTCEG6	—	—	—	—	—	—	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—	INT
IER	—	—	—	—	—	—	—	—	
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	—	—	—	—	—	—	—	—	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
SBYCR	SSBY	OPE	—	—	STS3	STS2	STS1	STS0	SYSTEM
SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	
SYSCR	—	—	MACS	—	FLSHE	—	EXPE	RAME	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRH	ACSE	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
PLLCR	—	—	—	—	—	—	STC1	STC0	
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV	
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	PPG
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0	
NDRH*1	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
NDRL*1	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
NDRH*1	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
NDRL*1	—	—	—	—	NDR3	NDR2	NDR1	NDR0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT2	P27	P26	P25	P24	P23	P22	P21	P20	
PORT3	—	—	P35	P34	P33	P32	P31	P30	
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT5	P57	P56	P55	P54	P53	P52	P51	P50	
PORT6	—	—	P65	P64	P63	P62	P61	P60	
PORT7	—	—	P75	P74	P73	P72	P71	P70	
PORT8	—	—	P85	P84	P83	P82	P81	P80	
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	—	PG6	PG5	PG4	PG3	PG2	PG1	PG0	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P5DR	—	—	—	—	P53DR	P52DR	P51DR	P50DR	
P6DR	—	—	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	
P7DR	—	—	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR	
P8DR	—	—	P85DR	P84DR	P83DR	P82DR	P81RD	P80DR	
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	—	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR	
PORTH	—	—	—	—	PH3	PH2	PH1	PH0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PHDR	—	—	—	—	PH3DR	PH2DR	PH1DR	PH0DR	PORT
PHDDR	—	—	—	—	PH3DDR	PH2DDR	PH1DDR	PH0DDR	
SMR_0	C/$\overline{A}$/ GM*2	CHR/ BLK*3	PE	O/$\overline{E}$	STOP/ BCP1*4	MP/ BCP0*5	CKS1	CKS0	SCI_0, Smart card interface_0
BRR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_0	TDRE	RDRF	ORER	FER/ ERS*6	PER	TEND	MPB	MPBT	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_1	C/$\overline{A}$/ GM*2	CHR/ BLK*3	PE	O/$\overline{E}$	STOP/ BCP1*4	MP/ BCP0*5	CKS1	CKS0	SCI_1, Smart card interface_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	TDRE	RDRF	ORER	FER/ ERS*6	PER	TEND	MPB	MPBT	
RDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2	C/$\overline{A}$/ GM*2	CHR/ BLK*3	PE	O/$\overline{E}$	STOP/ BCP1*4	MP/ BCP0*5	CKS1	CKS0	SCI_2, Smart card interface_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	TDRE	RDRF	ORER	FER/ ERS*6	PER	TEND	MPB	MPBT	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRE	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRF	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRG	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CH3	—	—	
DADR0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	D/A
DADR1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—	
DADR2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DADR3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DACR23	DAOE3	DAOE2	DAE	—	—	—	—	—	
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	
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORA_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORB_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORB_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCSR	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RSTCSR	WOVF	RSTE	—	—	—	—	—	—	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	FLASH (F-ZTAT version)
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	EB13	EB12	EB11	EB10	EB9	EB8	
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	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCNT_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_1
	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	

Notes:
1. If the PCR setting specifies the same output trigger for pulse output group 2 and pulse output group 3, the address is H'FF4C. If the triggers are different, the NDRH address corresponding to pulse output group 2 is H'FF4E and the NDRH address corresponding to pulse output group 3 is H'FF4C. In like manner, if the PCR setting specifies the same output trigger for pulse output group 0 and pulse output group 1, the address is H'FF4D. If the triggers are different, the NDRH address corresponding to pulse output group 0 is H'FF4F and the NDRH address corresponding to pulse output group 1 is H'FF4D.
2. Functions as C/$\overline{A}$ for SCI use, and as GM for smart card interface use.
3. Functions as CHR for SCI use, and as BLK for smart card interface use.
4. Functions as STOP for SCI use, and as BCP1 for smart card interface use.
5. Functions as MP for SCI use, and as BCP0 for smart card interface use.
6. Functions as FER for SCI use, and as ERS for smart card interface use.
7. Loaded in on-chip RAM. The bus width is 32 bits when the DTC accesses this area as register information, and 16 bits otherwise.

20.3 Register States in Each Operating Mode

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
MRA	Initialized	—	—	—	—	—	—	Initialized	DTC
SAR	Initialized	—	—	—	—	—	—	Initialized	
MRB	Initialized	—	—	—	—	—	—	Initialized	
DAR	Initialized	—	—	—	—	—	—	Initialized	
CRA	Initialized	—	—	—	—	—	—	Initialized	
CRB	Initialized	—	—	—	—	—	—	Initialized	
SEMR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI2
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	
ITSR	Initialized	—	—	—	—	—	—	Initialized	
SSIER	Initialized	—	—	—	—	—	—	Initialized	
ISCR	Initialized	—	—	—	—	—	—	Initialized	
IrCR_0	Initialized	—	—	—	—	—	—	Initialized	IrDA_0
P1DDR	Initialized	—	—	—	—	—	—	—	PORT
P2DDR	Initialized	—	—	—	—	—	—	—	
P3DDR	Initialized	—	—	—	—	—	—	—	
P5DDR	Initialized	—	—	—	—	—	—	—	
P6DDR	Initialized	—	—	—	—	—	—	—	
P7DDR	Initialized	—	—	—	—	—	—	—	
P8DDR	Initialized	—	—	—	—	—	—	—	
PADDR	Initialized	—	—	—	—	—	—	—	
PBDDR	Initialized	—	—	—	—	—	—	—	
PCDDR	Initialized	—	—	—	—	—	—	—	

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
PDDDR	Initialized	—	—	—	—	—	—	—	PORT
PEDDR	Initialized	—	—	—	—	—	—	—	
PFDDR	Initialized	—	—	—	—	—	—	—	
PGDDR	Initialized	—	—	—	—	—	—	—	
PFCR0	Initialized	—	—	—	—	—	—	—	
PFCR1	Initialized	—	—	—	—	—	—	—	
PFCR2	Initialized	—	—	—	—	—	—	—	
PAPCR	Initialized	—	—	—	—	—	—	—	
PBPCR	Initialized	—	—	—	—	—	—	—	
PCPCR	Initialized	—	—	—	—	—	—	—	
PDPCR	Initialized	—	—	—	—	—	—	—	
PEPCR	Initialized	—	—	—	—	—	—	—	
P3ODR	Initialized	—	—	—	—	—	—	—	
PAODR	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	
TGRB_3	Initialized	—	—	—	—	—	—	Initialized	
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	

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
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	
ABWCR	Initialized	—	—	—	—	—	—	Initialized	BSC
ASTCR	Initialized	—	—	—	—	—	—	Initialized	
WTCRAH	Initialized	—	—	—	—	—	—	Initialized	
WTCRAL	Initialized	—	—	—	—	—	—	Initialized	
WTCRBH	Initialized	—	—	—	—	—	—	Initialized	
WTCRBL	Initialized	—	—	—	—	—	—	Initialized	
RDNCR	Initialized	—	—	—	—	—	—	Initialized	
CSACRH	Initialized	—	—	—	—	—	—	Initialized	
CSACRL	Initialized	—	—	—	—	—	—	Initialized	
BCR	Initialized	—	—	—	—	—	—	Initialized	
RAMER	Initialized	—	—	—	—	—	—	Initialized	FLASH (F-ZTAT version)
DTCERA	Initialized	—	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	—	—	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	Initialized	
DTCERF	Initialized	—	—	—	—	—	—	Initialized	
DTCERG	Initialized	—	—	—	—	—	—	Initialized	
DTVECR	Initialized	—	—	—	—	—	—	Initialized	
INTCR	Initialized	—	—	—	—	—	—	Initialized	INT
IER	Initialized	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	Initialized	

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
SBYCR	Initialized	—	—	—	—	—	—	Initialized	SYSTEM
SCKCR	Initialized	—	—	—	—	—	—	Initialized	
SYSCR	Initialized	—	—	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRH	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRL	Initialized	—	—	—	—	—	—	Initialized	
PLLCR	Initialized	—	—	—	—	—	—	Initialized	
PCR	Initialized	—	—	—	—	—	—	Initialized	PPG
PMR	Initialized	—	—	—	—	—	—	Initialized	
NDERH	Initialized	—	—	—	—	—	—	Initialized	
NDERL	Initialized	—	—	—	—	—	—	Initialized	
PODRH	Initialized	—	—	—	—	—	—	Initialized	
PODRL	Initialized	—	—	—	—	—	—	Initialized	
NDRH	Initialized	—	—	—	—	—	—	Initialized	
NDRL	Initialized	—	—	—	—	—	—	Initialized	
NDRH	Initialized	—	—	—	—	—	—	Initialized	
NDRL	Initialized	—	—	—	—	—	—	Initialized	
PORT1	—	—	—	—	—	—	—	—	PORT
PORT2	—	—	—	—	—	—	—	—	
PORT3	—	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	—	—	
PORT6	—	—	—	—	—	—	—	—	
PORT7	—	—	—	—	—	—	—	—	
PORT8	—	—	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	—	—	
PORTG	—	—	—	—	—	—	—	—	

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
P1DR	Initialized	—	—	—	—	—	—	—	PORT
P2DR	Initialized	—	—	—	—	—	—	—	
P3DR	Initialized	—	—	—	—	—	—	—	
P5DR	Initialized	—	—	—	—	—	—	—	
P6DR	Initialized	—	—	—	—	—	—	—	
P7DR	Initialized	—	—	—	—	—	—	—	
P8DR	Initialized	—	—	—	—	—	—	—	
PADR	Initialized	—	—	—	—	—	—	—	
PBDR	Initialized	—	—	—	—	—	—	—	
PCDR	Initialized	—	—	—	—	—	—	—	
PDDR	Initialized	—	—	—	—	—	—	—	
PEDR	Initialized	—	—	—	—	—	—	—	
PFDR	Initialized	—	—	—	—	—	—	—	
PGDR	Initialized	—	—	—	—	—	—	—	
PORTH	—	—	—	—	—	—	—	—	
PHDR	Initialized	—	—	—	—	—	—	—	
PHDDR	Initialized	—	—	—	—	—	—	—	
SMR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI_0
BRR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SMR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI_1
BRR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
SMR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	SCI_2
BRR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRA	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	A/D
ADDRB	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRC	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRD	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRE	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRF	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRG	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADDRH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ADCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
DADR0	Initialized	—	—	—	—	—	—	Initialized	D/A
DADR1	Initialized	—	—	—	—	—	—	Initialized	
DACR01	Initialized	—	—	—	—	—	—	Initialized	
DADR2	Initialized	—	—	—	—	—	—	Initialized	
DADR3	Initialized	—	—	—	—	—	—	Initialized	
DACR23	Initialized	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	—	TMR_0
TCR_1	Initialized	—	—	—	—	—	—	—	TMR_1
TCSR_0	Initialized	—	—	—	—	—	—	—	
TCSR_1	Initialized	—	—	—	—	—	—	—	
TCORA_0	Initialized	—	—	—	—	—	—	—	
TCORA_1	Initialized	—	—	—	—	—	—	—	
TCORB_0	Initialized	—	—	—	—	—	—	—	
TCORB_1	Initialized	—	—	—	—	—	—	—	
TCNT_0	Initialized	—	—	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	—	—	

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
TCSR	Initialized	—	—	—	—	—	—	—	WDT
TCNT	Initialized	—	—	—	—	—	—	—	
RSTCSR	Initialized	—	—	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	—	Initialized	TPU
TSYR	Initialized	—	—	—	—	—	—	Initialized	
FLMCR1	Initialized	—	—	—	—	—	—	Initialized	FLASH (F-ZTAT version)
FLMCR2	Initialized	—	—	—	—	—	—	Initialized	
EBR1	Initialized	—	—	—	—	—	—	Initialized	
EBR2	Initialized	—	—	—	—	—	—	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	
TIER_1	Initialized	—	—	—	—	—	—	Initialized	
TSR_1	Initialized	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	
TGRA_1	Initialized	—	—	—	—	—	—	Initialized	
TGRB_1	Initialized	—	—	—	—	—	—	Initialized	

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module
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	

Section 21 Electrical Characteristics

21.1 Electrical Characteristics of F-ZTAT Version (H8S/2667)

21.1.1 Absolute Maximum Ratings

Table 21.1 lists the absolute maximum ratings.

Table 21.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC} $PLLV_{CC}$	–0.3 to +4.0	V
Input voltage (except port 4, P54 to P57)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4, P54 to P57)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +4.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75*	°C
		Wide-range specifications: –40 to +85*	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note: * The operating temperature ranges for flash memory programming/erasing are as follows:

T_a = 0°C to +75°C (regular specifications)

T_a = 0°C to +85°C (wide-range specifications)

21.1.2 DC Characteristics

Table 21.2 DC Characteristics (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V*1, 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	Test Conditions
Schmitt trigger input voltage	Port 1, 2, 8*2, P50 to P53*2, PH2*2, PH3*2	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
	P54 to P57*2	VT^-	$AV_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$AV_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$AV_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{STBY}$, MD2 to MD0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	$\overline{RES}$, NMI		$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Port 3, 6 to 8*3, P50 to P53*3, ports A to H*3		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Port 4, P54 to P57*3		$AV_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		–0.3	—	$V_{CC} \times 0.2$	V	
	Ports 3 to 8, ports A to H*3		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA

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. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.
2. When used as $\overline{IRQ0}$ to $\overline{IRQ7}$.
3. When used as other than $\overline{IRQ0}$ to $\overline{IRQ7}$.

Table 21.3 DC Characteristics (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V*1, 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	Test Conditions
Input leakage current	$\overline{RES}$	$\|I_{in}\|$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD2 to MD0		—	—	1.0	µA	
	Port 4, P54 to P57		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V
Three-state leakage current (off state)	Ports 1 to 3, 6 to 8, P50 to P53, ports A to H	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	µA	V_{CC} = 3.0 to 3.6 V V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V
	NMI		—	—	30	pF	f = 1 MHz
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	T_a = 25°C
Current dissipation*2	Normal operation	I_{CC}*4	—	80 (3.3 V)	150	mA	f = 33 MHz
	Sleep mode		—	70 (3.3 V)	125	mA	f = 33 MHz
	Standby mode*3		—	0.01	10	µA	$T_a \leq$ 50°C
			—	—	80	µA	50°C < T_a
	When all module clocks stop*5		—	50 (3.3 V)	125	mA	f = 33 MHz
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	0.2 (3.0 V)	2.0	mA	
	Idle		—	0.01	5.0	µA	

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	1.4 (3.0 V)	4.0	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. When the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

2. Current dissipation values are for V_{IH}min = V_{CC} – 0.5 V and V_{IL}max = 0.5 V with all output pins unloaded and all MOS input pull-ups 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 = 1.0 (mA) + 1.2 (mA/(MHz × V)) × V_{CC} × f (normal operation)
   I_{CC}max = 1.0 (mA) + 1.0 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

5. The values are for reference.

Table 21.4 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 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)	All output pins	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 21.4.

Note: * If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

21.1.3 AC Characteristics

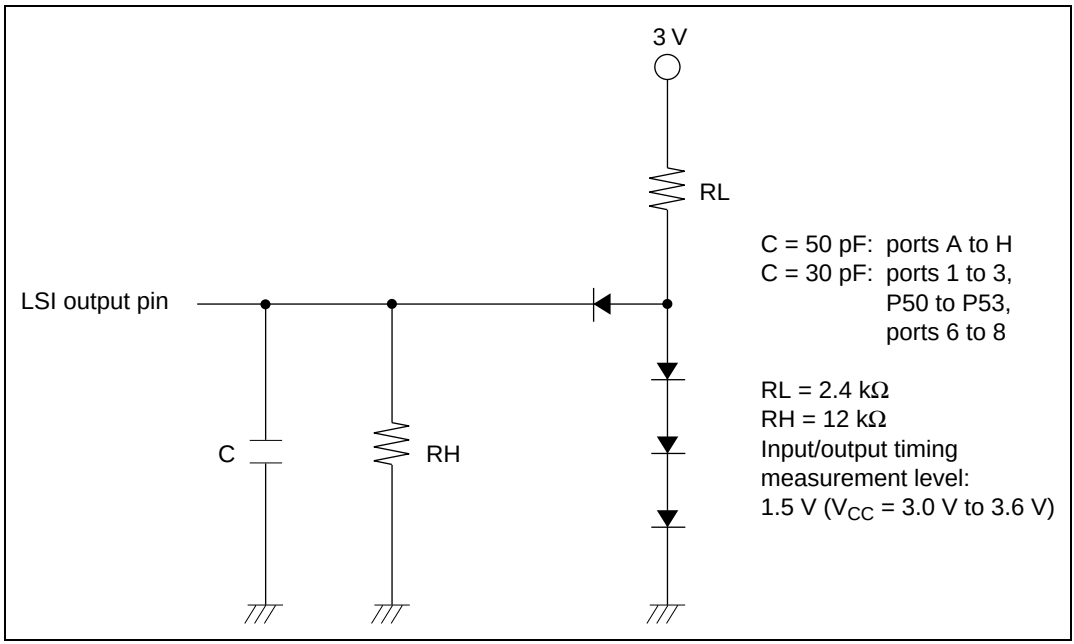


Figure 21.1 Output Load Circuit

Clock Timing

Table 21.5 Clock Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 8 MHz to 33 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}	30.3	125	ns	Figure 21.2
Clock pulse high width	t_{CH}	10	—	ns	
Clock pulse low width	t_{CL}	10	—	ns	
Clock rise time	t_{Cr}	—	5	ns	
Clock fall time	t_{Cf}	—	5	ns	
Reset oscillation stabilization time (crystal)	t_{OSC1}	10	—	ms	Figure 21.3(1)
Software standby oscillation stabilization time (crystal)	t_{OSC2}	10	—	ms	Figure 21.3(2)
External clock output delay stabilization time	t_{DEXT}	500	—	µs	Figure 21.3(1)

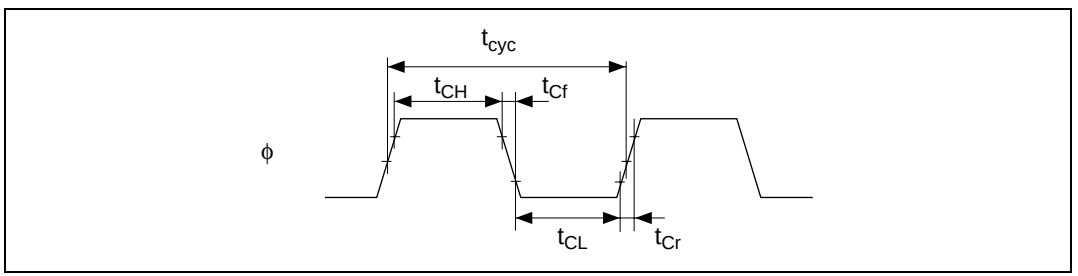


Figure 21.2 System Clock Timing

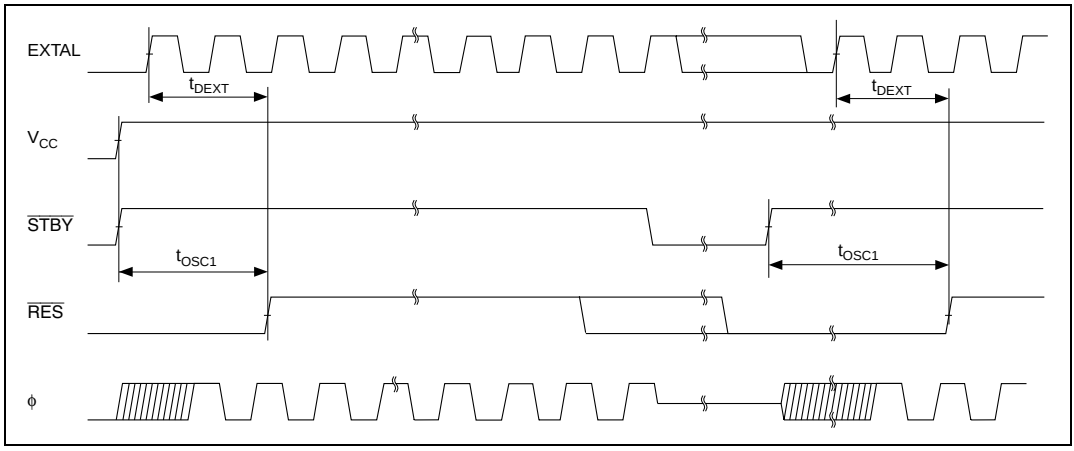


Figure 21.3(1) Oscillation Stabilization Timing

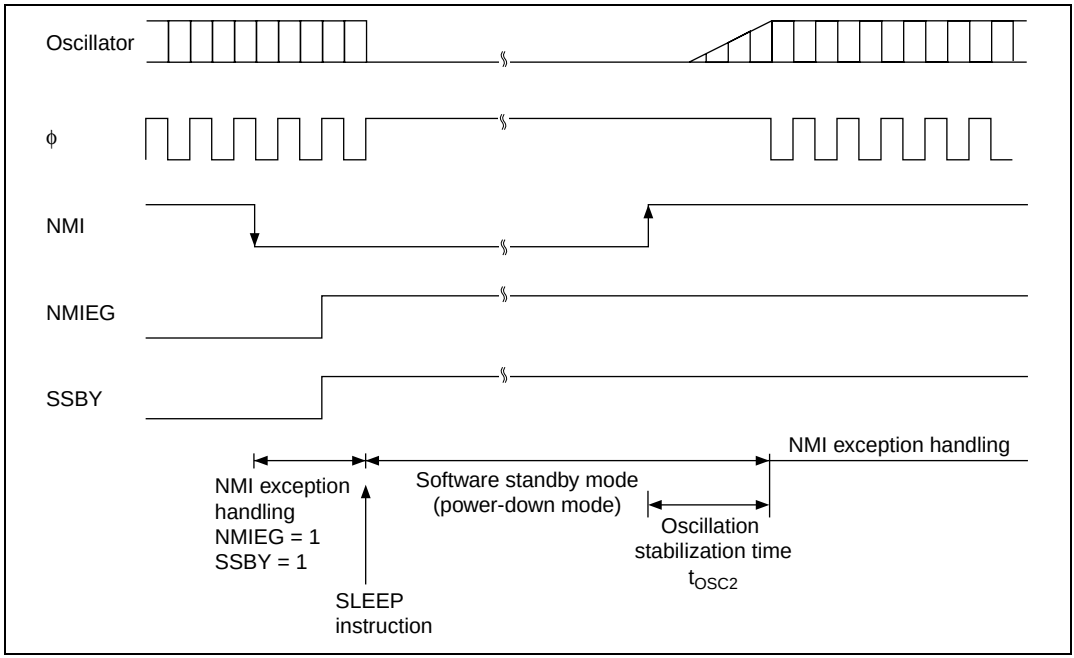


Figure 21.3(2) Oscillation Stabilization Timing

Control Signal Timing

Table 21.6 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 8 MHz to 33 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
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 21.4
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 21.5
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (in recovery from software standby mode)	t_{NMIW}	200	—		
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—		
$\overline{IRQ}$ pulse width (in recovery from software standby mode)	t_{IRQW}	200	—		

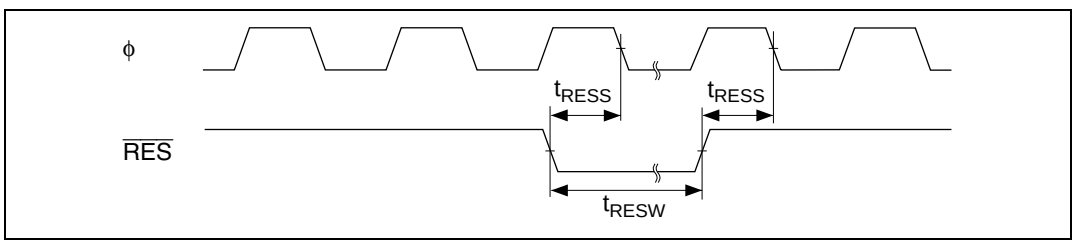
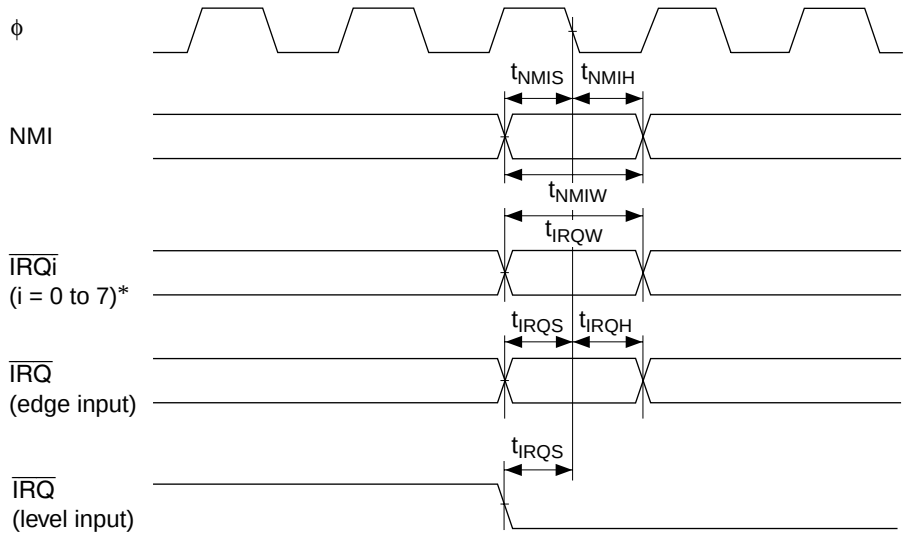


Figure 21.4 Reset Input Timing



Note: * Necessary for SSIER setting to clear software standby mode.

Figure 21.5 Interrupt Input Timing

Bus Timing

Table 21.7 Bus Timing (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 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
Address delay time	TA_D	—	20	ns	Figures 21.6 to 21.10
Address setup time 1	TA_{S1}	$0.5 \times t_{cyc} - 13$	—	ns	
Address setup time 2	TA_{S2}	$1.0 \times t_{cyc} - 13$	—	ns	
Address setup time 3	TA_{S3}	$1.5 \times t_{cyc} - 13$	—	ns	
Address setup time 4	TA_{S4}	$2.0 \times t_{cyc} - 13$	—	ns	
Address hold time 1	TA_{H1}	$0.5 \times t_{cyc} - 8$	—	ns	
Address hold time 2	TA_{H2}	$1.0 \times t_{cyc} - 8$	—	ns	
Address hold time 3	TA_{H3}	$1.5 \times t_{cyc} - 8$	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{AS}$ delay time	TA_{SD}	—	15	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	15	ns	
Read data setup time 1	t_{RDS1}	15	—	ns	
Read data setup time 2	t_{RDS2}	15	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	0	—	ns	
Read data access time 2	T_{AC2}	—	$1.5 \times t_{cyc} - 20$	ns	
Read data access time 4	T_{AC4}	—	$2.5 \times t_{cyc} - 20$	ns	
Read data access time 5	T_{AC5}	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 6	T_{AC6}	—	$2.0 \times t_{cyc} - 20$	ns	
Address read data access time 2	T_{AA2}	—	$1.5 \times t_{cyc} - 20$	ns	
Address read data access time 3	T_{AA3}	—	$2.0 \times t_{cyc} - 20$	ns	
Address read data access time 4	T_{AA4}	—	$2.5 \times t_{cyc} - 20$	ns	
Address read data access time 5	T_{AA5}	—	$3.0 \times t_{cyc} - 20$	ns	

Table 21.8 Bus Timing (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 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
$\overline{WR}$ delay time 1	t_{WRD1}	—	15	ns	Figures 21.6 to 21.10
$\overline{WR}$ delay time 2	t_{WRD2}	—	15	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 13$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time 1	t_{WDS1}	$0.5 \times t_{cyc} - 13$	—	ns	
Write data setup time 2	t_{WDS2}	$1.0 \times t_{cyc} - 13$	—	ns	
Write data setup time 3	t_{WDS3}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 1	t_{WDH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Write data hold time 3	t_{WDH3}	$1.5 \times t_{cyc} - 8$	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	25	—	ns	Figure 21.8
$\overline{WAIT}$ hold time	t_{WTH}	5	—	ns	
$\overline{BREQ}$ setup time	t_{BREQS}	30	—	ns	Figure 21.11
$\overline{BACK}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	40	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	25	ns	Figure 21.12

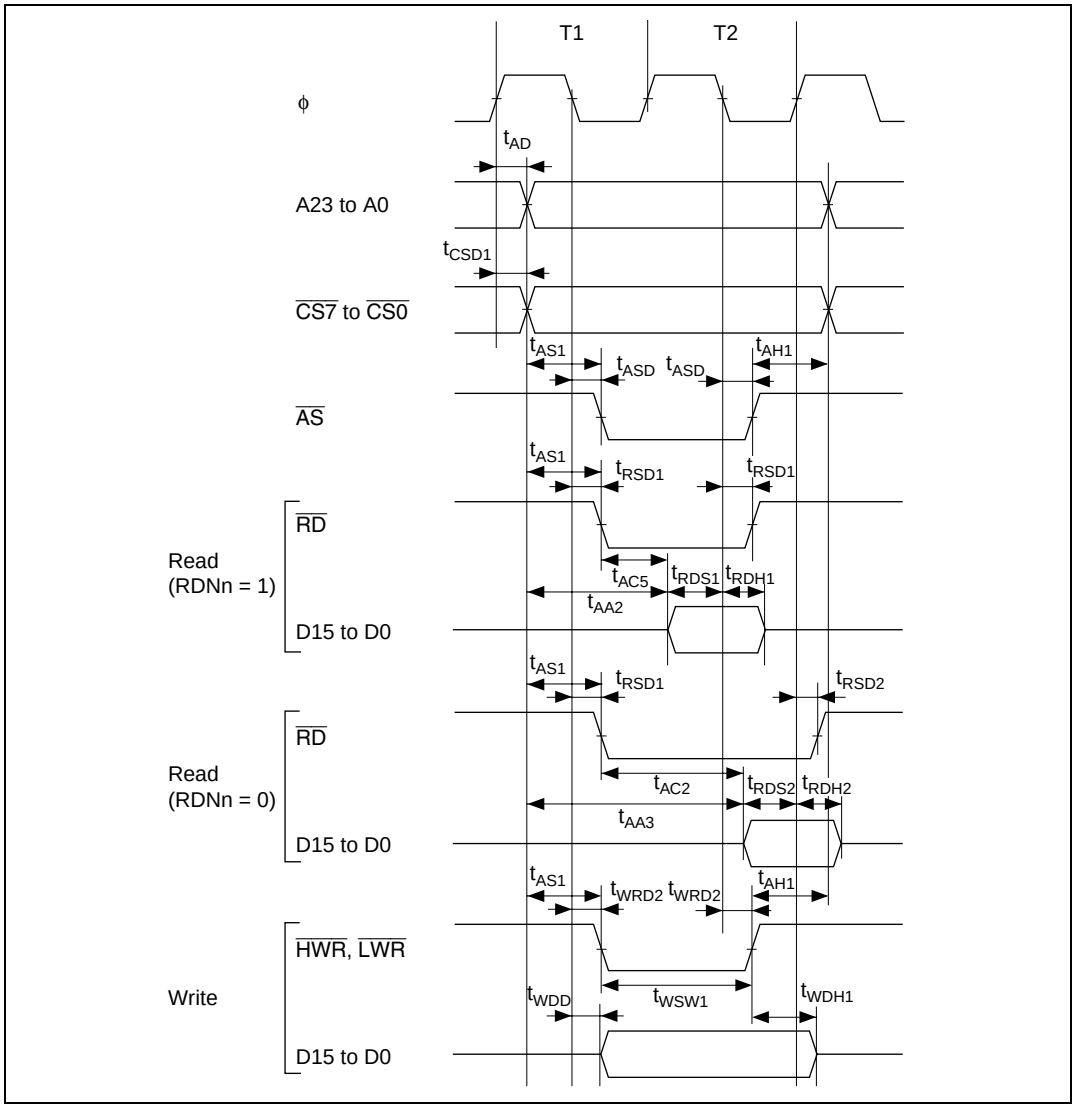


Figure 21.6 Basic Bus Timing: Two-State Access

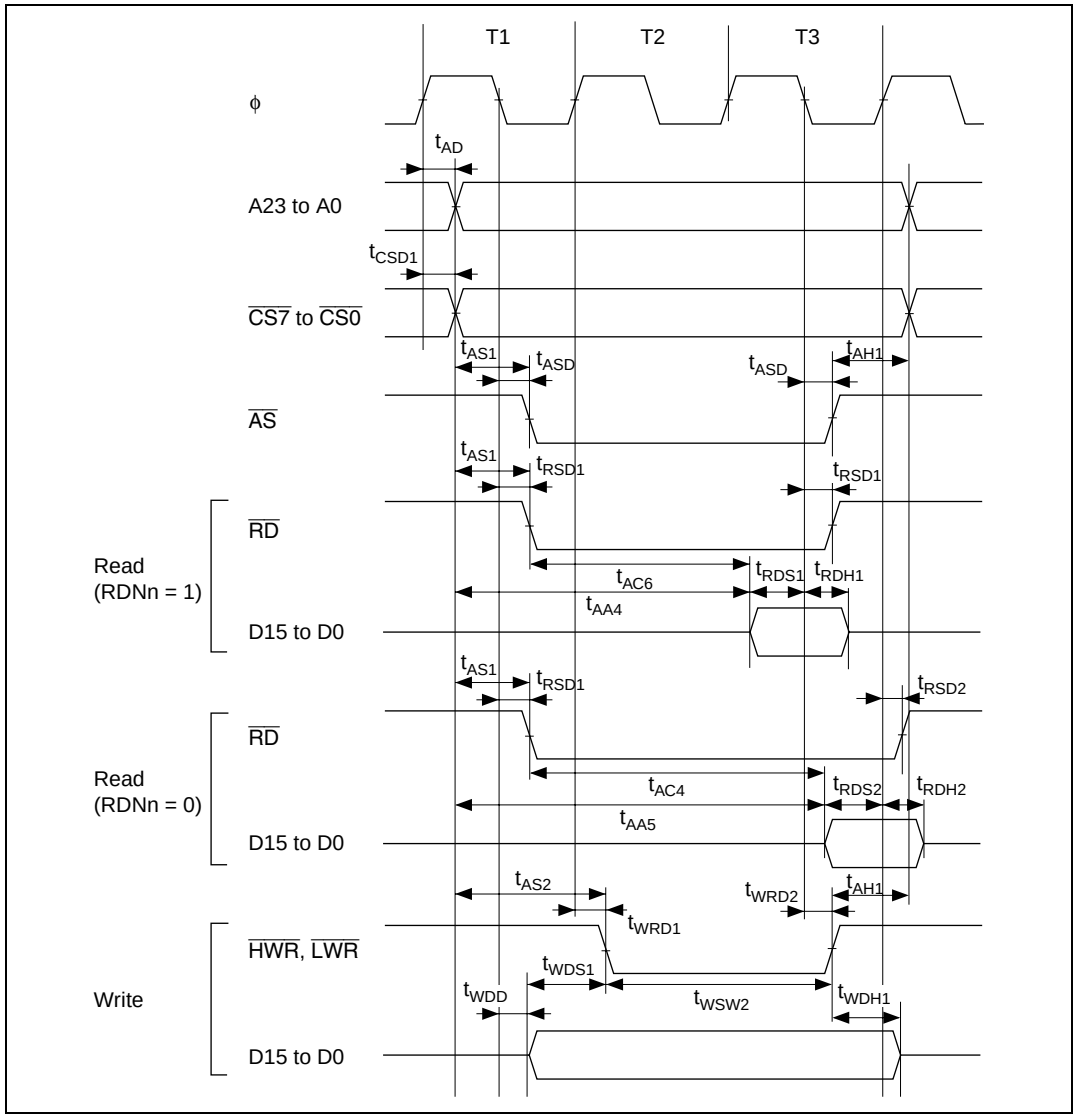


Figure 21.7 Basic Bus Timing: Three-State Access

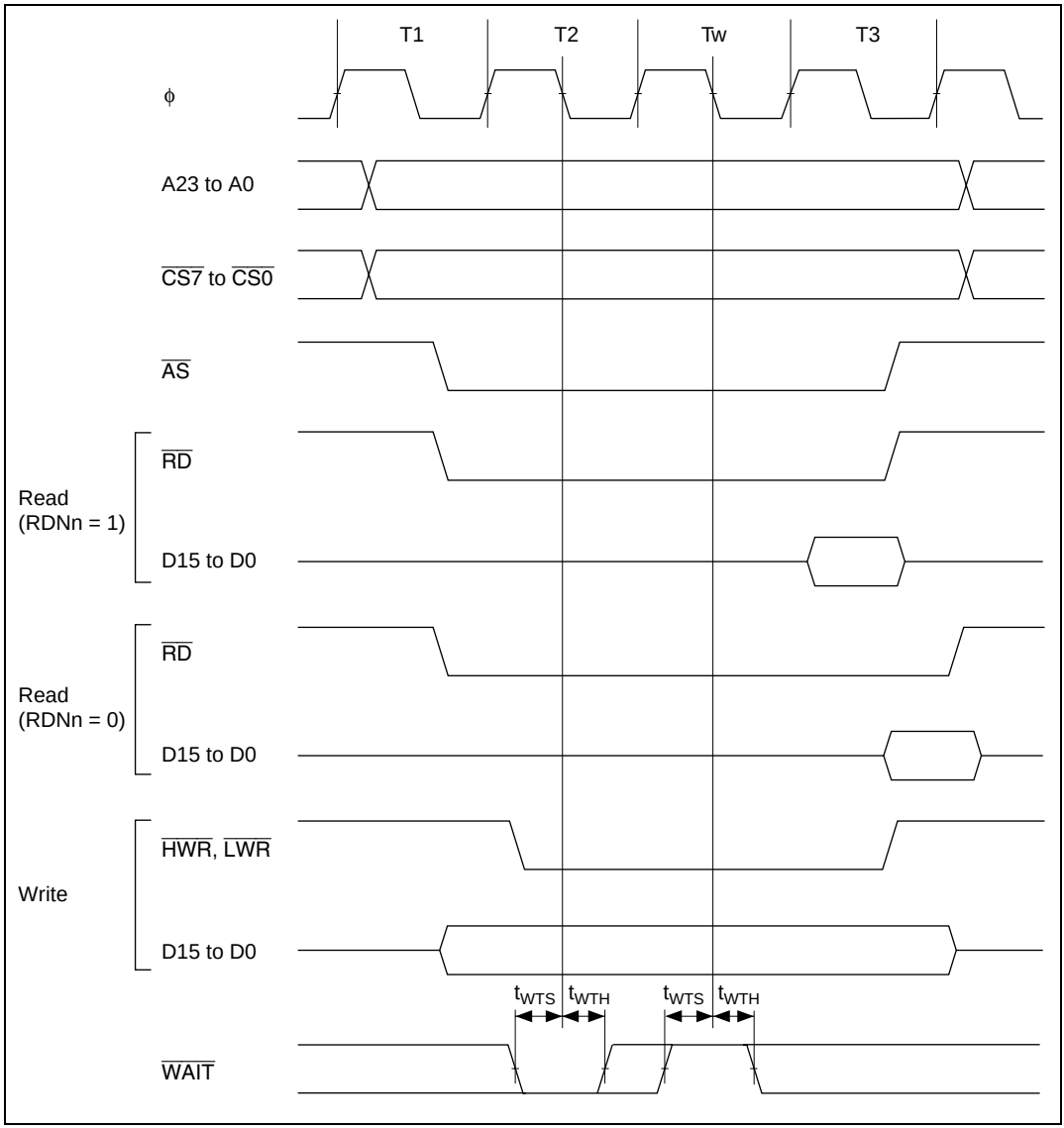


Figure 21.8 Basic Bus Timing: Three-State Access, One Wait

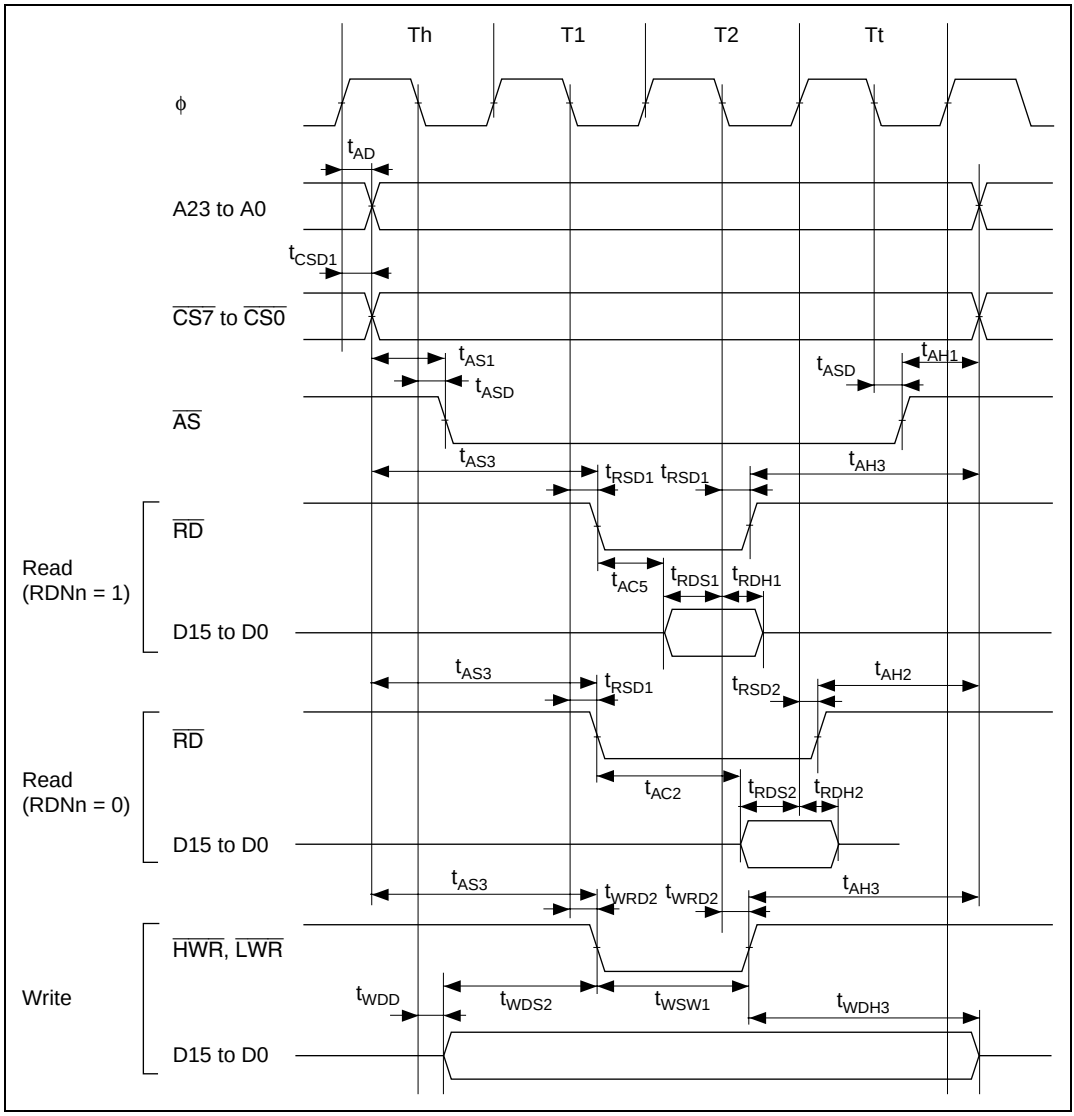


Figure 21.9 Basic Bus Timing: Two-State Access ($\overline{CS}$ Assertion Period Extended)

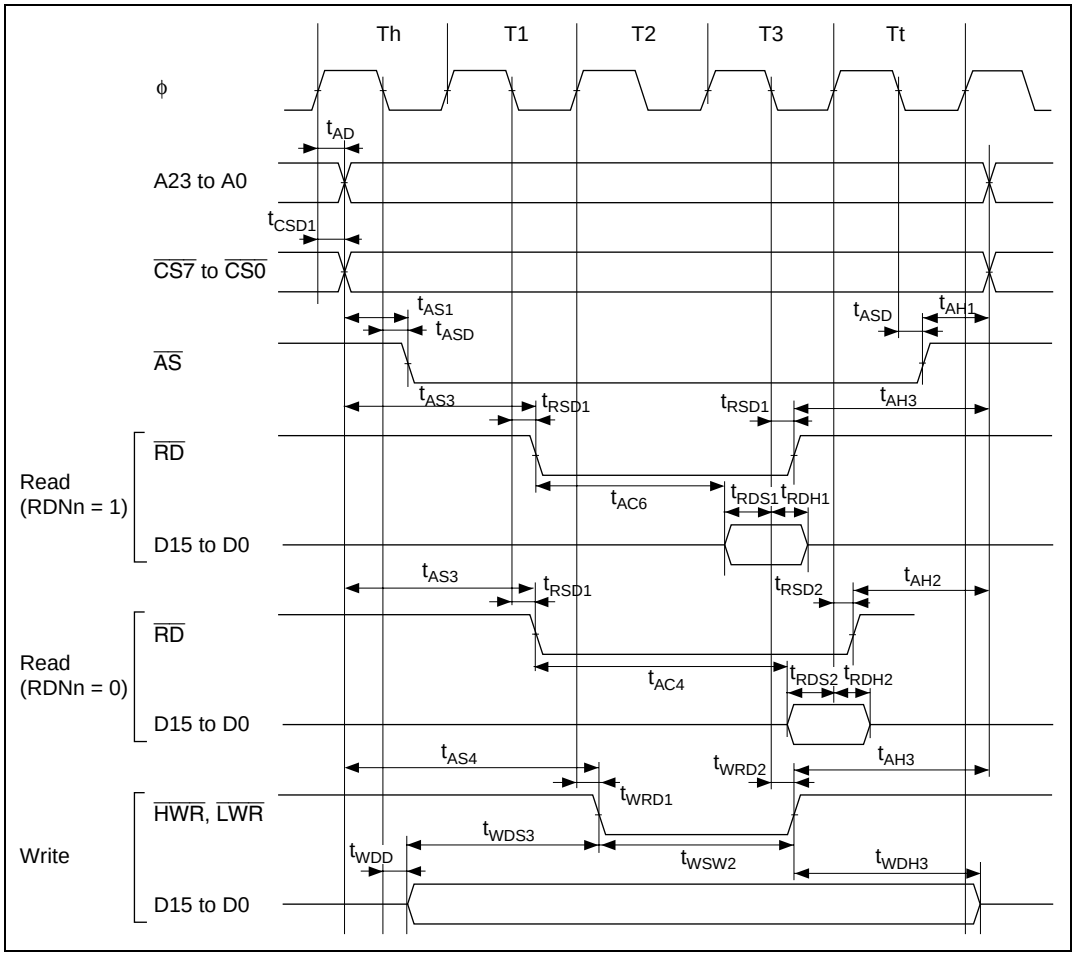


Figure 21.10 Basic Bus Timing: Three-State Access ($\overline{CS}$ Assertion Period Extended)

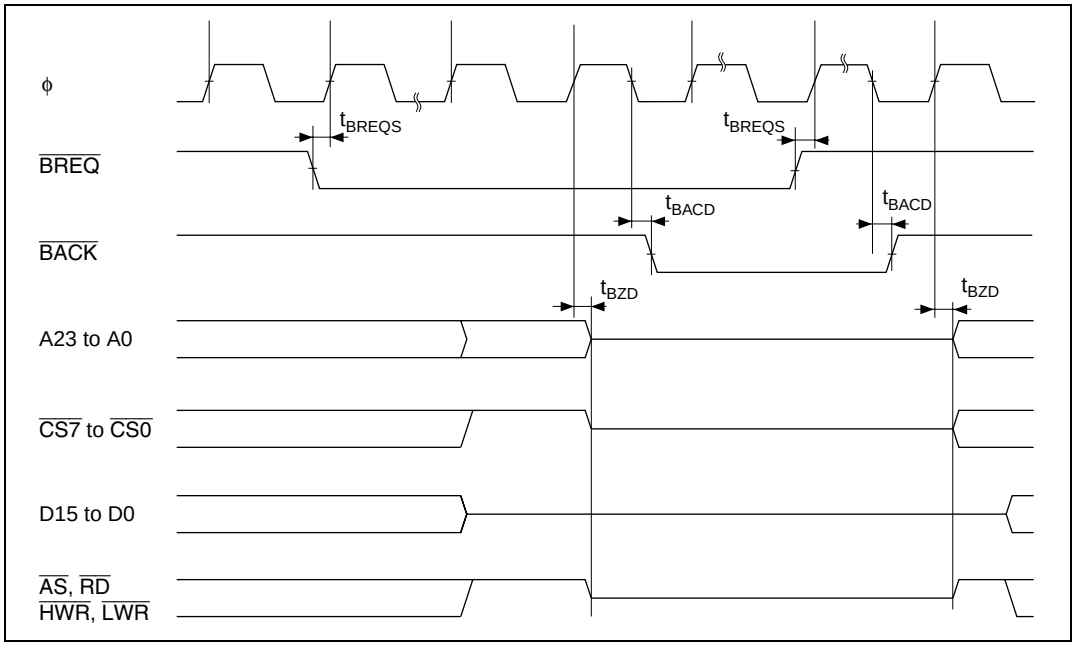


Figure 21.11 External Bus Release Timing

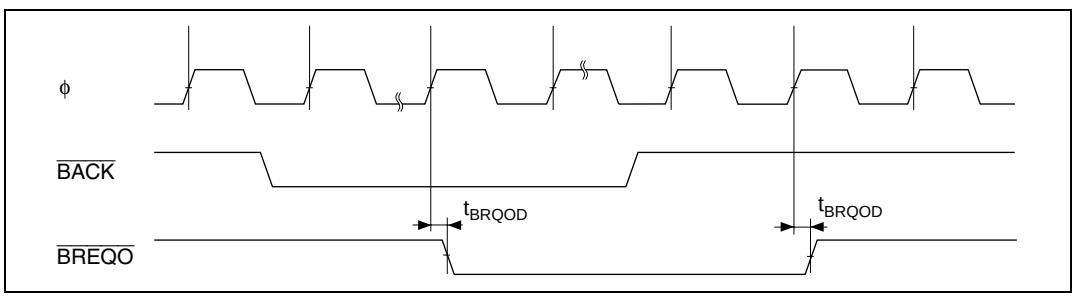


Figure 21.12 External Bus Request Output Timing

Timing of On-Chip Peripheral Modules

Table 21.9 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 33 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
I/O ports	Output data delay time		t_{PWD}	—	40	ns	Figure 21.13
	Input data setup time		t_{PRS}	25	—	ns	
	Input data hold time		t_{PRH}	25	—	ns	
PPG	Pulse output delay time		t_{POD}	—	40	ns	Figure 21.14
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure 21.15
	Timer input setup time		t_{TICS}	25	—	ns	
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure 21.16
	Timer clock pulse width	Single-edge specification	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TCKWL}	2.5	—	t_{cyc}	
8-bit timer	Timer output delay time		t_{TMOD}	—	40	ns	Figure 21.17
	Timer reset input setup time		t_{TMRS}	25	—	ns	Figure 21.19
	Timer clock input setup time		t_{TMCS}	25	—	ns	Figure 21.18
	Timer clock pulse width	Single-edge specification	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TMCWL}	2.5	—	t_{cyc}	
WDT	Overflow output delay time		t_{WOVD}	—	40	ns	Figure 21.20
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 21.21
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	40	ns	Figure 21.22
	Receive data setup time (synchronous)		t_{RXS}	40	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	40	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure 21.23

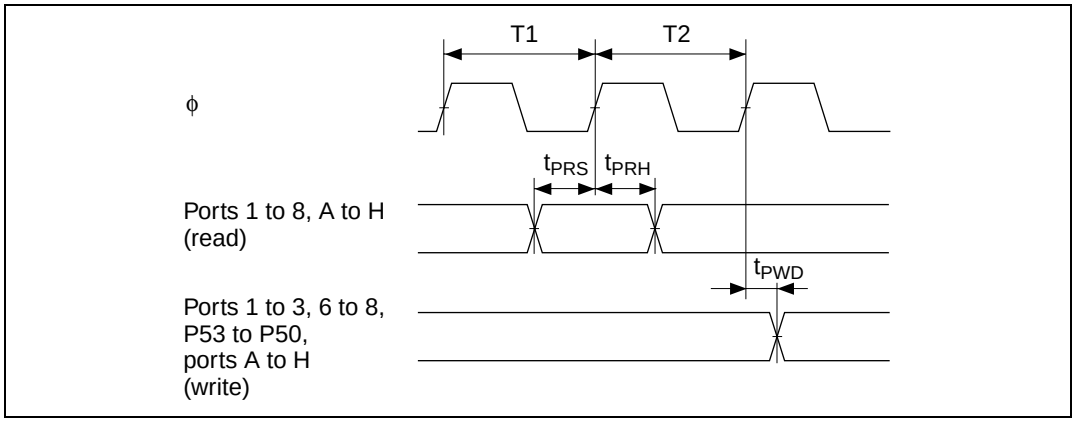


Figure 21.13 I/O Port Input/Output Timing

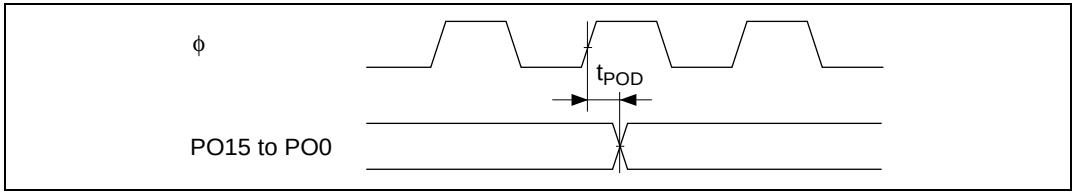
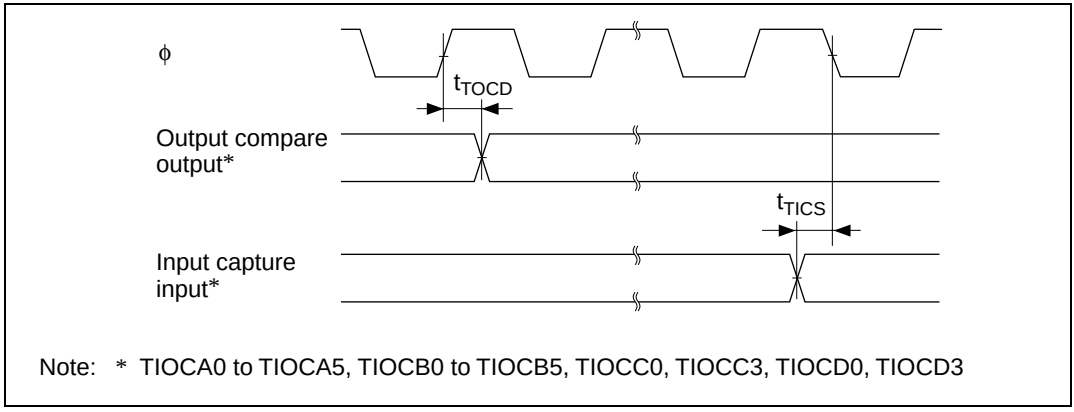


Figure 21.14 PPG Output Timing



Note: * TIOCA0 to TIOCA5, TIOCB0 to TIOCB5, TIOCC0, TIOCC3, TIOCD0, TIOCD3

Figure 21.15 TPU Input/Output Timing

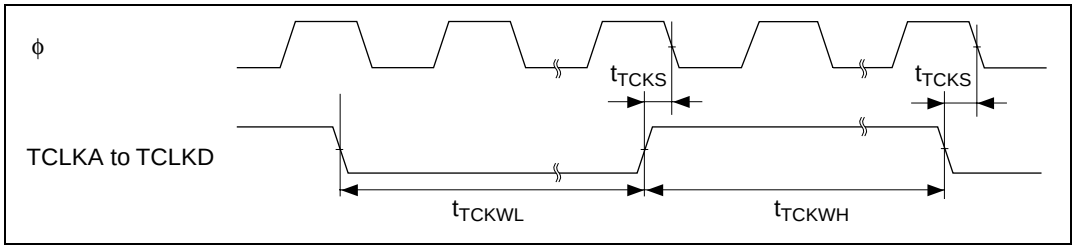


Figure 21.16 TPU Clock Input Timing

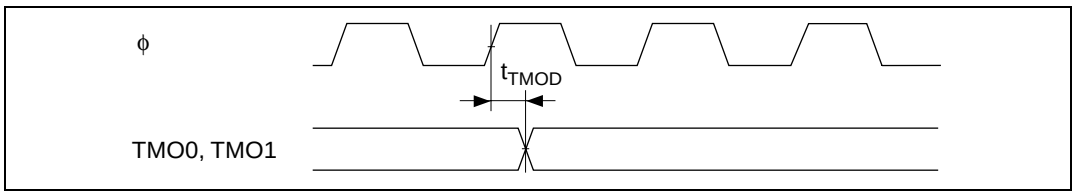


Figure 21.17 8-Bit Timer Output Timing

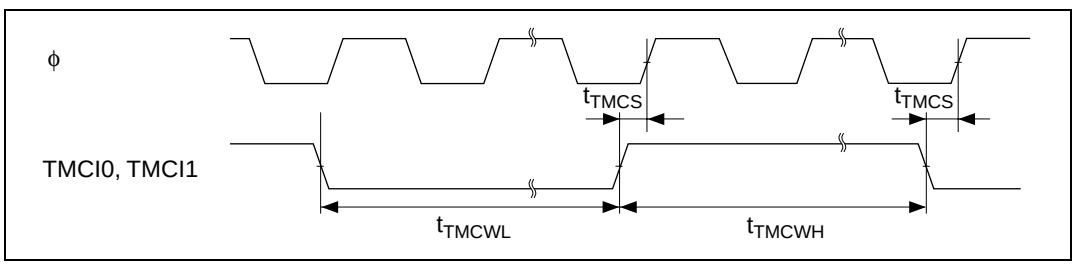


Figure 21.18 8-Bit Timer Clock Input Timing

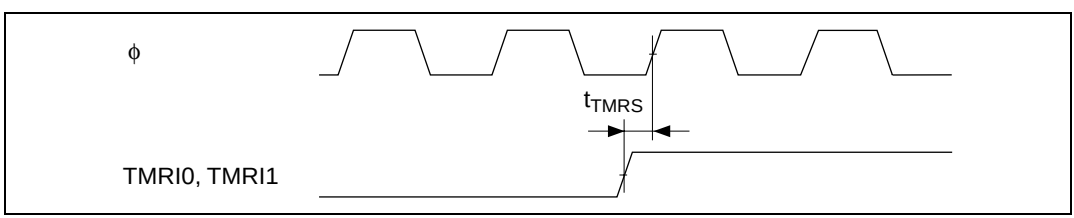


Figure 21.19 8-Bit Timer Reset Input Timing

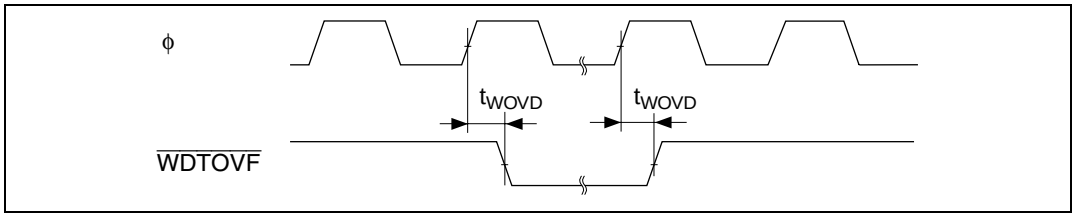


Figure 21.20 WDT Output Timing

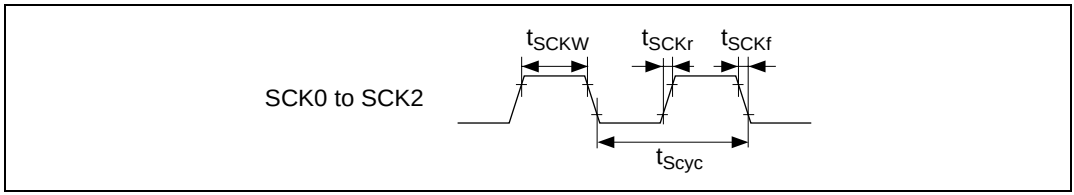


Figure 21.21 SCK Clock Input Timing

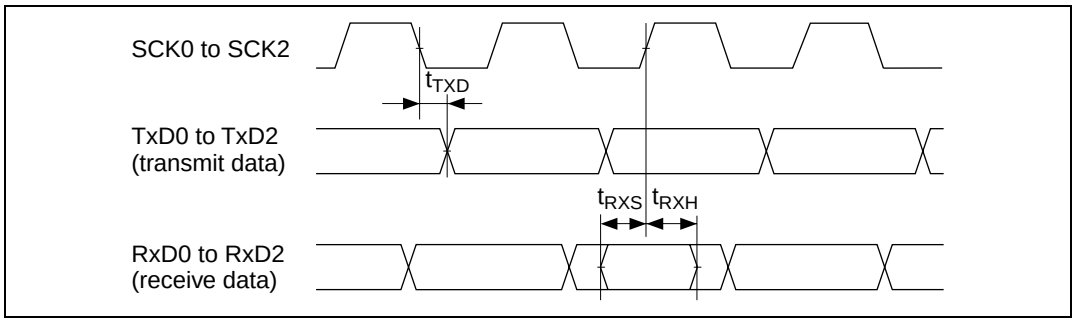


Figure 21.22 SCI Input/Output Timing: Synchronous Mode

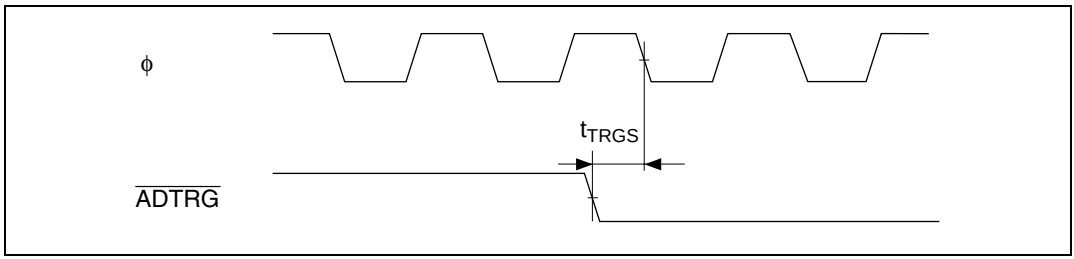


Figure 21.23 A/D Converter External Trigger Input Timing

21.1.4 A/D Conversion Characteristics

Table 21.10 A/D Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	Bit
Conversion time	—	—	8.1	µs
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±7.5	LSB
Offset error	—	—	±7.5	LSB
Full-scale error	—	—	±7.5	LSB
Quantization error	—	±0.5	—	LSB
Absolute accuracy	—	—	±8.0	LSB

21.1.5 D/A Conversion Characteristics

Table 21.11 D/A Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 8 MHz to 33 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	Bit	
Conversion time	—	—	10	µs	20 pF capacitive load
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ resistive load
	—	—	±2.0	LSB	4 MΩ resistive load

21.1.6 Flash Memory Characteristics

Table 21.12 Flash Memory Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = 0°C to 75°C (program/erase operating temperature range: regular specifications), T_a = 0°C to 85°C (program/erase operating temperature range: wide-range specifications)

Item		Symbol		Min	Typ	Max	Unit	Test Conditions
Programming time*1 *2 *4		t_P		—	10	200	ms/ 128 bytes	
Erase time*1 *3 *6		t_E		—	50	1000	ms/ 128 bytes	
Rewrite times		N_{WEC}		100*1	10000*2	—	Times	
Data retention time*3		t_{DRP}		10	—	—	Years	
Programming	Wait time after SWE bit setting*1	x		1	—	—	µs	
	Wait time after PSU bit setting*1	y		50	—	—	µs	
	Wait time after P bit setting*1 *4	z	z1	—	—	30	µs	1 ≤ n ≤ 6
			z2	—	—	200	µs	7 ≤ n ≤ 1000
			z3	—	—	10	µs	Additional program-ming wait
	Wait time after P bit clearing*1	α		5	—	—	µs	
	Wait time after PSU bit clearing*1	β		5	—	—	µs	
	Wait time after PV bit setting*1	γ		4	—	—	µs	
	Wait time after H'FF dummy write*1	ε		2	—	—	µs	
	Wait time after PV bit clearing*1	η		2	—	—	µs	
	Wait time after SWE bit clearing*1	θ		100	—	—	µs	
	Maximum number of writes*1 *4	N		—	—	1000*5	Times	

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Erasing	Wait time after SWE bit setting*1	x	1	—	—	µs	
	Wait time after ESU bit setting*1	y	100	—	—	µs	
	Wait time after E bit setting*1*6	z	—	—	10	µs	Erase time wait
	Wait time after E bit clearing*1	α	10	—	—	µs	
	Wait time after ESU bit clearing*1	β	10	—	—	µs	
	Wait time after EV bit setting*1	γ	20	—	—	µs	
	Wait time after H'FF dummy write*1	ε	2	—	—	µs	
	Wait time after EV bit clearing*1	η	4	—	—	µs	
	Wait time after SWE bit clearing*1	θ	100	—	—	µs	
	Maximum number of erases*1*6	N	—	—	100	Times	

Notes: 1. Follow the program/erase algorithms when making the time settings.

2. Programming time per 128 bytes. (Indicates the total time during which the P bit is set in flash memory control register 1 (FLMCR1). Does not include the program-verify time.)

3. Time to erase one block. (Indicates the time during which the E bit is set in FLMCR1. Does not include the erase-verify time.)

4. Maximum programming time

$$t_P(\max) = \sum_{i=1}^{N} \text{wait time after P bit setting (z)}$$

5. The maximum number of writes (N) should be set as shown below according to the actual set value of (z) so as not to exceed the maximum programming time (t_P(max)).

   The wait time after P bit setting (z) should be changed as follows according to the number of writes (n).

   Number of writes (n)

   $1 \leq n \leq 6$ z = 30 µs

   $7 \leq n \leq 1000$ z = 200 µs

   (Additional programming)

   Number of writes (n)

   $1 \leq n \leq 6$ z = 10 µs

6. For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (z) and the maximum number of erases (N):

   t_E(max) = Wait time after E bit setting (z) × maximum number of erases (N)

Appendix

A. I/O Port States in Each Pin State

Port Name	MCU Operating Mode*1	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port 1	1 to 7	T	T	Keep	Keep	I/O port
Port 2	1 to 7	T	T	Keep	Keep	I/O port
Port 3	1 to 7	T	T	Keep	Keep	I/O port
P47/DA1	1 to 7	T	T	[DAOE1 = 1] Keep [DAOE1 = 0] T	Keep	Input port
P46/DA0	1 to 7	T	T	[DAOE0 = 1] Keep [DAOE0 = 0] T	Keep	Input port
P45 to P40	1 to 7	T	T	T	T	Input port
P57/DA3	1 to 7	T	T	[DAOE3 = 1] Keep [DAOE3 = 0] T	Keep	Input port
P56/DA2	1 to 7	T	T	[DAOE2 = 1] Keep [DAOE2 = 0] T	Keep	Input port
P55, P54	1 to 7	T	T	T	T	Input port
P53 to P50	1 to 7	T	T	Keep	Keep	I/O port
Port 6	1 to 7	T	T	Keep	Keep	I/O port
Port 7	1 to 7	T	T	Keep	Keep	I/O port
Port 8	1 to 7	T	T	Keep	Keep	I/O port

Port Name	MCU Operating Mode*1	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PA7/A23 PA6/A22 PA5/A21	1 to 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A23 to A21 [Other than the above] I/O port
PA4/A20 PA3/A19 PA2/A18 PA1/A17	1, 2, 5, 6	L	T	[OPE = 0] T [OPE = 1] Keep	T	Address output A20 to A16
PA0/A16	3, 4, 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A20 to A16 [Other than the above] I/O port
Port B	1, 2, 5, 6	L	T	[OPE = 0] T [OPE = 1] Keep	T	Address output A15 to A8

Port Name	MCU Operating Mode*1	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port B	4	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A15 to A8 [Other than the above] I/O port
	3, 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A15 to A8 [Other than the above] I/O port
Port C	1, 2, 5, 6	L	T	[OPE = 0] T [OPE = 1] Keep	T	Address output A7 to A0
	4	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port

Port Name	MCU Operating Mode*1		Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port C	3, 7		T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port
Port D	1, 2, 4 to 6		T	T	T	T	D15 toD8
	3, 7		T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D15 to D8 [Other than the above] I/O port
Port E	1, 2, 4 to 6	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	T	T	D7 to D0
	3, 7	8-bit bus	T	T	Keep	Keep	I/O port
		16-bit bus	T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D7 to D0 [Other than the above] I/O port
PF7/ϕ	1, 2, 4 to 6		Clock output	T	[Clock output] H [Other than the above] Keep	[Clock output] Clock output [Other than the above] Keep	[Clock output] Clock output [Other than the above] Input port
	3, 7		T				

Port Name	MCU Operating Mode*1	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF6/$\overline{AS}$	1, 2, 4 to 6	H	T	[OPE = 0, $\overline{AS}$ output] T [OPE = 1, $\overline{AS}$ output] H [Other than the above] Keep	[$\overline{AS}$ output] T [Other than the above] Keep	[$\overline{AS}$ output] $\overline{AS}$ [Other than the above] I/O port
	3, 7	T				
PF5/$\overline{RD}$ PF4/$\overline{HWR}$	1, 2, 4 to 6	H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{RD}$, $\overline{HWR}$
	3, 7	T		[OPE = 0, $\overline{RD}$, $\overline{HWR}$ output] T [OPE = 1, $\overline{RD}$, $\overline{HWR}$ output] H [Other than the above] Keep	[$\overline{RD}$, $\overline{HWR}$ output] T [Other than the above] Keep	[$\overline{RD}$, $\overline{HWR}$ output] $\overline{RD}$, $\overline{HWR}$ [Other than the above] I/O port
PF3/$\overline{LWR}$	1, 2, 4 to 6	H	T	[OPE = 0, $\overline{LWR}$ output] T [OPE = 1, $\overline{LWR}$ output] H [Other than the above] Keep	[$\overline{LWR}$ output] T [Other than the above] Keep	[$\overline{LWR}$ output] $\overline{LWR}$ [Other than the above] I/O port
	3, 7	T				

Port Name	MCU Operating Mode*1	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF2	1 to 7	T	T	Keep	Keep	I/O port
PF1	1 to 7	T	T	Keep	Keep	I/O port
PF0/$\overline{\text{WAIT}}$	1 to 7	T	T	[$\overline{\text{WAIT}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT}}$ input] T [Other than the above] Keep	[$\overline{\text{WAIT}}$ input] $\overline{\text{WAIT}}$ [Other than the above] I/O port
PG6/$\overline{\text{BREQ}}$	1 to 7	T	T	[$\overline{\text{BREQ}}$ input] T [Other than the above] Keep	$\overline{\text{BREQ}}$ input $\overline{\text{BREQ}}$	[$\overline{\text{BREQ}}$ input] $\overline{\text{BREQ}}$ [Other than the above] I/O port
PG5/$\overline{\text{BACK}}$	1 to 7	T	T	[$\overline{\text{BACK}}$ output] T [Other than the above] Keep	$\overline{\text{BACK}}$	[$\overline{\text{BACK}}$ output] $\overline{\text{BACK}}$ [Other than the above] I/O port
PG4/ $\overline{\text{BREQO}}$	1 to 7	T	T	[$\overline{\text{BREQO}}$ output] T [Other than the above] Keep	[$\overline{\text{BREQO}}$ output] $\overline{\text{BREQO}}$ [Other than the above] Keep	[$\overline{\text{BREQO}}$ output] $\overline{\text{BREQO}}$ [Other than the above] I/O port
PG3/$\overline{\text{CS3}}$ PG2/$\overline{\text{CS2}}$ PG1/$\overline{\text{CS1}}$	1 to 7	T	T	[OPE = 0, $\overline{\text{CS}}$ output] T [OPE = 1, $\overline{\text{CS}}$ output] H [Other than the above] Keep	[$\overline{\text{CS}}$ output] T [Other than the above] Keep	[$\overline{\text{CS}}$ output] $\overline{\text{CS}}$ [Other than the above] I/O port

Port Name	MCU Operating Mode*1	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PG0/$\overline{CS0}$	1, 2, 5, 6	H	T	[OPE = 0, $\overline{CS}$ output]	[$\overline{CS}$ output]	[$\overline{CS}$ output]
	3, 4, 7	T		T	T	$\overline{CS}$
				[OPE = 1, $\overline{CS}$ output]	[Other than the above]	[Other than the above]
				H	Keep	I/O port
				[Other than the above]		
				Keep		
PH3/$\overline{CS7}$ PH2/$\overline{CS6}$ PH1/$\overline{CS5}$ PH0/$\overline{CS4}$	1 to 7	T	T	[OPE = 0, $\overline{CS}$ output]	[$\overline{CS}$ output]	[$\overline{CS}$ output]
				T	T	$\overline{CS}$
				[OPE = 1, $\overline{CS}$ output]	[Other than the above]	[Other than the above]
				H	Keep	I/O port
				[Other than the above]		
				Keep		

Legend:

L: Low level

H: High level

Keep: Input port becomes high-impedance, output port retains state

T: High impedance

DDR: Data direction register

OPE: Output port enable

Note: * Indicates the state after the currently executed bus cycle ends.

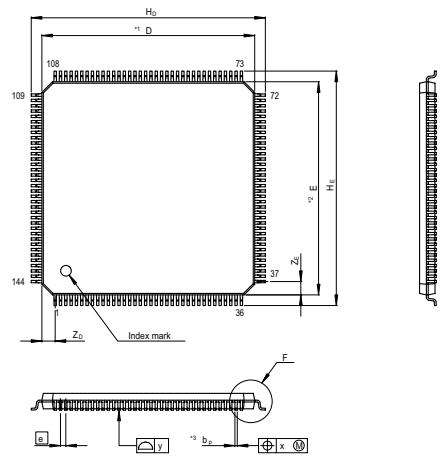
B. Product Lineup

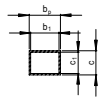
Product		Type Name	Model Marking	Package (Code)
H8S/2667	F-ZTAT version	HD64F2667	HD64F2667	144-pin LQFP (FP-144H)

C. Package Dimensions

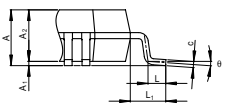
For package dimensions, dimensions described in Renesas Package data book have priority.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KC-A	FP-144H/FP-144HV	1.4g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	20	—
E	—	20	—
A_2	—	1.45	—
H_D	21.7	22.0	22.3
H_E	21.7	22.0	22.3
A	—	—	1.70
A_1	0.04	0.12	0.20
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.4	0.5	0.6
L_1	—	1.0	—

Figure C.1 Package Dimensions (FP-144H)

Index

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2668 Group, H8S/2667 F-ZTAT™

Publication Date:	1st Edition, September 2001
	Rev.3.00, February 22, 2006
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2668 Group, H8S/2667 F-ZTAT™ Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan