To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User's Manual

16

RENESAS

H8S/2117R Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer

H8S Family / H8S/2100 Series

H8S/2117R R4F2117R

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.2.00 2009.09

and careful attention to additional and different information to be disclosed by Renesas such as that disclos through our website. (http://www.renesas.com)

5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the informatio included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended applicatio Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which requir especially high quality and reliability such as safety systems, or equipment or systems for transportation anc traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communicat transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed bel
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers wh elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renes Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respe to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, a injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products o system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is ver high. You should implement safety measures so that Renesas products may not be easily detached from yo products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

vicinity of LSI, an associated shoot-through current flows internally, and malfunc
occur due to the false recognition of the pin state as an input signal. Unused pin
be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of regis
   settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, t
   of pins are not guaranteed from the moment when power is supplied until the re
   process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power
   function are not guaranteed from the moment when power is supplied until the p
   reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functi
   not access these addresses; the correct operation of LSI is not guaranteed if the
   accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal ha
   stable. When switching the clock signal during program execution, wait until the targ
   signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an extern
   oscillator) during a reset, ensure that the reset line is only released after full stab
   the clock signal. Moreover, when switching to a clock signal produced with an e
   resonator (or by an external oscillator) while program execution is in progress, w
   the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different part numbe
   that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part nu
   differ because of the differences in internal memory capacity and layout pattern.
   changing to products of different part numbers, implement a system-evaluation
   each of the products.

When designing an application system that includes this LSI, take all points to note account. Points to note are given in their contexts and at the final part of each sect in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier It does not cover all revised items. For details on the revised points, see the actual in the manual.

The following documents have been prepared for the H8S/2117R Group. Before usi the documents, please visit our web site to verify that you have the most up-to-date a version of the document.

Document Type	Contents	Document Title	Docu
Data Sheet	Overview of hardware and electrical characteristics	—	—
Hardware Manual	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	H8S/2117R Group Hardware Manual	This r
Software Manual	Detailed descriptions of the CPU and instruction set	H8S/2600 Series H8S/2000 Series Software Manual	REJ0
Application Note	Examples of applications and sample programs	The latest versions are available f web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

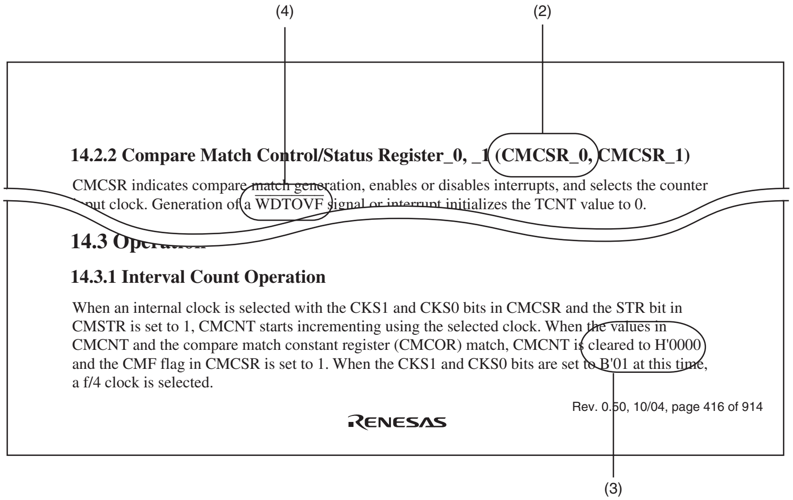
Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.

[Examples]	Binary:	B'11 or 11
	Hexadecimal:	H'EFA0 or 0xEFA0
	Decimal:	1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.

[Example] $\overline{\text{WDTOVF}}$



14.2.2 Compare Match Control/Status Register_0, _1 (CMCSR_0, CMCSR_1)

CMCSR indicates compare match generation, enables or disables interrupts, and selects the counter input clock. Generation of a $\overline{\text{WDTOVF}}$ signal or interrupt initializes the TCNT value to 0.

14.3 Operation

14.3.1 Interval Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and the compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CKS1 and CKS0 bits are set to B'01 at this time, a f/4 clock is selected.

Rev. 0.50, 10/04, page 416 of 914

RENESAS

Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

9	–	1	R	Reserved This bit is always read as 1.
	–	0		

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the conten manual.

(1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.

(2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "–".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.

(3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
–: The initial value is undefined

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable or both writing to and reading from the bit or field are impossible.
The notation is as follows:

R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

(5) Description
Describes the function of the bit or field and specifies the values for writing.

TPU	16-bit timer pulse unit
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communication interface adapter
bps	Bits per second
CRC	Cyclic redundancy check
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
IEBus	Inter Equipment Bus (IEBus is a trademark of NEC Electronics Corpo
I/O	Input/output
IrDA	Infrared Data Association
LSB	Least significant bit
MSB	Most significant bit
NC	No connection
PLL	Phase-locked loop
PWM	Pulse width modulation
SFR	Special function register
SIM	Subscriber Identity Module
UART	Universal asynchronous receiver/transmitter
VCO	Voltage-controlled oscillator

2.4.4 Condition-Code Register (CCR)

40

Table amended

Bit	Bit Name	Initial Value	R/W	Description
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be read or written by software usi ANDC, ORC, and XORC instructions.

7.2.15 Port F

(1) PF7/PWMU5A, PF6/PWMU4A, PF5/PWMU3A, PF4/PWMU2A

178

Title amended

7.2.19 Port J

Table 7.4 Available Output Signals and Settings in Each Port

189

Table amended

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
P4	6	PWX0_OE	PWX0		PWMX.DACR.OEA = 1
		PWMU4B_OE	PWMU4B		PWMU_B.PWMCONB.PW PWMU_B.PWMCOND.CN
	4	TMO1_OE	TMO1		Except TMR_1.TCSR.OS[3
		PWMU2B_OE	PWMU2B		PWMU_B.PWMCONB.PW PWMU_B.PWMCOND.CN

191

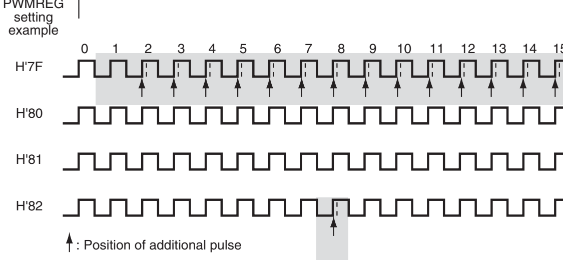
Table amended

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
PB	2	PWMU0B_OE	PWMU0B		PWMU_B.PWMCONB.PWM PWMU_B.PWMCONC.CNT

192

Table amended

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
PF	6	PWMU4A_OE	PWMU4A		PWMU_A.PWMCONB.PW PWMU_A.PWMCOND.CN
	4	PWMU2A_OE	PWMU2A		PWMU_A.PWMCONB.PW PWMU_A.PWMCOND.CN
	0	PWMU0A_OE	PWMU0A		PWMU_A.PWMCONB.PW PWMU_A.PWMCONC.CN



10.3.3 Timer I/O Control Register (TIOR)

Table 10.13 TIORL_0 (channel 0)

253

Table amended

Bit 3	Bit 2	Bit 1	Bit 0
IOC3	IOC2	IOC1	IOC0

11.3.6 TCM Status Register (TCMCSR)

312

Table amended

Bit	Bit Name	Initial Value	R/W	Description
0		0	R/W	Reserved The initial value should not be changed.

14.3.2 Timer Control/Status Register (TCSR)

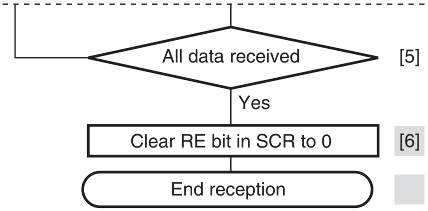
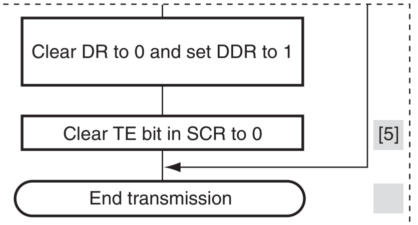
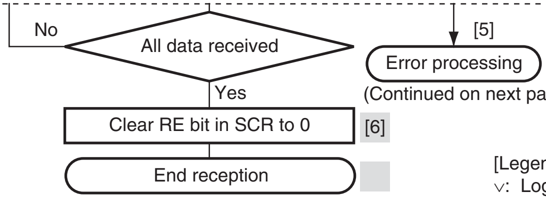
394

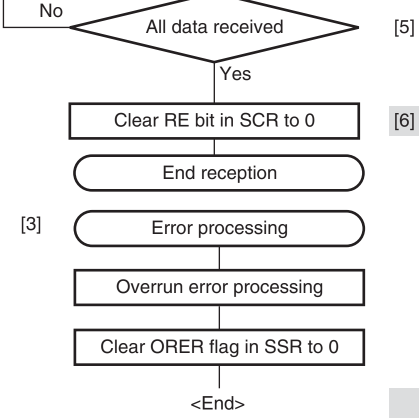
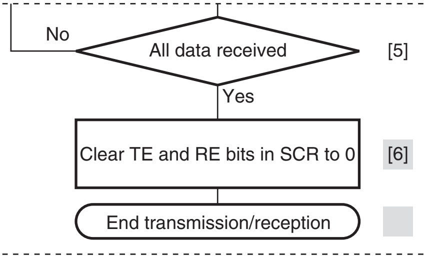
Table amended

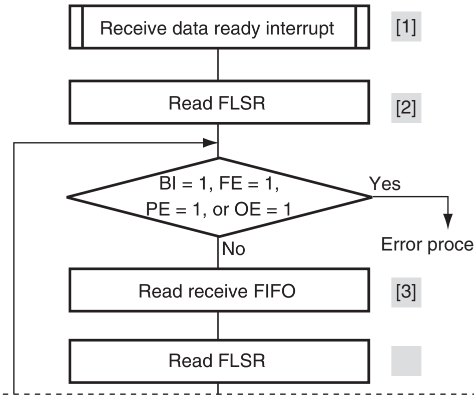
Bit	Bit Name	Initial Value	R/W	Description
4		0	R/W	Reserved The initial value should not be changed.

15.3 Register Descriptions Table 15.2 Register Configuration	406	Table amended

Channel	Register Name	Abbreviation	R/W	Initial Value	Ad
Channel 1	Serial mode register_1	SMR_1	R/W	H'00	H'F
Channel 2	Serial mode register_2	SMR_2	R/W	H'00	H'F

15.4.6 Serial Data Reception (Asynchronous Mode) Figure 15.9 Sample Serial Reception Flowchart (1)	433	Figure amended  All data received [5] Yes Clear RE bit in SCR to 0 [6] End reception
15.5.1 Multiprocessor Serial Data Transmission Figure 15.11 Sample Multiprocessor Serial Transmission Flowchart	437	Figure amended  Clear DR to 0 and set DDR to 1 Clear TE bit in SCR to 0 [5] End transmission
15.5.2 Multiprocessor Serial Data Reception Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1)	439	Figure amended  No All data received [5] Error processing (Continued on next pa Yes Clear RE bit in SCR to 0 [6] End reception [Leger ∨: Log

Synchronous Mode) Figure 15.19 Sample Serial Reception Flowchart		 No All data received [5] Yes Clear RE bit in SCR to 0 [6] End reception [3] Error processing Overrun error processing Clear ORER flag in SSR to 0 <End>
15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode) Figure 15.20 Sample Flowchart of Simultaneous Serial Transmission and Reception	448	Figure amended  No All data received [5] Yes Clear TE and RE bits in SCR to 0 [6] End transmission/reception

with Flow Control Figure 17.6 Example of Initialization Flowchart		[5] Select parity with the EPS and PEN bits in FLCR, a set the stop bit with the STOP bit in FLCR. Then, s the data length with the CLS1 and CLS0 bits in FL [6] Set the FIFOE bit in FFCR to 1 to enable the FIFO. Set the receive FIFO trigger level with the RCVRTF and RCVRTRIG0 bits in FFCR. Select the best trig level to prevent an overflow of the receive FIFO.
Figure 17.10 Example of Data Reception Flowchart	523	Figure amended  Receive data ready interrupt [1] Read FLSR [2] BI = 1, FE = 1, PE = 1, or OE = 1 Yes No Error proce Read receive FIFO [3] Read FLSR
17.6.2 FLCR Access During Serial Transmission and Reception	528	Description added

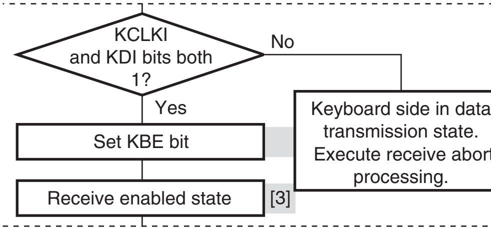
		These bits are always read as 1. The i should not be changed.
19.3.6 Keyboard Buffer Transmit Data Register (KBTR)	598	Table amended **Description** Keyboard Buffer Transmit Data Register 7 to 0 Initialized to H'FF at reset.
19.4.1 Receive Operation Figure 19.3 Sample Receive Processing Flowchart	565	Figure amended  KCLKI and KDI bits both 1? No Yes Set KBE bit Keyboard side in data transmission state. Execute receive abort processing. Receive enabled state [3]
19.4.9 KCLK Fall Interrupt Operation Figure 19.14 Example of KCLK Input Fall Interrupt Operation	608	Note amended Note: * The KBF setting timing is the same as the ti KBF setting and KCLK automatic I/O inhibit b generation in figure 19.11. When the KBF bit as the KCLK input fall interrupt flag, the auto inhibit function does not operate.
19.5.4 Medium-Speed Mode	614	Description amended In medium-speed mode, the PS2 operates with the m speed clock. For normal operation of the PS2, set the speed clock to a frequency of 300 kHz or higher.

Table 20.2 Register Configuration

Register Name	Abbreviation	Slave	Host	Value	Ad
Bidirectional data register 0MW	TWR0MW	R	W	H'00	H'F
Bidirectional data register 0SW	TWR0SW	W	R	H'00	H'F

20.3.1 Host Interface Control Registers 0 and 1 (HICR0 and HICR1)

- HICR1

625

Table amended

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
0	LSCIB	0	R/W	—	LSCI output Bit Controls LSCI output in combination bit . For details, refer to descriptic bit in HICR0.

20.3.2 Host Interface Control Registers 2 and 3 (HICR2 and HICR3)

- HICR2

627

Table amended

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
3	IBFIE3	0	R/W	—	IDR3 and TWR Receive Complete i Enables or disables IBFI3 interrupt t LSI). 0: Input data register IDR3 and TWF complete interrupt requests disab 1: [When TWRE = 0 in LADR3] Input data register (IDR3) receive interrupt requests enabled [When TWRE = 1 in LADR3] Input data register (IDR3) and TW complete interrupt requests enab

TWR0 and the state of TWR0SW is returned when t
reads TWR0MW. Attempts by the host to write to TW
are invalid.

For the registers selected from the host according to
address, see section 20.3.7, LPC Channel 3 Address
H and L (LADR3H and LADR3L).

20.3.12 Status Registers 1 to 4 (STR1 to STR4)

- STR4

644

Table amended

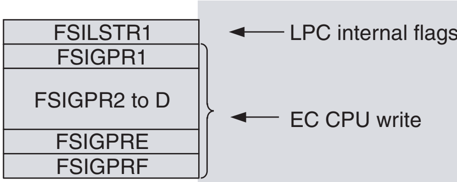
			R/W		
Bit	Bit Name	Initial Value	Slave	Host	Description
0	OBF4	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] When the host reads ODR4 in I/C When the slave writes 0 to the OE 1: [Setting condition] When the slave writes to ODR4

21.4.5 FSI Memory Cycle (LPC-SPI Command Transfer)

Figure 21.13 FSI Command Read (Example)

716

Figure amended



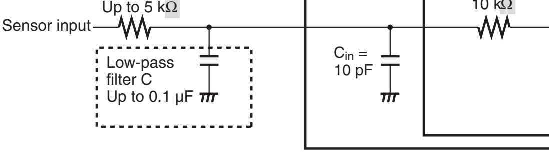
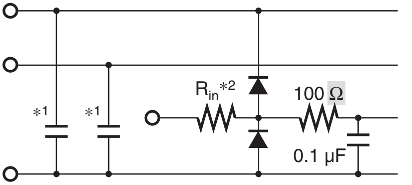
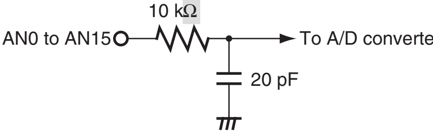
21.5 Reset Conditions

Table 21.8 Range of Initialization of FSI in Each Mode

723, 724

Table amended

Register Name		System Reset	LPC Reset	LPC Shutdown	LPC Abor
FSILSTR1	Bits 7, 6, 4, and 3	Initialized	Initialized	Retained	Retained
	Bit 2	Initialized	Initialized	Retained	Retained
	Bits 5, 1, and 0	Initialized	Retained	Retained	Retained
FSISTR	Bits 6 and 5	Initialized	Retained	Retained	Retained

Analog Input Circuit		 Up to 5 kΩ Sensor input Low-pass filter C Up to 0.1 µF C_{in} = 10 pF 10 kΩ
22.7.6 Notes on Noise Countermeasures Figure 22.6 Example of Analog Input Protection Circuit	743	Figure amended 
Figure 22.7 Analog Input Pin Equivalent Circuit	744	Figure amended 

		– The user boot memory MAT is initiated at a powe in user boot mode: 8K bytes • Three on-board programming modes Boot mode: Using the on-chip SCI-1, the user MA programmed/erased. In boot mode, the bit rate be host and this LSI can be adjusted automatically. User program mode: Using a desired interface, th MAT can be programmed/erased. User boot mode: The User boot program of The c interface can be made and The User MAT can be programmed.
24.2 Mode Transition Diagram Table 24.1 Differences between Boot Mode, User Program Mode, and Programmer Mode	751	Note replaced Notes: 2. First, the reset vector is fetched from the e program storage MAT. After the flash mem related registers are checked, the reset ve fetched from the user boot MAT.
	752	Description amended • The user boot MAT can be programmed or erased boot mode and programmer mode. • In boot mode, the user boot MAT are totally erased the user MAT or user boot MAT can be programme means of commands. Note that the contents of the cannot be read until this state. Boot mode can be used for programming only the b and then programming the user MAT in user boot Another way is to program only the user MAT since mode is not used. • In user boot mode, boot operation of the optional in can be performed with mode pin settings different fro user program mode.

4	MS4	0	R/W*2
3	MS3	0/1*1	R/W*2
2	MS2	0	R/W*2
1	MS1	0/1*1	R/W*2
0	MS0	0	R/W*2

is written. The MAT is switched by wr
FMATS. To switch the MAT, make sur
24.10, Switching between User MAT a
MAT. (The user boot MAT cannot be p
user program mode even if the user b
selected by FMATS. The user boot MA
programmed in boot mode or program

H'AA: User boot MAT is selected (use
when the value of these bits is o
Initial value when initiated in use

H'00: Initial value when initiated in a m
user boot mode (user MAT is se

[Programmable condition]

Execution state in the on-chip RAM

Note added

Notes: *1 The value is 1 in user boot mode and 0 o

24.8.4 Storable Areas for On-Chip Program and Program Data | 793

Description amended

- In an operating mode in which the external addres not accessible, such as single-chip mode, the requ procedure programs should be transferred to th RAM before programming/erasing starts (downloa determined).
- The flash memory is not accessible during programming/erasing. Programming/erasing is exe the program downloaded to the on-chip RAM. The procedure program that initiates operation shou stored in the on-chip RAM other than the flash me

Table 24.10 Usable Area for Programming in User Program Mode | 794

Table amended

	Storable/Executable Area		Sel
Item	**On-Chip RAM**	**User MAT**	**User MAT**
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
Operation for disabling interrupts	O	O	O

Item	On-Chip RAM	User Boot MAT	User MAT	User Boot MAT	S
Determination of initialization result	O	O		O	
Initialization error processing	O	O		O	
Disabling interrupts	O	O		O	

Table 24.13 Usable Area for Erasure in User Boot Mode	797	Table amended

	Storable/Executable Area			Selected MAT	
Item	On-Chip RAM	User Boot MAT	User MAT	User Boot MAT	E P S
Determination of initialization result	O	O		O	
Initialization error processing	O	O		O	
Disabling interrupts	O	O		O	

24.12 Standard Serial Communication Interface Specifications for Boot Mode	802	Description amended

2. Inquiry/selection state

In this state, the boot program responds to inquiry commands from the host. The device name, clock and bit rate are selected. After selection of these s the program is made to enter the programming/era by the command for a transition to the programmin state. The program transfers the libraries required erasure to the on-chip RAM and erases the user M user boot MATs before the transition.

(3) Inquiry and Selection States (f) Operating Clock Frequency Inquiry	811	Description amended • Minimum value of operating clock frequency (two minimum value of the divided clock frequency The minimum and maximum values of the operati frequency represent the values in MHz, valid to th hundredths place of MHz, and multiplied by 100. the value is 17.00 MHz, it will be 2000, which is H • Maximum value (two bytes): Maximum value amo divided clock frequencies. There are as many pairs of minimum and maximu as there are operating clock frequencies.
(8) Programming/Erasing State 3. Programming/Erasing State Information (c) 128-Byte Programming	823	Description amended • ERROR: (one byte): Error code H'11: Checksum Error H'2A: Address Error The address is not within the specified MA H'53: Programming error A programming error has occurred and pr cannot be continued.
(f) Memory Read	826	Description amended • Area (one byte) H'00: User boot MAT H'01: User MAT An address error occurs when the area se incorrect.

Port 7	$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$
Ports A, G, PE4, PE2 to PE0, P97, and P52	$V_{CC} \times 0.7$	—	5.5

Table 28.2 DC Characteristics (4) Using FSI Function

961

Figure amended

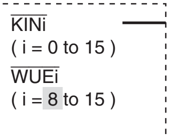
Item		Symbol	Min.	Typ.	Max.	Unit	T
Output high voltage	PB7 to PB4	V_{OH}	$V_{CC} - 0.5$	—	—	V	I
			$V_{CC} - 1.0$	—	—		I
Output low voltage		V_{OL}	—	—	0.4		I

28.3.2 Control Signal Timing

Figure 28.8 Interrupt Input Timing

967

Figure amended



All trademarks and registered trademarks are the property of their respective owners.

Section 12 16-Bit Duty Period Measurement Timer (TDP) ....

FIFO, an I^{2}C bus interface, an A/D converter, and various types of timers. Together, the realize low-cost system configurations. The power consumption of these modules is kep dynamically by power-down modes. The on-chip ROM is a flash memory (F-ZTAT^{TM}*) capacity of 160 Kbytes.

Note: * F-ZTAT^{TM} is a trademark of Renesas Technology Corp.

1.1.1 Applications

Examples of the applications of this LSI include PC peripheral equipment, office autom equipment, and industrial equipment.

CPU	CPU	• 16-bit high-speed H8S/2600 CPU (CISC type) Upward-compatibility with H8/300, H8/300H, and H8S C object level • General-register architecture (sixteen 16-bit general reg • Eight addressing modes • 4-Gbyte address space Program: 4 Gbytes available Data: 4 Gbytes available • 69 basic instructions (bit arithmetic and logic instruction multiply and divide instructions, bit manipulation instruc multiply-and-accumulate instructions, and others) • Minimum instruction execution time: 50.0 ns (for an AD instruction while system clock ϕ = 20 MHz and V_{CC} = 3.0 to 3.6 V) • On-chip multiplier (16 × 16 → 32 bits) • Supports multiply-and-accumulate instructions (16 × 16 + 32 → 32 bits)
	Operating mode	• Advanced and single-chip modes

		Note: MD0 is not available as a pin and is internally fixed • Power-down state (transition to the power-down state the SLEEP instruction)
Interrupt (source)	Interrupt controller	• 41 external interrupt pins (NMI, $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ ($\overline{\text{ExIRQ}}$ $\overline{\text{ExIRQ6}}$), $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) • 66 internal interrupt sources • Two interrupt control modes (specified by the system register) • Two levels of interrupt priority orders specifiable (by s interrupt control register) • Independent vector addresses
Clock	Clock pulse generator (CPG)	• Two clock generation circuits • Clock pulse generator and subclock input circuit System clock (φ) synchronization: 8 to 20 MHz • Five power-down modes: Medium-speed mode, sleep watch mode, software standby mode, and module sto
A/D converter	A/D converter (ADC)	• 10-bit resolution × 16 input channels • Sample and hold function included • Conversion time: 4 μs per channel (with A/D conversi ADCLK at 10 MHz operation) • Two operating modes: single mode and scan mode • Three methods to start A/D conversion: software and (TPU/TMR) triggers

	16-bit timer pulse unit (TPU)	• 16 bits × three channels • Selectable from eight counter input clocks for each cha • Maximum 8-pulse inputs/outputs • The following operations can be set. — Counter clear operation — Multiple timer counters (TCNT) can be written to simultaneously. — Simultaneous clearing by compare match and input possible — Register simultaneous input/output possible by cou synchronous operation — Maximum of 7-phase PWM output possible by com with synchronous operation • Supports buffer operation and phase counting mode (tv phase encoder input) for some channels • Supports input capture function • Supports output compare function (waveform output at compare match)
	16-bit cycle measurement timer (TCM)	• 16 bits × four channels • Selectable from seven clocks: six internal clocks and or external clock • Capable of measuring the periods of input waveforms
	16-bit duty period measurement timer (TDP)	• 16 bits × three channels • Selectable from seven clocks: six internal clocks and or external clock • Capable of measuring the periods and pulse width of ir waveforms

	communication interface with FIFO (SCIF)	• 16-stage FIFO buffers for transmission and reception • Full-duplex communication capability • On-chip baud rate generator allows any bit rate to be s • Direct control from the LPC host
	Serial communication interface (SCI)	• Two channels (choice of asynchronous or clocked syn serial communication mode) • Full-duplex communication capability • Selection of the desired bit rate and LSB-first or MSB- transfer
Smart card/ SIM		• The SCI module supports a smart card (SIM) interface
High-performance communication	I^2C bus interface (IIC)	• Three channels (two channels are switchable between and output pin) • Two types of communication formats • I^2C bus format: addressing format with an acknowledg master/slave operation • Clocked synchronous serial format: non-addressing fo without an acknowledge bit, for master operation only
	Keyboard buffer control unit (PS2)	• Four channels • Conforms to PS/2 interface specifications • Direct bus drive • Interrupt and error detection
	LPC interface (LPC)	• Four channels • Serial transfer of cycle type, address, and data in synchronization with the PCI clock • Supports LPC interface I/O read and I/O write cycles • Supports the shutdown function (LPCPD) of the LPC i

	• Input/output pins: 112 pins (TFP-144V and TLP-145V) 128 pins (BP-176V) • 76 pull-up resistors for TFP-144V and TLP-145V, and 8 up resistors for BP-176V • 40 pins with LED drive capability • 24 on-chip noise cancellers
Package	• 144-pin thin QFP package (PTQP0144LC-A) (old code: TFP-144V, package dimensions: 16 × 16 mr pitch: 0.40 mm) • 176-pin BGA package (PLBG0176GA-A) (old code: BP-176V, package dimensions: 13 × 13 mm pitch: 0.80 mm) • 145-pin TLP package (PTLG0145JB-A) (package dimensions: 9 × 9 mm, pin pitch: 0.65 mm) • Lead- (Pb-) free version
Operating frequency/ Power supply voltage	• Operating frequency: 8 to 20 MHz • Power supply voltage: Vcc = 3.0 to 3.6 V, AVcc = 3.0 to • Supply current: 25 mA (typ.) (Vcc = 3.3 V, AVcc = 3.3 V, ϕ = 20 MHz)
Operating peripheral temperature (°C)	• –20 to +75°C (regular specifications)

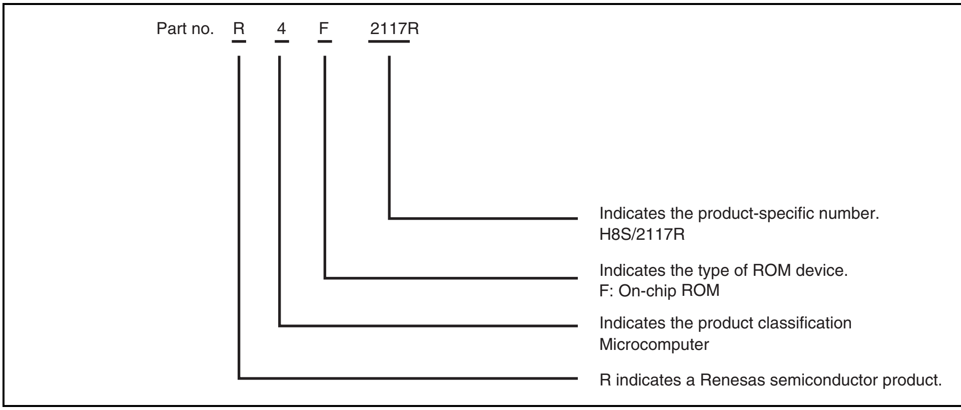
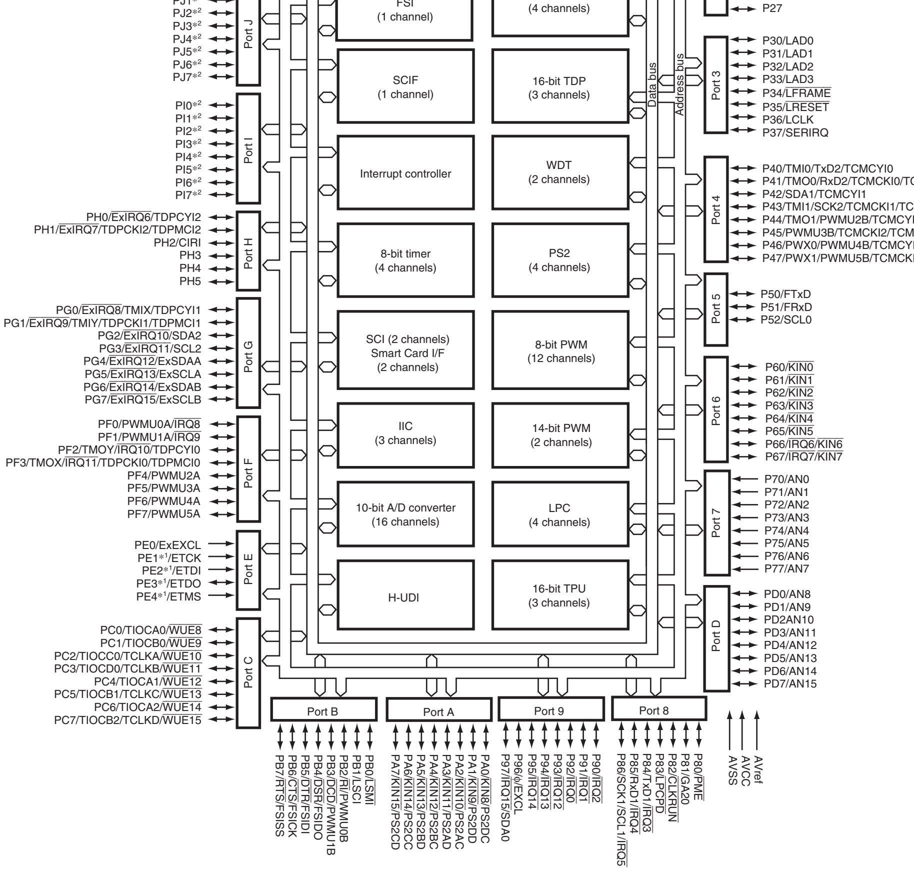
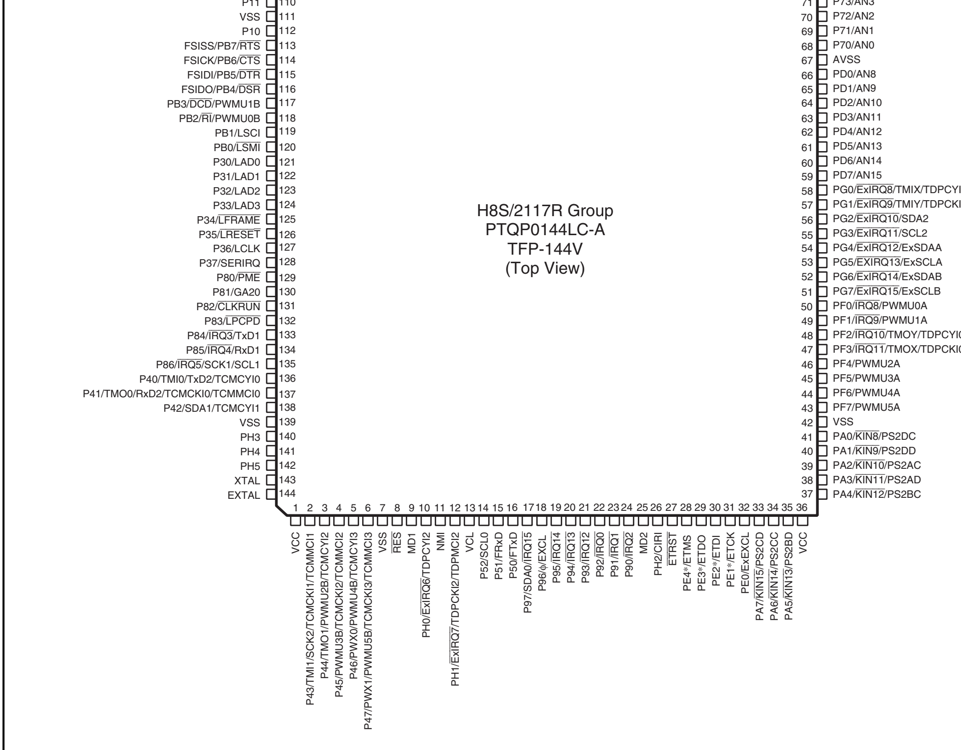

Part no. R 4 F 2117R
Indicates the product-specific number.
H8S/2117R
Indicates the type of ROM device.
F: On-chip ROM
Indicates the product classification
Microcomputer
R indicates a Renesas semiconductor product.

Figure 1.1 How to Read the Product Name Code



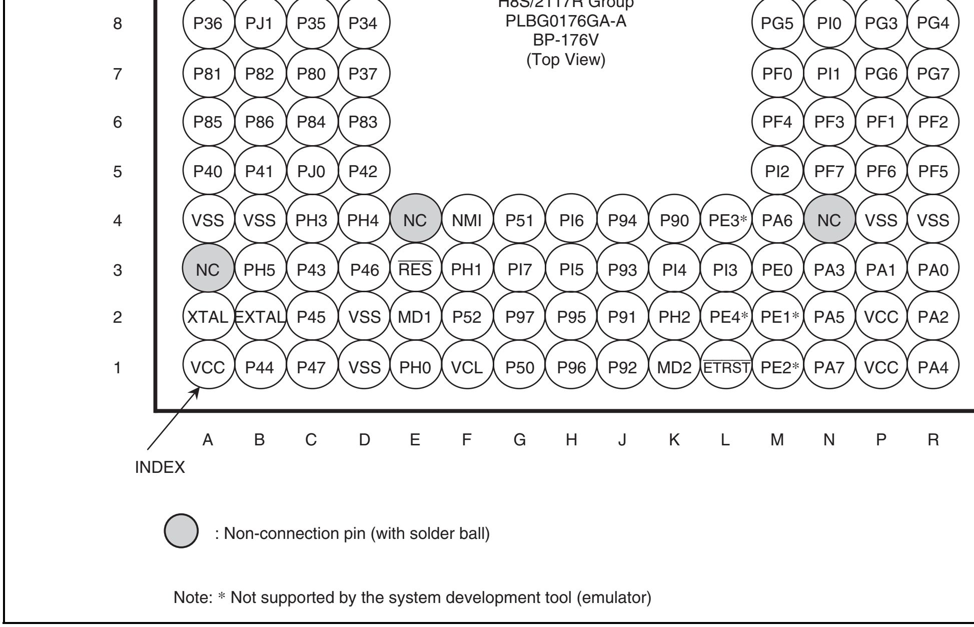
Notes: 1. Not supported by the system development tool (emulator)
2. Not supported by TFP-144 V and TLP-145 V

Figure 1.2 Internal Block Diagram



Note: * Not supported by the system development tool (emulator)

Figure 1.3 Pin Assignments (TFP-144V)



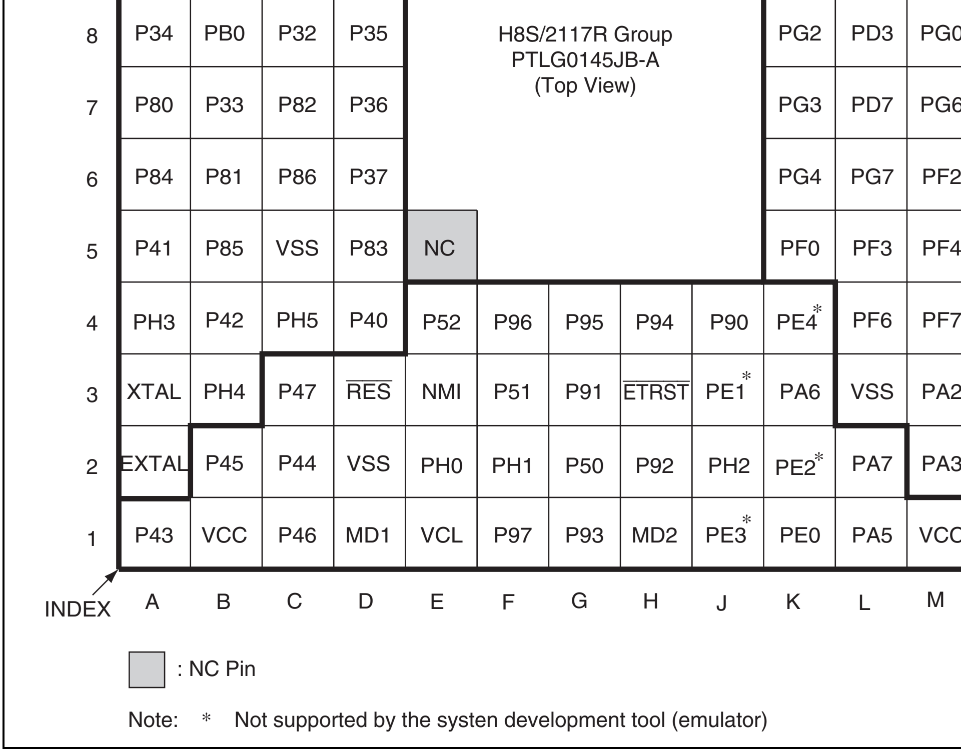
H8S/2117R Group
PLBG0176GA-A
BP-176V
(Top View)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R
8	P36	PJ1	P35	P34								PG5	PI0	PG3	PG4
7	P81	P82	P80	P37								PF0	PI1	PG6	PG7
6	P85	P86	P84	P83								PF4	PF3	PF1	PF2
5	P40	P41	PJ0	P42								PI2	PF7	PF6	PF5
4	VSS	VSS	PH3	PH4	NC	NMI	P51	PI6	P94	P90	PE3*	PA6	NC	VSS	VSS
3	NC	PH5	P43	P46	$\overline{\text{RES}}$	PH1	PI7	PI5	P93	PI4	PI3	PE0	PA3	PA1	PA0
2	XTAL	EXTAL	P45	VSS	MD1	P52	P97	P95	P91	PH2	PE4*	PE1*	PA5	VCC	PA2
1	VCC	P44	P47	VSS	PH0	VCL	P50	P96	P92	MD2	$\overline{\text{ETRST}}$	PE2*	PA7	VCC	PA4

● : Non-connection pin (with solder ball)

Note: * Not supported by the system development tool (emulator)

Figure 1.4 Pin Assignments (BP-176V)



	A	B	C	D	E	F	G	H	J	K	L	M
8	P34	PB0	P32	P35	H8S/2117R Group PTLG0145JB-A (Top View)					PG2	PD3	PG0
7	P80	P33	P82	P36						PG3	PD7	PG6
6	P84	P81	P86	P37						PG4	PG7	PF2
5	P41	P85	VSS	P83	NC					PF0	PF3	PF4
4	PH3	P42	PH5	P40	P52	P96	P95	P94	P90	PE4*	PF6	PF7
3	XTAL	PH4	P47	$\overline{\text{RES}}$	NMI	P51	P91	$\overline{\text{ETRST}}$	PE1*	PA6	VSS	PA2
2	EXTAL	P45	P44	VSS	PH0	PH1	P50	P92	PH2	PE2*	PA7	PA3
1	P43	VCC	P46	MD1	VCL	P97	P93	MD2	PE3*	PE0	PA5	VCC

▨ : NC Pin

Note: * Not supported by the systen development tool (emulator)

Figure 1.5 Pin Assignments (TLP-145V)

4	C2	B2	P45/PWMU3B/TCMCKI2/TCMMCI2
5	D3	C1	P46/PWX0/PWMU4B/TCMCYI3
6	C1	C3	P47/PWX1/PWMU5B/TCMCKI3/TCMMCI3
7	D2	D2	VSS
—	E4	—	NC
8	E3	D3	$\overline{\text{RES}}$
—	D1	—	VSS
9	E2	D1	MD1
10	E1	E2	PH0/$\overline{\text{ExIRQ6}}$/TDPCYI2
11	F4	E3	NMI
12	F3	F2	PH1/$\overline{\text{ExIRQ7}}$/TDPCKI2/TDPMCI2
13	F1	E1	VCL
14 (N)	F2 (N)	E4 (N)	P52/SCL0
15	G4	F3	P51/FRxD
—	G3 (N)	—	PI7
16	G1	G2	P50/FTxD
17 (N)	G2 (N)	F1 (N)	P97/SDA0/$\overline{\text{IRQ15}}$
—	H4 (N)	—	PI6
—	H3 (N)	—	PI5
18	H1	F4	P96/ϕ/EXCL
19	H2	G4	P95/$\overline{\text{IRQ14}}$
20	J4	H4	P94/$\overline{\text{IRQ13}}$
21	J3	G1	P93/$\overline{\text{IRQ12}}$

—	L3 (N)	—	PI3
27	L1	H3	$\overline{\text{ETRST}}$
28 (T)	L2 (T)	K4 (T)	PE4*/ETMS
29	L4	J1	PE3*/ETDO
30 (T)	M1 (T)	K2 (T)	PE2*/ETDI
31 (T)	M2 (T)	J3 (T)	PE1*/ETCK
32 (T)	M3 (T)	K1 (T)	PE0/ExEXCL
33 (N)	N1 (N)	L2 (N)	PA7/$\overline{\text{KIN15}}$/PS2CD
34 (N)	M4 (N)	K3 (N)	PA6/$\overline{\text{KIN14}}$/PS2CC
35 (N)	N2 (N)	L1 (N)	PA5/$\overline{\text{KIN13}}$/PS2BD
36	P1	M1	VCC
—	P2	—	VCC
37 (N)	R1 (N)	N2 (N)	PA4/$\overline{\text{KIN12}}$/PS2BC
38 (N)	N3 (N)	M2 (N)	PA3/$\overline{\text{KIN11}}$/PS2AD
39 (N)	R2 (N)	M3 (N)	PA2/$\overline{\text{KIN10}}$/PS2AC
40 (N)	P3 (N)	N1 (N)	PA1/$\overline{\text{KIN9}}$/PA2DD
—	N4	—	NC
41 (N)	R3 (N)	N3 (N)	PA0/$\overline{\text{KIN8}}$/PA2DC
42	P4	L3	VSS
—	M5 (N)	—	PI2
—	R4	—	VSS
43	N5	M4	PF7/PWMU5A
44	P5	L4	PF6/PWMU4A

—	N7 (N)	—	PI1
51 (N)	R7 (N)	L6 (N)	PG7/$\overline{\text{ExIRQ15}}$/ExSCLB
52 (N)	P7 (N)	M7 (N)	PG6/$\overline{\text{ExIRQ14}}$/ExSDAB
53 (N)	M8 (N)	N6 (N)	PG5/$\overline{\text{ExIRQ13}}$/ExSCLA
—	N8 (N)	—	PI0
54 (N)	R8 (N)	K6 (N)	PG4/$\overline{\text{ExIRQ12}}$/ExSDAA
55 (N)	P8 (N)	K7 (N)	PG3/$\overline{\text{ExIRQ11}}$/SCL2
—	M9 (N)	—	NC
56 (N)	N9 (N)	K8 (N)	PG2/$\overline{\text{ExIRQ10}}$/SDA2
57 (N)	R9 (N)	N7 (N)	PG1/$\overline{\text{ExIRQ9}}$/TMIY/TDPCKI1/TDPMCI1
58 (N)	P9 (N)	M8 (N)	PG0/$\overline{\text{ExIRQ8}}$/TMIX/TDPCYI1
59	M10	L7	PD7/AN15
60	N10	K9	PD6/AN14
61	R10	N8	PD5/AN13
62	P10	M9	PD4/AN12
63	N11	L8	PD3/AN11
64	R11	K10	PD2/AN10
65	P11	N9	PD1/AN9
66	M11	M10	PD0/AN8
67	R12	L9	AVSS
—	P12	—	AVSS
68	N12	N10	P70/AN0
69	R13	M11	P71/AN1

75	P15	L12	P77/AN7
76	N14	M12	AVCC
—	M13	—	NC
—	N15	—	AVCC
77	M14	L11	AVref
—	L12	E5	NC
—	M15	—	AVref
78	L13	L13	P60/$\overline{\text{KIN0}}$
79	L14	K12	P61/$\overline{\text{KIN1}}$
80	L15	K11	P62/$\overline{\text{KIN2}}$
81	K12	J12	P63/$\overline{\text{KIN3}}$
82	K13	K13	P64/$\overline{\text{KIN4}}$
—	K15	—	PJ7
83	K14	J10	P65/$\overline{\text{KIN5}}$
84	J12	J11	P66/$\overline{\text{IRQ6}}$/$\overline{\text{KIN6}}$
85	J13	H12	P67/$\overline{\text{IRQ7}}$/$\overline{\text{KIN7}}$
86	J15	H10	VCC
—	J14	—	PJ6
87	H12	J13	PC7/TIOCB2/TCLKD/$\overline{\text{WUE15}}$
88	H13	H11	PC6/TIOCA2/$\overline{\text{WUE14}}$
89	H15	G12	PC5/TIOCB1/TCLKC/$\overline{\text{WUE13}}$
90	H14	G10	PC4/TIOCA1/$\overline{\text{WUE12}}$
91	G12	H13	PC3/TIOCD0/TCLKB/$\overline{\text{WUE11}}$

96	F14	E10	P27
97	E13	F13	P26
98	E15	E12	P25
99	E14	E13	P24
100	E12	F11	P23
101	D15	D12	P22
102	D14	E11	P21
103	D13	D13	P20
104	C15	D10	P17
105	D12	C12	P16
106	C14	C13	P15
107	B15	D11	P14
108	B14	B13	P13
109	A15	A12	P12
110	C13	A13	P11
—	A14	—	PJ4
111	B13	B11	VSS
—	C12	—	PJ3
—	A13	—	VSS
112	B12	B12	P10
113	D11	A11	PB7/$\overline{\text{RTS}}$/FSISS
114	A12	C11	PB6/$\overline{\text{CTS}}$/FSICK
115	C11	B10	PB5/$\overline{\text{DTR}}$/FSIDI

121	D9	A9	P30/LAD0
122	C9	D9	P31/LAD1
123	A9	C8	P32/LAD2
124	B9	B7	P33/LAD3
125	D8	A8	P34/$\overline{\text{LFRAME}}$
126	C8	D8	P35/$\overline{\text{LRESET}}$
127	A8	D7	P36/LCLK
—	B8	—	PJ1
128	D7	D6	P37/SERIRQ
129	C7	A7	P80/$\overline{\text{PME}}$
130	A7	B6	P81/GA20
131	B7	C7	P82/$\overline{\text{CLKRUN}}$
132	D6	D5	P83/$\overline{\text{LPCPD}}$
133	C6	A6	P84/$\overline{\text{IRQ3}}$/TxD1
134	A6	B5	P85/$\overline{\text{IRQ4}}$/RxD1
135 (N)	B6 (N)	C6 (N)	P86/$\overline{\text{IRQ5}}$/SCK1/SCL1
—	C5	—	PJ0
136	A5	D4	P40/TMI0/TxD2/TCMCYI0
137	B5	A5	P41/TMO0/RxD2/TCMCKI0/TCMMCI0
138 (N)	D5 (N)	B4 (N)	P42/SDA1/TCMCYI1
139	A4	C5	VSS
—	B4	—	VSS
140	C4	A4	PH3

tolerance.

(T) in Pin No. indicates the pin has 5 V input tolerance.

* This pin is not supported by the system development tool (emulator).

	VCL	13	F1	E1	Input	External capacitance pin internal step-down powe this pin to VSS through a capacitor (that is located pin) to stabilize internal s power.
	VSS	7, 42, 95, 111, 139	D1, D2, P4, R4, F12, F13, B13, A13, A4, B4	D2, L3, F10, B11, C5	Input	Ground pins. Connect al pins to the system powe (0 V).
Clock	XTAL	143	A2	A3	Input	For connection to a cryst resonator. An external cl supplied from the EXTAL an example of crystal res connection, see section Pulse Generator.
	EXTAL	144	B2	A2	Input	
	ϕ	18	H1	F4	Output	Supplies the system cloc external devices.
	EXCL	18	H1	F4	Input	32.768 kHz external sub should be supplied. To w the external clock is inpu selected from the EXCL ExEXCL pin.
	ExEXCL	32	M3	K1	Input	
Operating mode control	MD2 MD1	25 9	K1 E2	H1 D1	Input	These pins set the opera Inputs at these pins shou changed during operatio
System control	$\overline{\text{RES}}$	8	E3	D3	Input	Reset pin. When this pin chip is reset.

		J2, J1	G3, J4		
	$\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$	51 to 58, 12, 10	R7, P7, M8, R8, P8, N9, R9, P9, F3, E1	L6, M7, N6, K6, K7, K8, N7, M8, F2, E2	Input	
H-UDI	$\overline{\text{ETRST}}$*[2]	27	L1	H3	Input	Interface pins for emulat
	ETMS	28	L2	K4	Input	Reset by holding the $\overline{\text{ET}}$ low level regardless of th activation. At this time, th pin should be held low le clocks of ETCK. Then, to the H-UDI, the $\overline{\text{ETRST}}$ p set to high level and the ETMS, and ETDI should appropriately. In the nor operation without activat UDI, pins ETCK, ETMS, ETDO should be pulled u level. The $\overline{\text{ETRST}}$ pin is inside the chip.
	ETDO	29	L4	J1	Output	
	ETDI	30	M1	K2	Input	
	ETCK	31	M2	J3	Input	
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMO0 TMO1 TMOX TMOY	137 3 47 48	B5 B1 N6 R6	A5 C2 L5 M6	Output	Waveform output pins w compare function
	TMI0 TMI1 TMIX TMIY	136 2 58 57	A5 C3 P9 R9	D4 A1 M8 N7	Input	Counter event input and input pins

	TIOCB1	89	H15	G12	Output	compare output/PW pins for TGRA_1 an
	TIOCA2 TIOCB2	88 87	H13 H12	H11 J13	Input/ Output	Input capture input/o compare output/PW pins for TGRA_2 an
16-bit cycle measurement timer (TCM)	TCMCKI3 to TCMCKI0	6, 4, 2, 137	C1, C2, C3, B5	C3, B2, A1, A5	Input	Timer external clock
	TCMMCI3 to TCMMCI0	6, 4, 2, 137	C1, C2, C3, B5	C3, B2, A1, A5	Input	Cycle measurement input pins
	TCMCYI3 to TCMCYI0	5, 3, 138, 136	D3, B1, D5, A5	C1, C2, B4, D4	Input	Timer input capture
16-bit duty period measurement timer (TDP)	TDPCKI2 to TDPCKI0	12, 57, 47	F3, R9, N6	F2, N7, L5	Input	Timer external clock
	TDPMCI2 to TDPMCI0	12, 57, 47	F3, R9, N6	F2, N7, L5	Input	Cycle measurement input pins
	TDPCYI2 to TDPCYI0	10, 58, 47	E1, P9, R6	E2, M8, M6	Input	Timer input capture
8-bit PWM timer U (PWMU)	PWMU5A to PWMU0A PWMU5B to PWMU0B	43 to 46, 49, 50, 6 to 3, 117, 118	N5, P5, R5, M6, P6, M7, C1, D3, C2, B1, A11, D10	M4, L4, N4, M5, N5, K5, C3, C1, B2, C2, A10, B9	Output	PWM timer pulse ou

						SCK2 is NMOS push
Keyboard buffer control unit (PS2)	PS2AC PS2BC PS2CC PS2DC	39 37 34 41	R2 R1 M4 R3	M3 N2 K3 N3	Input/ Output	Synchronous clock input/output pins for keyboard buffer cont
	PS2AD PS2BD PS2CD PS2DD	38 35 33 40	N3 N2 N1 P3	M2 L1 L2 N1	Input/ Output	Data input/output pir keyboard buffer cont
Keyboard control	$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$	33 to 35, 37 to 41, 85 to 78	N1, M4, N2, R1, N3, R2, P3, R3, J13, J12, K14, K13, K12, L15, L14, L13	L2, K3, L1, N2, M2, M3, N1, N3, H12, J11, J10, K13, J12, K11, K12, L13	Input	Input pins for matrix Normally, $\overline{\text{KIN15}}$ to K function as key scan and P17 to P10 and P20 function as key outputs. Thus, comp a maximum of 16 ou inputs, a 256-key ma be configured.
	$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	87 to 94	H12, H13, H15, H14, G12, G13, G15, G14	J13, H11, G12, G10, H13, F12, G13, G11	Input	Wake-up event input Same wake up as ke up can be performed various sources.

	$\overline{\text{CTS}}$	114	A12	C11	Input	Transmission pen input pin
	$\overline{\text{RTS}}$	113	D11	A11	Output	Transmission requ pin
LPC Interface (LPC)	LAD3 to LAD0	124 to 121	B9, A9, C9, D9	B7, C8, D9, A9	Input/ Output	LPC command, ad and data input/out
	$\overline{\text{LFRAME}}$	125	D8	A8	Input	Input pin indicating cycle start and for termination of an a LPC cycle
	$\overline{\text{LRESET}}$	126	C8	D8	Input	Input pin indicating reset
	LCLK	127	A8	D7	Input	LPC clock input pi
	SERIRQ	128	D7	D6	Input/ Output	LPC serial host int (HIRQ1 to HIRQ15 input/output pin
	LSCI, $\overline{\text{LSMI}}$, $\overline{\text{PME}}$	119, 120, 129	A10, B10, C7	C9, B8, A7	Input/ Output	LPC auxiliary outp Functionally, they general I/O ports.
	GA20	130	A7	B6	Input/ Output	GATE A20 control output pin. Output monitoring input is
	$\overline{\text{CLKRUN}}$	131	B7	C7	Input/ Output	Input/output pin th requests the start operation when LC stopped.
	$\overline{\text{LPCPD}}$	132	D6	D5	Input	Input pin that cont module shutdown.

converter	AN0	75 to 68	R10, P10, N11, R11, P11, M11, P15, N13, R15, P14, R14, P13, R13, N12	M9, L8, K10, N9, M10, L12, N13, M13, N12, N11, L10, M11, N10		
	AVCC	76	N14, N15	M12	Input	Analog power supp the A/D converter. When the A/D conv not used, this pin sh connected to the sy power supply (+3 V
	AVref	77	M14, M15	L11	Input	Reference power su for the A/D convert When the A/D conv not used, this pin sh connected to the sy power supply (+3 V
	AVSS	67	R12, P12	L9	Input	Ground pin for the A converter. This pin connected to the sy power supply (0 V).

	SDA2	56	N9	K8		To which pin the c input or output car selected from the SDA1, ExSDAA, a ExSDAB pins.
	ExSDAA	54	R8	K6		
	ExSDAB	52	P7	M7		
I/O port	P17 to P10	104 to 110, 112	C15, D12, C14, B15, B14, A15, C13, B12	D10, C12, C13, D11, B13, A12, A13, B12	Input/ Output	8-bit input/output p
	P27 to P20	96 to 103	F14, E13, E15, E14, E12, D15, D14, D13	E10, F13, E12, E13, F11, D12, E11, D13	Input/ Output	8-bit input/output p
	P37 to P30	128 to 121	D7, A8, C8, D8, B9, A9, C9, D9	D6, D7, D8, A8, B7, C8, D9, A9	Input/ Output	8-bit input/output p
	P47 to P40	6 to 2, 138 to 136	C1, D3, C2, B1, C3, D5, B5, A5	C3, C1, B2, C2, A1, B4, A5, D7	Input/ Output	8-bit input/output p (The output type o NMOS push-pull.)
	P52 to P50	14 to 16	F2, G4, G1	E4, F3, G2	Input/ Output	3-bit input/output p (The output type o NMOS push-pull.)
	P67 to P60	85 to 78	J13, J12, K14, K13, K12, L15, L14, L13	H12, J11, J10, K13, J12, K11, K12, L13	Input/ Output	8-bit input/output p

P90		H2, J4, J3, J1, J2, K4	G4, H4, G1, H2, G3, J4	Output	(The output type of NMOS push-pull.)
PA7 to PA0	33 to 35, 37 to 41	N1, M4, N2, R1, N3, R2, P3, R3	L2, K3, L1, N2, M2, M3, N1, N3	Input/ Output	8-bit input/output pi (The output type of PA0 is NMOS push
PB7 to PB0	113 to 120	D11, A12, C11, B11, A11, D10, A10, B10	A11, C11, B10, C10, A10, B9, C9, B8	Input/ Output	8-bit input/output pi
PC7 to PC0	87 to 94	H12, H13, H15, H14, G12, G13, G15, G14	J13, H11, G12, G10, H13, F12, G13, G11	Input/ Output	8-bit input/output pi
PD7 to PD0	59 to 66	M10, N10, R10, P10, N11, R11, P11, M11	L7, K9, N8, M9, L8, K10, N9, M10	Input/ Output	8-bit input/output pi
PE4 to PE0*[1]	28 to 32	L2, L4, M1, M2, M3	K4, J1, K2, J3, K1	Input	5-bit input pins
PF7 to PF0	43 to 50	N5, P5, R5, M6, N6, R6, P6, M7	M4, L4, N4, M5, L5, M6, N5, K5	Input/ Output	8-bit input/output pi
PG7 to PG0	51 to 58	R7, P7, M8, R8, P8, N9, R9, P9	L6, M7, N6, K6, K7, K8, N7, M8	Input/ Output	8-bit input/output pi (The output type of PG0 is NMOS push

B8, C5

Notes: 1. Pins PE4 to PE1 are not supported by the system development tool (emulato

2. Following precautions are required on the power-on reset signal that is applie $\overline{\text{ETRST}}$ pin.
The reset signal should be applied on power supply.
Set apart the power-on reset circuit from this LSI to prevent the $\overline{\text{ETRST}}$ pin of emulator from affecting the operation of this LSI.
Set apart the power-on reset circuit from this LSI to prevent the system reset from affecting the $\overline{\text{ETRST}}$ pin of the emulator.

- Upward-compatible with H8/300 and H8/300H CPUs
  — Can execute H8/300 and H8/300H CPUs object programs
- General-register architecture
  — Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-b
- 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: 2 states

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

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

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

- Register configuration

  The MAC register is supported by the H8S/2600 CPU only.
- Basic instructions

  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported by the H CPU only.
- The number of execution states of the MULXU and MULXS instructions;

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	2*	12
	MULXU.W Rs, ERd	2*	20
MULXS	MULXS.B Rs, Rd	3*	13
	MULXS.W Rs, ERd	3*	21
CLRMAC	CLRMAC	1*	Not supported
LDMAC	LDMAC ERs,MACH	1*	
	LDMAC ERs,MACL	1*	
STMAC	STMAC MACH,ERd	1*	
	STMAC MACl,ERd	1*	

Note: * This becomes one state greater immediately after a MAC instruction.

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

- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyt 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 instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

2.1.3 Differences from H8/300H CPU

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

- More control registers
  - 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 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.

Linear access to a 64-kbyte maximum address space is provided.

- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-b segments of 32-bit registers. When En is used as a 16-bit register it can contain any va when the corresponding general register (Rn) is used as an address register. If the gen register is referenced in the register indirect addressing mode with pre-decrement (@- post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corre 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 tab branch address is stored per 16 bits. The exception vector table structure in normal mo 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 instr uses an 8-bit absolute address included in the instruction code to specify a memory op 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 area from H' H'00FF. Note that the first part of this range 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 condition-code register (CCR), and extended control register (EXR) are pushed onto t in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.

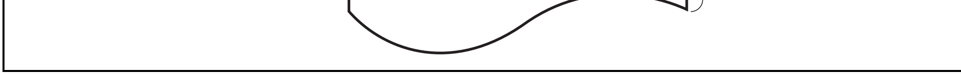
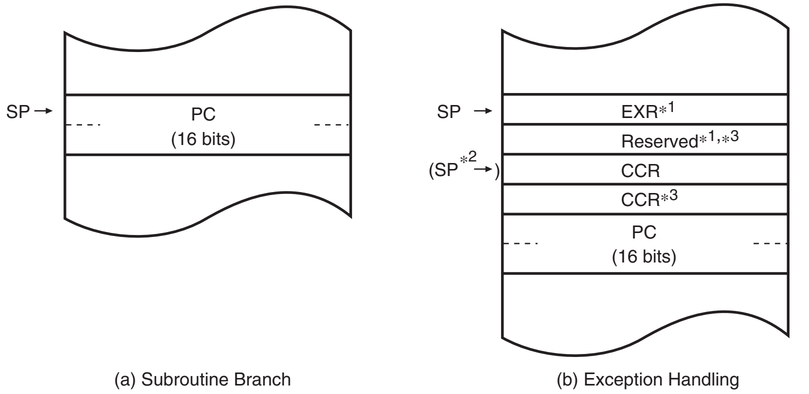


Figure 2.1 Exception Vector Table (Normal Mode)



(a) Subroutine Branch

(b) 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 2.2 Stack Structure in Normal Mode

Exception Vector Table and Memory Indirect Branch Addresses

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

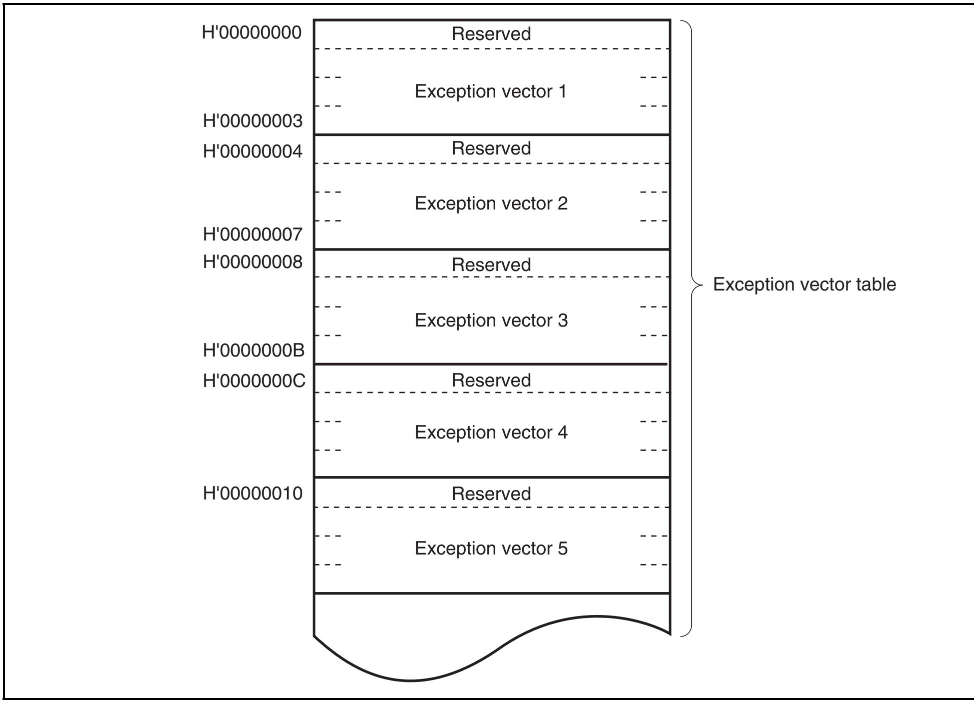
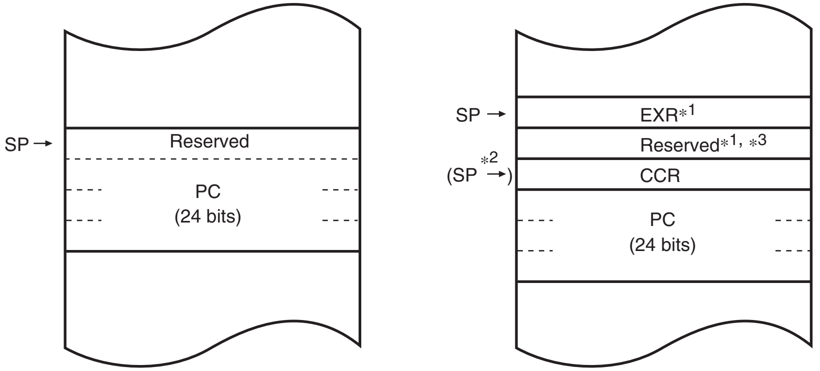


Figure 2.3 Exception Vector Table (Advanced Mode)

EXR is not pushed onto the stack in interrupt control mode 0. For details, see section
Exception Handling.



(a) Subroutine Branch

(b) 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

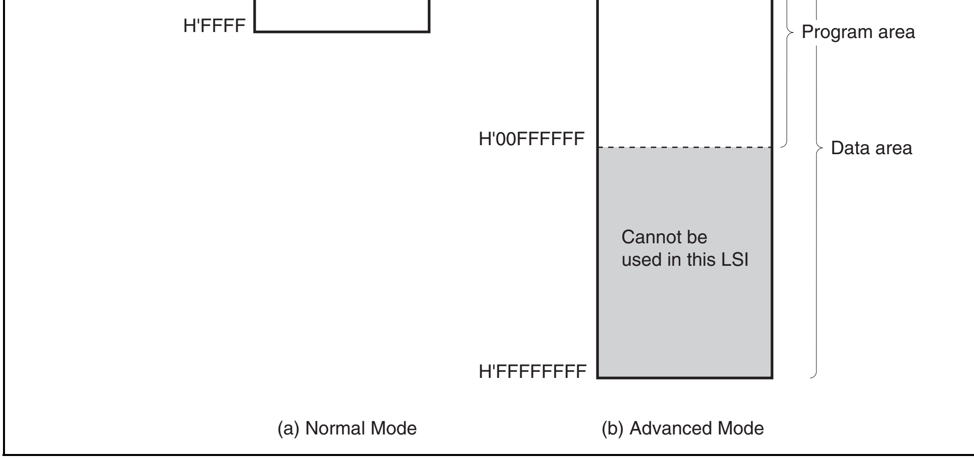
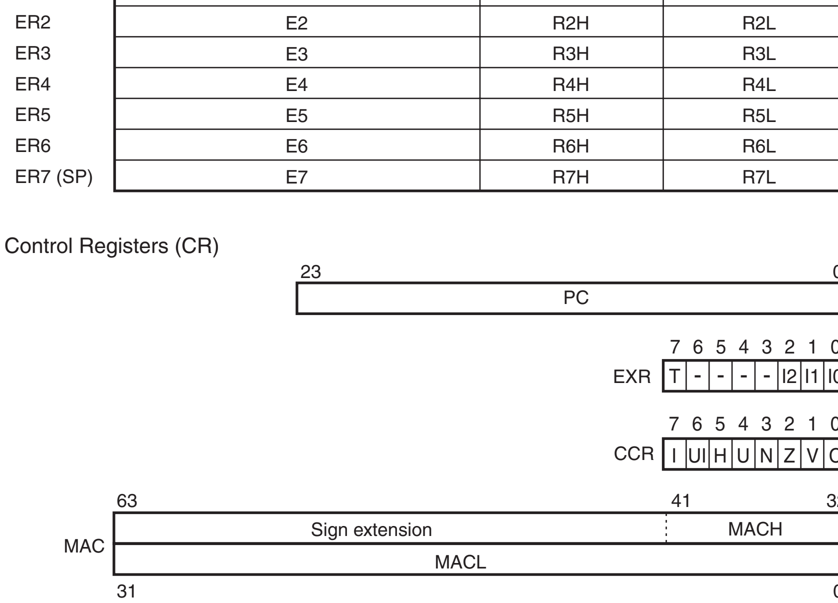


Figure 2.5   Memory Map



[Legend]

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control 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		

Figure 2.6 CPU Registers

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R
RL (R0L to R7L). These registers are functionally equivalent, providing a maximum of s
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-regis
function, and is used implicitly in exception handling and subroutine calls. Figure 2.8 sho
stack.

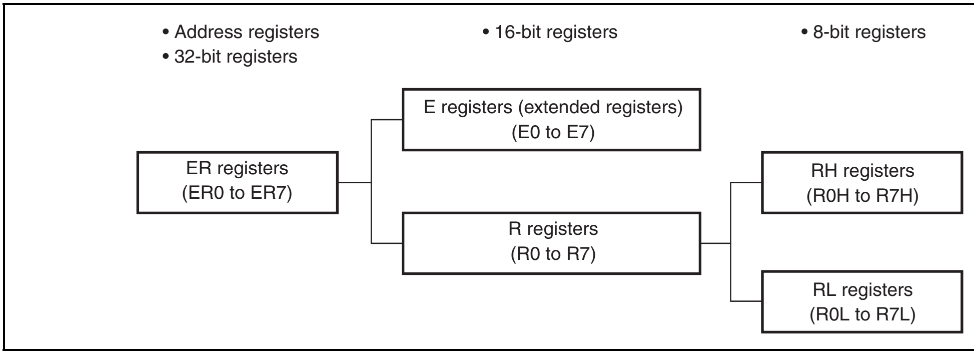


Figure 2.7 Usage of General Registers

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. Th of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (W instruction is fetched, the least significant PC bit is regarded as 0).

2.4.3 Extended Control Register (EXR)

EXR is an 8-bit register that manipulates the LDC, STC, ANDC, ORC, and XORC instr When these instructions, except for the STC instruction, are executed, all interrupts inclu 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 This bit has no effect on the operation of the M
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	Interrupt Request Mask Bits 2 to 0
1	I1	1	R/W	These bits have no effect on the operation of
0	I0	1	R/W	

7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. accepted regardless of the I bit setting. The I b 1 at the start of an exception-handling sequenc details, refer to section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be read or written by software using the LI ANDC, ORC, and XORC instructions.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, C or NEG.B instruction is executed, this flag is se there is a carry or borrow at bit 3, and cleared t otherwise. When the ADD.W, SUB.W, CMP.W NEG.W instruction is executed, the H flag is se there is a carry or borrow at bit 11, and cleared otherwise. When the ADD.L, SUB.L, CMP.L, o instruction is executed, the H flag is set to 1 if t carry or borrow at bit 27, and cleared to 0 othe
4	U	Undefined	R/W	User Bit Can be read or written by software using the LI ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of da sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 indicate non-zero data.

- Shift and rotate instructions, to indicate a

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

2.4.5 Multiply-Accumulate Register (MAC)

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

2.4.6 Initial Values of CPU Registers

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

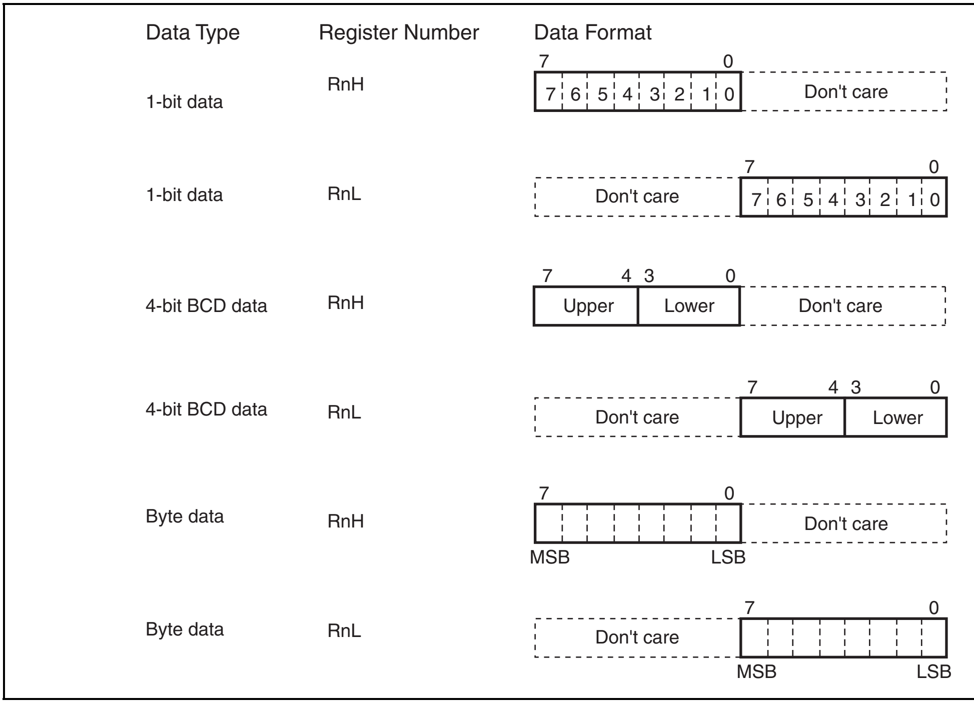
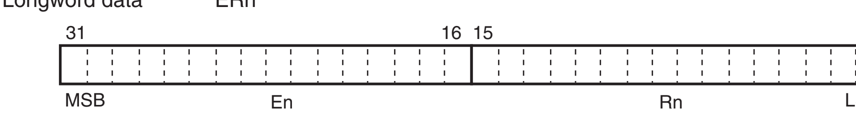


Figure 2.9 General Register Data Formats (1)



[Legend]

ERn:	General register ER
En:	General register E
Rn:	General register R
RnH:	General register RH
RnL:	General register RL
MSB:	Most significant bit
LSB:	Least significant bit

Figure 2.9 General Register Data Formats (2)

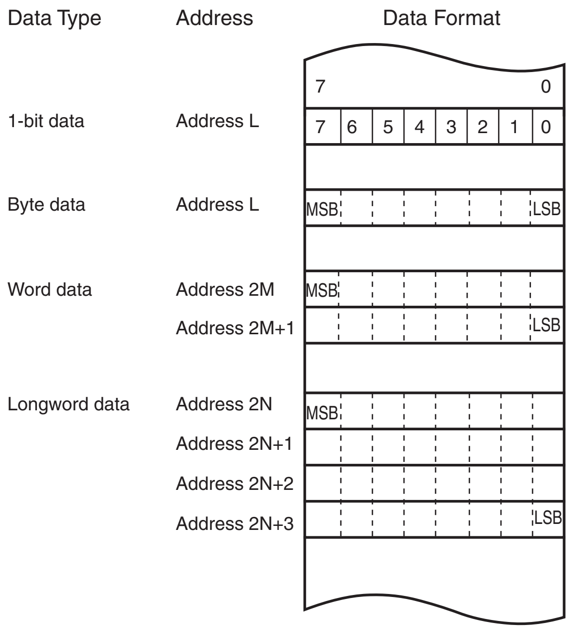


Figure 2.10 Memory Data Formats

Arithmetic operation	ADD, SUB, CMP, NEG	B/W
	ADDX, SUBX, DAA, DAS	B
	INC, DEC	B/W
	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
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—
Block data transfer	EEPMOV	—

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

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+,Rn and MOV.W 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 ins

ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR
→	Move
∼	NOT (logical complement)

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 MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+,
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is ident MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn,
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

Note: * Refers to the operand size.

B: Byte

W: Word

L: Longword

INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte op 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
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general re referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registe 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: 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: eit bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 16-bit quotient and 16-bit remainder.

Note: * Refers to the operand size.

B: Byte

W: Word

L: Longword

NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data i general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by padding with zeros left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by extending the sign
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds t to the multiply-accumulate register. The following operations c 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-accu register.

Note: 1. Refers to the operand size.

B: Byte

W: Word

L: Longword

2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS ins

		another general register or immediate data.
NOT	B/W/L	~(Rd) → (Rd) Takes the one's complement (logical complement) of general re contents.

Note: * Refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.6 Shift Instructions

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

Note: * Refers to the operand size.

B: Byte

W: Word

L: Longword

BNOT	B	~(<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand number is specified by 3-bit immediate data or the lower three general register.
BTST	B	~(<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand a or clears the Z flag accordingly. The bit number is specified by 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 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 gene register or memory operand and stores the result in the carry f 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 m 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 gener or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Note: * Refers to the operand size.

B: Byte

		carry flag.
BILD	B	∼(<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or m 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 regi memory operand.
BIST	B	∼C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in general register or memory operand. The bit number is specified by 3-bit immediate data.

Note: * Refers to the operand size.

B: Byte

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

		Transfers CCR or EXR contents to a general register or memor Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are val
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically XORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: * Refers to the operand size.

B: Byte

W: Word

```
  else next;
```

Transfers a data block. Starting from the address set in ER5, t data for the number of bytes set in R4L or R4 to the address lo in ER6.

Execution of the next instruction begins as soon as the transfe completed.

Some instructions have two operation fields.

- Register Field

  Specifies a general register. Address registers are specified by 3 bits, and data register 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.

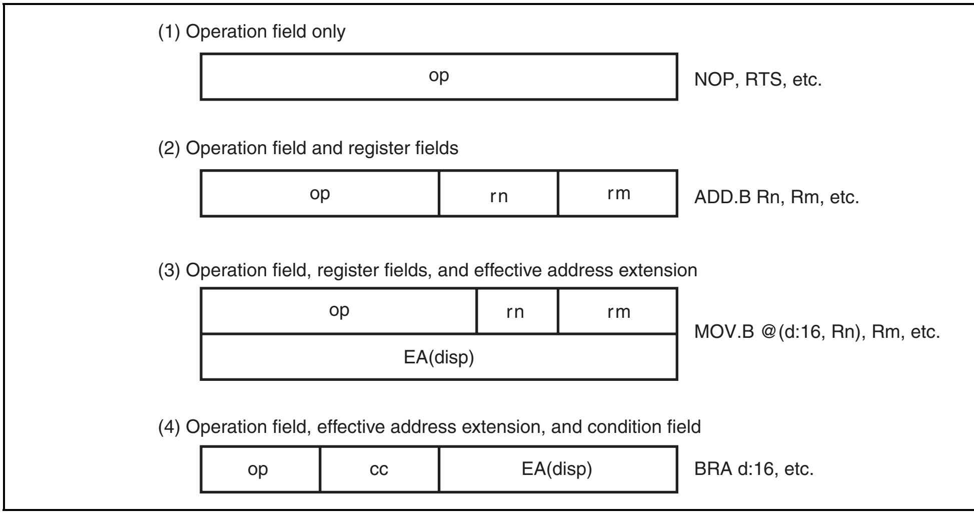

(1) Operation field only
op
NOP, RTS, etc.
(2) Operation field and register fields
op
rn
rm
ADD.B Rn, Rm, etc.
(3) Operation field, register fields, and effective address extension
op
rn
rm
EA(disp)
MOV.B @(d:16, Rn), Rm, etc.
(4) Operation field, effective address extension, and condition field
op
cc
EA(disp)
BRA d:16, etc.

Figure 2.11 Instruction Formats (Examples)

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/@a
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register contai
operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and
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 conta
address of the operand on memory. If the address is a program instruction address, the lo
bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4
longword transfer instruction. For the word or longword transfer instructions, the register
should be even.

Register indirect with pre-decrement—@-ERn: The value 1, 2, or 4 is subtracted from
address register (ERn) specified by the register field in the instruction code, and the result
address of a memory operand. The result is also stored in the address register. The value
subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer
instruction. For the word or longword transfer instructions, the register value should be ev

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 ad
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
For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute add
access the entire address space.

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

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

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

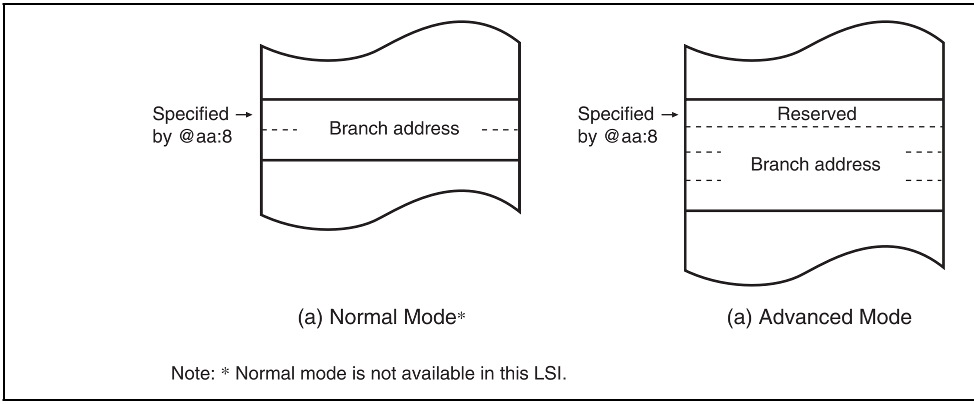
The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some
manipulation instructions contain 3-bit immediate data in the instruction code, specifyin
number. The TRAPA instruction contains 2-bit immediate data in its instruction code, s
vector address.

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 con
the instruction is sign-extended and added to the 24-bit PC contents to generate a branch
Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed t
(H′00). The PC value to which the displacement is added is the address of the first byte
instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or
+32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting valu
be an even number.

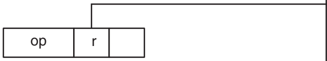
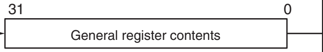
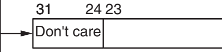
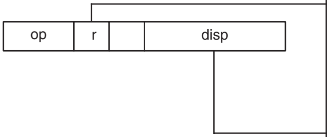
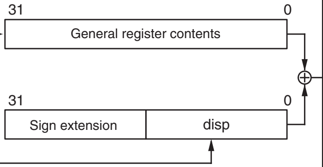
If an odd address is specified in word or longword memory access, or as a branch address least significant bit is regarded as 0, causing data to be accessed or instruction code to be at the address preceding the specified address. (For further information, see section 2.5.2, Data Formats.)

Note: Normal mode is not available in this LSI.

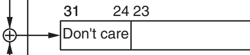
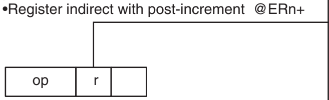
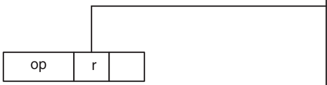
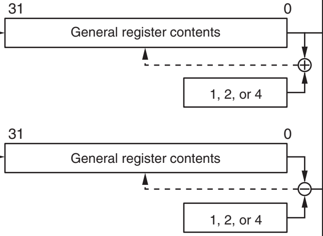


Note: * Normal mode is not available in this LSI.

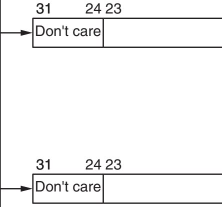
Figure 2.12 Branch Address Specification in Memory Indirect Mode

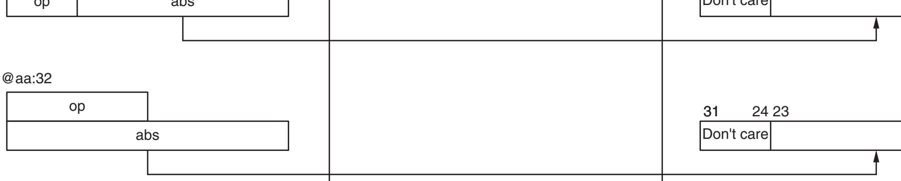
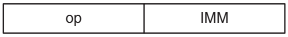
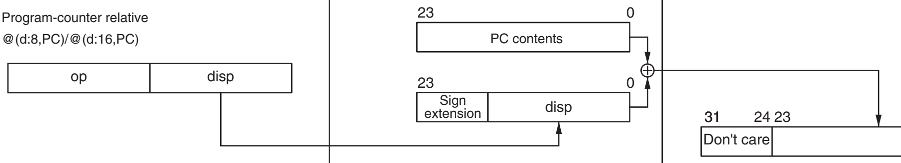
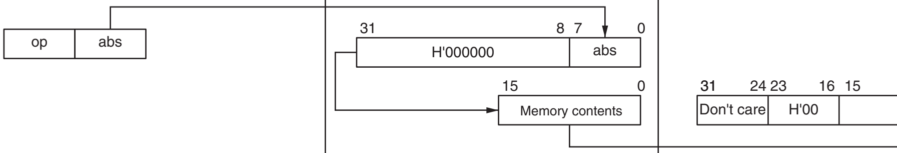
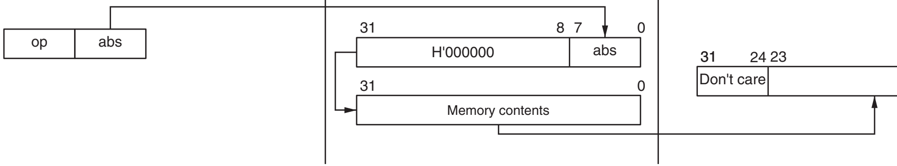
<table>
<tr><td></td><td> </td><td></td><td></td></tr>
<tr><td>2</td><td>Register indirect(@ERn)
 </td><td> </td><td> </td></tr>
<tr><td>3</td><td>Register indirect with displacement
@(d:16,ERn) or @(d:32,ERn)
 </td><td>

31
0
General register contents
31
0
Sign extension
disp

</td><td> </td></tr>
<tr><td>4</td><td>Register indirect with post-increment or pre-decrement
•Register indirect with post-increment @ERn+

•Register indirect with pre-decrement @-ERn
 </td><td>

31
0
General register contents
1, 2, or 4
31
0
General register contents
1, 2, or 4

<table><tr><td>Operand Size</td><td>Offset</td></tr><tr><td>Byte</td><td>1</td></tr><tr><td>Word</td><td>2</td></tr><tr><td>Longword</td><td>4</td></tr></table></td><td> </td></tr>
</table>

	 @aa:32 		
6	Immediate #xx:8/#xx:16/#xx:32 		Operand is immediate data.
7	Program-counter relative @(d:8,PC)/@(d:16,PC)  op disp 23 0 PC contents 23 0 Sign extension disp 31 24 23 Don't care		
8	Memory indirect @@aa:8 • Normal mode*  op abs 31 8 7 0 H'000000 abs 15 0 Memory contents 31 24 23 16 15 Don't care H'00 • Advanced mode  op abs 31 8 7 0 H'000000 abs 31 0 Memory contents 31 24 23 Don't care		

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

- Exception-Handling State

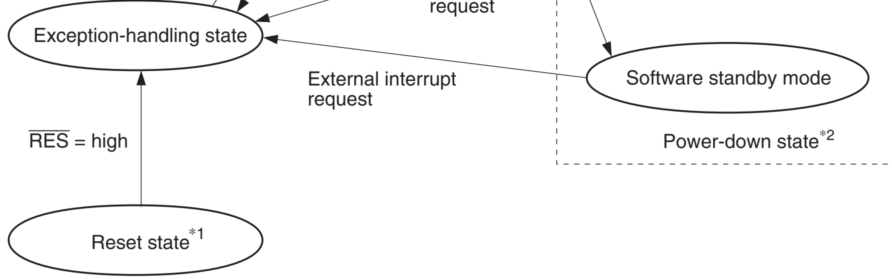
  The exception-handling state is a transient state that occurs when the CPU alters the processing flow due to an exception source, such as a reset, trace, interrupt, or trap in The CPU fetches a start address (vector) from the exception vector table and branche address. For further details, refer to section 4, Exception Handling.

- Program Execution State

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

- Program Stop State

  This is a power-down state in which the CPU stops operating. The program stop stat when a SLEEP instruction is executed or the CPU enters software standby mode. Fo details, refer to section 26, Power-Down Modes.



Notes: 1. From any state, a transition to the reset state is made whenever the $\overline{\text{RES}}$ pin goes low. A transition can also be made to the reset state when the watchdog timer overflows.

2. The power-down state also includes watch mode. For details, refer to section 26, Power-Down

Figure 2.13 State Transitions

beforehand.

MCU Operating Mode	MD2	MD1	MD0*	CPU Operating Mode	Description	On-C
2	0	1	0	Advanced	Single-chip mode	Enab
4	1	0	0	—	Flash memory programming/erasing	—
6	1	1	0	Emulation	On-chip emulation mode	Enab

Note: * MD0 is not available as a pin and is internally fixed to 0.

Modes 2 is single-chip mode.

Modes 0, 1, 3, 5 and 7 are not available in this LSI. Modes 4 and 6 are operating modes special purpose. Thus, mode pins should be set to enable mode 2 in the normal program state. Mode pin settings should not be changed during operation. After a reset is cancele mode pin inputs should be latched by reading MDCR.

Mode 4 is a boot mode for programming or erasing the flash memory. For details, see se Flash Memory.

Mode 6 is an on-chip emulation mode. In this mode, this LSI is controlled by an on-chip (E10A) via the JTAG, thus enabling on-chip emulation.

System control register 3	SYSCR3	R/W	H'60	H'FE7D	8

3.2.1 Mode Control Register (MDCR)

MDCR is used to set an operating mode and to monitor the current operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	EXPE	0	R/W	Reserved The initial value should not be changed.
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	MDS2	—*	R	Mode Select 2 and 1
1	MDS1	—*	R	These bits indicate the input levels at mode pir and MD1) (the current operating mode). The M MDS1 bits correspond to the MD2 and MD1 pi respectively. These bits are read-only bits and be written to. The input levels of the mode pins (MD2 and M latched into these bits when MDCR is read. Th latches are canceled by a reset.
0	—	0	R	Reserved The initial value should not be changed.

Note: * The initial values are determined by the settings of the MD2 and MD1 pins.

5	INTM1	0	R	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select the interrupt control mode o interrupt controller. For details on the interrupt control modes, se 5.6, Interrupt Control Modes and Interrupt Op 00: Interrupt control mode 0 01: Interrupt control mode 1 10: Setting prohibited 11: Setting prohibited
3	XRST	1	R	External Reset Indicates the reset source. A reset is caused external reset input, or when the watchdog tin overflows. 0: A reset is caused when the watchdog timer 1: A reset is caused by an external reset
2	NMIEG	0	R/W	NMI Edge Select Selects the valid edge of the NMI interrupt inp 0: An interrupt is requested at the falling edge input 1: An interrupt is requested at the rising edge input

				0: Enables CPU access for registers of TMR_X TMR_Y in areas from H'(FF)FFF0 to H'(FF)F from H'(FF)FFFC to H'(FF)FFFF 1: Enables CPU access for the keyboard matri interrupt registers and input pull-up MOS co register in areas from H'(FF)FFF0 to H'(FF)F from H'(FF)FFFC to H'(FF)FFFF When the RELOCATE bit is set to 1, this bit is For details, see section 3.2.4, System Control 3 (SYSCR3) and section 27, List of Registers.
0	RAME	1	R/W	RAM Enable Enables or disables on-chip RAM. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

				details on the transfer rate, see table 18.4.
4	IICE	0	R/W	I²C Master Enable When the RELOCATE bit is cleared to 0, ena disables CPU access for IIC registers (ICCR, ICDR/SARX, ICMR/SAR, and ICRES), PWM (DADRAH/DACR, DADRAL, DADRBH/DACN DADRBL/DACNTL), and SCI registers (SMR, SCMR). 0: SCI_1 registers are accessed in areas fro H'(FF)FF88 to H'(FF)FF89 and from H'(FF H'(FF)FF8F. SCI_2 registers are accessed in areas fro H'(FF)FFA0 to H'(FF)FFA1 and from H'(F H'(FF)FFA7. Access is prohibited in areas from H'(FF)F H'(FF)FFD9 and from H'(FF)FFDE to H'(F 1: IIC_1 registers are accessed in areas fron H'(FF)FF88 to H'(FF)FF89 and from H'(FF H'(FF)FF8F. PWMX registers are accessed in areas fro H'(FF)FFA0 to H'(FF)FFA1 and from H'(F H'(FF)FFA7. IIC_0 registers are accessed in areas fron H'(FF)FFD8 to H'(FF)FFD9 and from H'(F H'(FF)FFDF. ICRES is accessed in areas of H'(FF)FEE When the RELOCATE bit is set to 1, this bit is For details, see section 3.2.4, System Contro 3 (SYSCR3) and section 27, List of Registers

				H'(FF)FEA8 to H'(FF)FEAE is reserved. When RELOCATE is 1, control registers of p down state and peripheral modules are acce an area from H'(FF)FF80 to H'(FF)FF87. Are H'(FF)FEA8 to H'(FF)FEAE is reserved. 1: When RELOCATE is 0, control registers of f memory are accessed in an area from H'(FF H'(FF)FEAE. Area from H'(FF)FF80 to H'(FF reserved. When RELOCATE is 1, control registers of p down state and peripheral modules are acce an area from H'(FF)FF80 to H'(FF)FF87. Co registers of flash memory are accessed in ar from H'(FF)FEA8 to H'(FF)FEAE.
2	IICS	0	R/(W)	I^{2}C Extra Buffer Select Specifies bits 7 to 4 of port A as output buffers SLC and SDA. These pins are used to impleme interface only by software. 0: PA7 to PA4 are normal input/output pins. 1: PA7 to PA4 are input/output pins enabling b driving.
1	ICKS1	0	R/W	Internal Clock Source Select 1 and 0
0	ICKS0	0	R/W	These bits select a clock to be input to the time (TCNT) and a count condition together with bits to CKS0 in the timer control register (TCR). Fo see section 13.3.4, Timer Control Register (TC

				0: H8S/2140B Group compatible vector mode 1: Extended vector mode For details, see section 5, Interrupt Controller
5	RELOCATE	1	R/W	Register Address Map Select Selects compatible mode or extended mode f register map. When extended mode is selected for the regis CPU access for registers can be controlled w using the KINWUE bit in SYSCR or the IICE b STCR to switch the registers to be accessed. 0: H8S/2140B Group compatible register map 1: Extended register map mode For details, see section 27, List of Registers.
4 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

Note: * Switch the modes when an interrupt occurrence is disabled.

3.3 Operating Mode Descriptions

3.3.1 Mode 2

The CPU can access a 16-Mbyte address space in either advanced mode or single-chip n on-chip ROM is enabled.

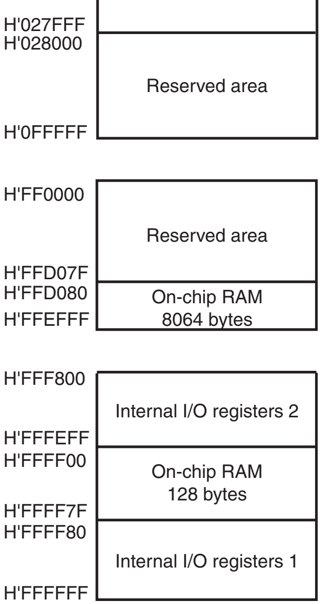


Figure 3.1 Address Map

Priority	Exception Type	Start of Exception Handling
High ↑	Reset	Starts immediately after a low-to-high transition of pin, or when the watchdog timer overflows.
	Illegal instruction	Exception handling starts when an undefined cod executed.
	Interrupt	Starts when execution of the current instruction or handling ends, if an interrupt request has been iss Interrupt detection is not performed on completion ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
	Direct transition	Starts when a direct transition occurs as the resul SLEEP instruction execution.
Low	Trap instruction	Started by execution of a trap (TRAPA) instruction instruction exception handling requests are accep times in the program execution state.

Exception Source		Number	Advanced Mode
Reset		0	H'000000 to H'000003
Reserved for system use		1 │ 3	H'000004 to H'000007 │ H'00000C to H'00000F
Illegal instruction		4	H'000010 to H'000013
Reserved for system use		5	H'000014 to H'000017
Direct transition		6	H'000018 to H'00001B
External interrupt (NMI)		7	H'00001C to H'00001F
Trap instruction (four sources)		8	H'000020 to H'000023
		9	H'000024 to H'000027
		10	H'000028 to H'00002B
		11	H'00002C to H'00002F
Reserved for system use		12 │ 15	H'000030 to H'000033 │ H'00003C to H'00003F
External interrupt	IRQ0	16	H'000040 to H'000043
	IRQ1	17	H'000044 to H'000047
	IRQ2	18	H'000048 to H'00004B
	IRQ3	19	H'00004C to H'00004F
	IRQ4	20	H'000050 to H'000053
	IRQ5	21	H'000054 to H'000057
	IRQ6, KIN7 to KIN0	22	H'000058 to H'00005B
	IRQ7, KIN15 to KIN8	23	H'00005C to H'00005F

		│	│
		55	H'0000DC to H'0000DI
External interrupt	IRQ8	56	H'0000E0 to H'0000E3
	IRQ9	57	H'0000E4 to H'0000E7
	IRQ10	58	H'0000E8 to H'0000EE
	IRQ11	59	H'0000EC to H'0000EI
	IRQ12	60	H'0000F0 to H'0000F3
	IRQ13	61	H'0000F4 to H'0000F7
	IRQ14	62	H'0000F8 to H'0000FB
	IRQ15	63	H'0000FC to H'0000FF
Internal interrupt*		64	H'000100 to H'000103
		│	│
		127	H'0001FC to H'0001FF

Note: * For details on the internal interrupt vector table, see section 5.5, Interrupt Exce Handling Vector Tables.

Direct transition		6	H'000018 to H'00001B
External interrupt (NMI)		7	H'00001C to H'00001F
Trap instruction (four sources)		8	H'000020 to H'000023
		9	H'000024 to H'000027
		10	H'000028 to H'00002B
		11	H'00002C to H'00002F
Reserved for system use		12	H'000030 to H'000033
		|	|
		15	H'00003C to H'00003F
External interrupt	IRQ0	16	H'000040 to H'000043
	IRQ1	17	H'000044 to H'000047
	IRQ2	18	H'000048 to H'00004B
	IRQ3	19	H'00004C to H'00004F
	IRQ4	20	H'000050 to H'000053
	IRQ5	21	H'000054 to H'000057
	IRQ6	22	H'000058 to H'00005B
	IRQ7	23	H'00005C to H'00005F
Internal interrupt*		24	H'000060 to H'000063
		|	|
		29	H'000074 to H'000077
External interrupt	KIN7 to KIN0	30	H'000078 to H'00007B
External interrupt	KIN15 to KIN8	31	H'00007C to H'00007F
Reserved for system use		32	H'000080 to H'000083
External interrupt	WUE15 to WUE8	33	H'000084 to H'000087

	IRQ13	61	H'0000F4 to H'0000F7
	IRQ14	62	H'0000F8 to H'0000FB
	IRQ15	63	H'0000FC to H'0000FF
Internal interrupt*		64 │ 127	H'000100 to H'000103 │ H'0001FC to H'0001FF

Note: * For details on the internal interrupt vector table, see section 5.5, Interrupt Exce
Handling Vector Tables.

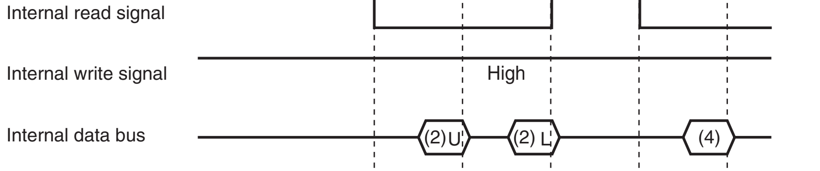
4.3 Reset

A reset has the highest exception priority. When the $\overline{\text{RES}}$ pin goes low, all processing ha
this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for a
ms at power-on. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20
reset initializes the internal state of the CPU and the registers of on-chip peripheral mod
chip can also be reset by overflow of the watchdog timer. For details, see section 14, Wa
Timer (WDT).

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
exception handling as follows:

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



(1) Reset exception handling vector address (1) U = H'000000 (1) L = H'000002
(2) Start address (contents of reset exception handling vector address)
(3) Start address ((3) = (2)U + (2)L)
(4) First program instruction

Figure 4.1 Reset Sequence (Mode 2)

4.3.2 Interrupts Immediately after Reset

If an interrupt is accepted immediately after a reset and before the stack pointer (SP) is in the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, interrupt requests, including NMI, are disabled immediately after a reset. Since the first in of a program is always executed immediately after a reset, make sure that this instruction initializes the SP (example: MOV.L #xx: 32, SP).

4.3.3 On-Chip Peripheral Modules after Reset is Cancelled

After a reset is cancelled, the module stop control registers (MSTPCRH, MSTPCRL, MS MSTPCRB) are initialized, and all modules except the DTC operate in module stop mode Therefore, the registers of on-chip peripheral modules cannot be read from or written to. from and write to these registers, clear module stop mode. For details on module stop mo section 26, Power-Down Modes.

2. A vector address corresponding to the interrupt source is generated, the start address from the vector table to the PC, and program execution starts from that address.

4.5 Trap Instruction Exception Handling

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

Trap instruction exception handling is conducted as follows:

1. The values in the program counter (PC) and condition code register (CCR) are saved stack.
2. A vector address corresponding to the interrupt source is generated, the start address 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 corresponding to a ve number from 0 to 3, as specified in the instruction code.

Table 4.4 shows the status of CCR after execution of trap instruction exception handling

Table 4.4 Status of CCR after Trap Instruction Exception Handling

	CCR	
Interrupt Control Mode	**I**	**UI**
0	Set to 1	Retains value prior to ex
1	Set to 1	Set to 1

the PC, and program execution starts from that address.

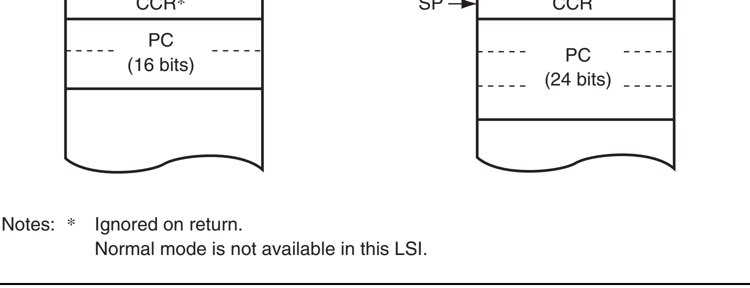
Table 4.5 shows the state of CCR after execution of illegal instruction exception handling

Table 4.5 Status of CCR after Illegal Instruction Exception Handling

	CCR	
Interrupt Control Mode	**I**	**UI**
0	Set to 1	Retains the previous valu
1	Set to 1	Set to 1

Illegal instruction code is not detected for fields that do not affect the definition of the ins such as an effective address extension (EA) and register fields. In addition, the instructio instructions consisting of multiple words are detected individually and are not detected as combinations of instruction codes.

Do not execute instruction codes that are not defined. The contents of general registers ar guaranteed after the execution of an undefined instruction code or exception handling by illegal instruction. The value of the stack pointer at the time of exception handling by the instruction and the saved contents of the PC are also not guaranteed.



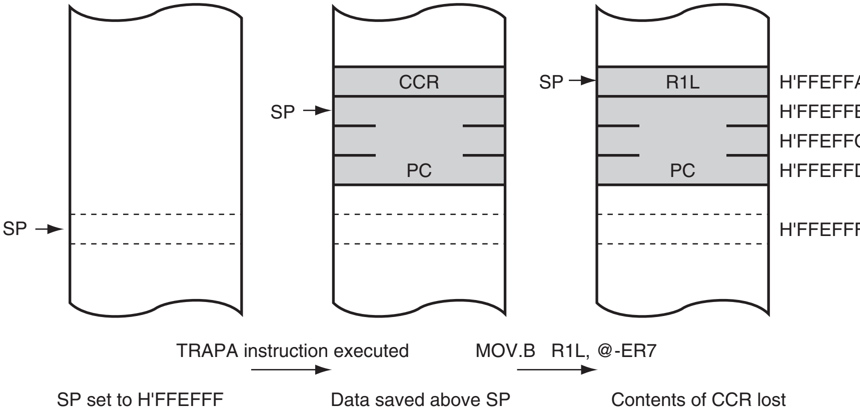
Notes: * Ignored on return.
Normal mode is not available in this LSI.

Figure 4.2 Stack Status after Exception Handling

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 4.3 shows an example of wh
occurs when the SP value is odd.



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

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

Figure 4.3 Operation when SP Value Is Odd

levels for all interrupt requests excluding NMI and address breaks.

- Three-level interrupt mask control

  By means of the interrupt control mode, I and UI bits in CCR and ICR, 3-level interr control is performed.

- Forty-one external interrupt pins

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or fall detection can be selected for NMI. Falling-edge, rising-edge, or both-edge detection, sensing, can be independently selected for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$. When the EIVS bit in the control register 3 (SYSCR3) is cleared to 0, the IRQ6 interrupt is generated by $\overline{\text{IRQ6}}$ to $\overline{\text{KIN0}}$. The IRQ7 interrupt is generated by $\overline{\text{IRQ7}}$ or $\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$. When the EIV the system control register 3 (SYSCR3) is set to 1, interrupts are requested on the fal of $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$. For $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$, either rising-edge or falling-edge detectio selected individually for each pin regardless of the EIVS bit setting.

- Two interrupt vector addresses are selectable

  H8S/2140B Group compatible interrupt vector addresses or extended interrupt vecto addresses are selected depending on the EIVS bit in system control register 3 (SYSC extended mode, independent vector addresses are assigned for the interrupt vector ad KIN7 to KIN0 or KIN15 to KIN8 interrupts.

- General ports for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ6}}$ input are selectable

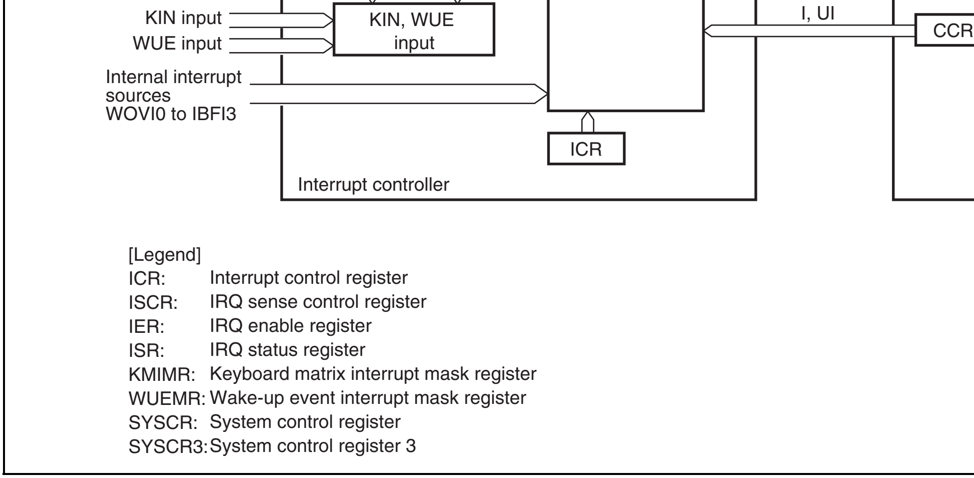

KIN input
WUE input
KIN, WUE input
I, UI
CCR
Internal interrupt sources WOVI0 to IBFI3
ICR
Interrupt controller
[Legend]
ICR: Interrupt control register
ISCR: IRQ sense control register
IER: IRQ enable register
ISR: IRQ status register
KMIMR: Keyboard matrix interrupt mask register
WUEMR: Wake-up event interrupt mask register
SYSCR: System control register
SYSCR3: System control register 3

Figure 5.1 Block Diagram of Interrupt Controller

		sensing, can be selected individually for each pin. T pin the IRQ15 to IRQ6 interrupt is input can be sele the $\overline{\text{IRQm}}$ and $\overline{\text{ExIRQm}}$ pins. (n = 15 to 6)
$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$	Input	Maskable external interrupt pins When EIVS = 0, falling-edge or level-sensing can b selected. When EIVS = 1, an interrupt is requested at the fall
$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	Input	Maskable external interrupt pins Either rising edge or falling edge detection can be s for each pin.

Interrupt control registers B	ICRB	R/W	H'00	H'FEE9	8
Interrupt control registers C	ICRC	R/W	H'00	H'FEEA	8
Interrupt control registers D	ICRD	R/W	H'00	H'FE87	8
Address break control register	ABRKCR	R/W	—	H'FEF4	8
Break address registers A	BARA	R/W	H'00	H'FEF5	8
Break address registers B	BARB	R/W	H'00	H'FEF6	8
Break address registers C	BARC	R/W	H'00	H'FEF7	8
IRQ sense control register 16H	ISCR16H	R/W	H'00	H'FEFA	8
IRQ sense control register 16L	ISCR16L	R/W	H'00	H'FEFB	8
IRQ sense control register H	ISCRH	R/W	H'00	H'FEEC	8
IRQ sense control register L	ISCRL	R/W	H'00	H'FEED	8
IRQ enable register 16	IER16	R/W	H'00	H'FEF8	8
IRQ enable register	IER	R/W	H'00	H'FFC2	8
IRQ status register 16	ISR16	R/W	H'00	H'FEF9	8
IRQ status register	ISR	R/W	H'00	H'FEEB	8
Keyboard matrix interrupt mask register A	KMIMRA	R/W	H'FF	H'FFF3 H'FE83*[1]	8
Keyboard matrix interrupt mask register	KMIMR	R/W	H'BF H'FF*[2]	H'FFF1 H'FE81*[1]	8
Wake-up event interrupt mask registers	WUEMR	R/W	H'00	H'FE45	8
IRQ sense port select register 16	ISSR16	R/W	H'00	H'FEFC	8
IRQ sense port select register	ISSR	R/W	H'00	H'FEFD	8

5.3.1 Interrupt Control Registers A to D (ICRA to ICRD)

The ICR registers set interrupt control levels for interrupts other than NMI. The correspe
between interrupt sources and ICRA to ICRD settings is shown in tables 5.2 and 5.3.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ICRn7 to ICRn0	All 0	R/W	Interrupt Control Level 0: Corresponding interrupt source is inte control level 0 (no priority) 1: Corresponding interrupt source is inte control level 1 (priority)

Note: n: A to D

4	ICRn4	IRQ4, IRQ5	CIR	IIC_0	WUE8 to
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1, IIC_2	TPU_0
2	ICRn2	—	TMR_1	FSI	TPU_1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC, FSI	TPU_2
0	ICRn0	WDT_1	PS2	—	—

Note: n: A to D

—: Reserved. The initial value should not be changed.

Table 5.4 Correspondence between Interrupt Source and ICR (Extended Vector Mode: EIVS = 1)

		Register			
Bit	**Bit Name**	**ICRA**	**ICRB**	**ICRC**	**ICRD**
7	ICRn7	IRQ0	A/D converter	SCIF	IRQ8 to I
6	ICRn6	IRQ1	TCM_0, TCM_1, TCM_2, TCM_3	SCI_1	IRQ12 to
5	ICRn5	IRQ2, IRQ3	TDP_0, TDP_1, TDP_2	SCI_2	KIN0 to K
4	ICRn4	IRQ4, IRQ5	CIR	IIC_0	WUE8 to
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1, IIC_2	TPU_0
2	ICRn2	—	TMR_1	FSI	TPU_1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC, FSI	TPU_2
0	ICRn0	WDT_1	PS2	—	—

Note: n: A to D

—: Reserved. The initial value should not be changed.

				When an exception handling is executed for ar break interrupt. [Setting condition] When an address specified by BARA to BARC prefetched while the BIE bit is set to 1.
6 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be
0	BIE	0	R/W	Break Interrupt Enable Enables or disables address break. 0: Disabled 1: Enabled

- BARB

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A15 to A8	All 0	R/W	Addresses 15 to 8 The A15 to A8 bits are compared with A15 to internal address bus.

- BARC

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	A7 to A1	All 0	R/W	Addresses 7 to 1 The A7 to A1 bits are compared with A7 to A1 internal address bus.
0	—	0	R	Reserved This bit is always read as 0 and cannot be mo

4	IRQ14SCA	0	R/W	00: Interrupt request generated at low leve or $\overline{\text{ExIRQn}}$ input 01: Interrupt request generated at falling e $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 10: Interrupt request generated at rising e $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 11: Interrupt request generated at both fal rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input (n = 15 to 12) Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selecte sense port select register 16 (ISSF
3	IRQ13SCB	0	R/W	
2	IRQ13SCA	0	R/W	
1	IRQ12SCB	0	R/W	
0	IRQ12SCA	0	R/W	

0	IRQ8SCA	0	R/W	$\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 11: Interrupt request generated at both falli rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input (n = 11 to 8) Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selectec sense port select register 16 (ISSR1

- ISCRH

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7SCB	0	R/W	IRQn Sense Control B
6	IRQ7SCA	0	R/W	IRQn Sense Control A
5	IRQ6SCB	0	R/W	BA
4	IRQ6SCA	0	R/W	00: Interrupt request generated at low level or $\overline{\text{ExIRQn}}$ input 01: Interrupt request generated at falling ec $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 10: Interrupt request generated at rising ed $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 11: Interrupt request generated at both falli rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input (n = 7 to 4) Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selectec IRQ sense port select register (ISSI ExIRQ5 and ExIRQ4 pins are not su
3	IRQ5SCB	0	R/W	
2	IRQ5SCA	0	R/W	
1	IRQ4SCB	0	R/W	
0	IRQ4SCA	0	R/W	

0	IRQ0SCA	0	R/W	10: Interrupt request generated at rising e $\overline{\text{IRQn}}$ input 11: Interrupt request generated at both fal rising edges of $\overline{\text{IRQn}}$ input (n = 3 to 0)

4	IRQ12E	0	R/W	(n = 15 to 8)
3	IRQ11E	0	R/W	
2	IRQ10E	0	R/W	
1	IRQ9E	0	R/W	
0	IRQ8E	0	R/W	

- IER

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQn Enable
6	IRQ6E	0	R/W	The IRQn interrupt request is enabled wher is 1.
5	IRQ5E	0	R/W	
4	IRQ4E	0	R/W	(n = 7 to 0)
3	IRQ3E	0	R/W	
2	IRQ2E	0	R/W	
1	IRQ1E	0	R/W	
0	IRQ0E	0	R/W	

4	IRQ12F	0	R/(W)*	[Clearing conditions] • When writing 0 to IRQnF flag after rea IRQnF = 1 • When interrupt exception handling is e when low-level detection is set and $\overline{\text{IR}}$ $\overline{\text{ExIRQn}}$ input is high • When IRQn interrupt exception handlin executed when falling-edge, rising-edg both-edge detection is set (n = 15 to 8) Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selecte IRQ sense port select register 16 (
3	IRQ11F	0	R/(W)*	
2	IRQ10F	0	R/(W)*	
1	IRQ9F	0	R/(W)*	
0	IRQ8F	0	R/(W)*	

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

0	IRQ0F	0	R/(W)*	when low-level detection is set and IRQ $\overline{\text{ExIRQn}}$ input is high • When IRQn interrupt exception handling executed when falling-edge, rising-edge both-edge detection is set (n = 7 to 0) Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected IRQ sense port select register (ISSR ExIRQ5 to ExIRQ0 pins are not sup

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

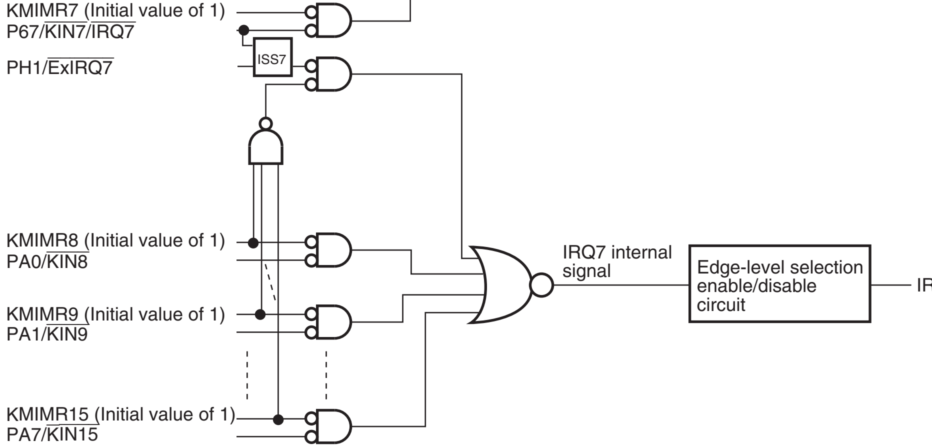
6	KMIMR14	1	R/W	These bits enable or disable a key-sensin
5	KMIMR13	1	R/W	interrupt request (KIN15 to KIN8).
4	KMIMR12	1	R/W	0: Enables a key-sensing input interrupt re
3	KMIMR11	1	R/W	1: Disables a key-sensing input interrupt r
2	KMIMR10	1	R/W	
1	KMIMR9	1	R/W	
0	KMIMR8	1	R/W	

- KMIMR

Bit	Bit Name	Initial Value	R/W	Description
7	KMIMR7	1	R/W	Keyboard Matrix Interrupt Mask
6	KMIMR6	0/1*	R/W	These bits enable or disable a key-sensin interrupt request (KIN7 to KIN0).
5	KMIMR5	1	R/W	
4	KMIMR4	1	R/W	0: Enables a key-sensing input interrupt re
3	KMIMR3	1	R/W	1: Disables a key-sensing input interrupt r
2	KMIMR2	1	R/W	When the EIVS bit in SYSCR3 is cleared t KMIMR6 bit also simultaneously controls e and disabling of the IRQ6 interrupt reques case, the initial value of the KMIMR6 bit is the EIVS bit is set to 1, the initial value of t KMIMR6 bit becomes 1.
1	KMIMR1	1	R/W	
0	KMIMR0	1	R/W	

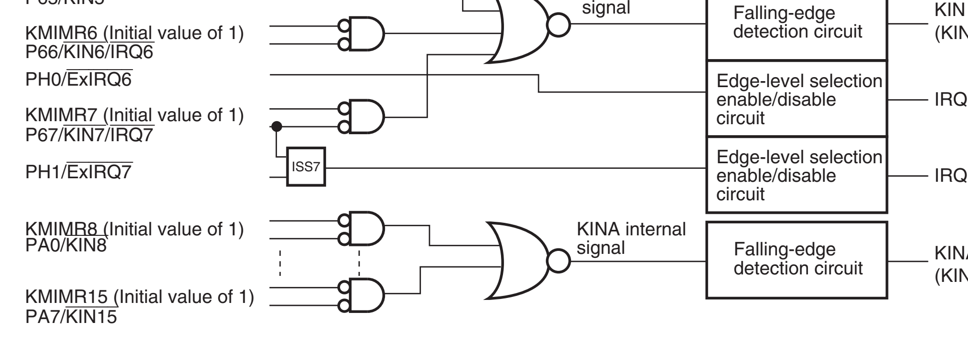
Note: * The initial value is 0 when EIVS = 0 and the initial value is 1 when EIVS = 1.

0	WUEMR8	1	R/W


KMIMR7 (Initial value of 1)
P67/KIN7/IRQ7
PH1/ExIRQ7
ISS7
KMIMR8 (Initial value of 1)
PA0/KIN8
KMIMR9 (Initial value of 1)
PA1/KIN9
KMIMR15 (Initial value of 1)
PA7/KIN15
IRQ7 internal signal
Edge-level selection enable/disable circuit
IR

Note: The ISS7 bit is an external interrupt pin switch bit. For details, see section 5.3.8, IRQ Sense Port Select Register 16 (ISSR16) IRQ Sense Port Select Register (ISSR).

Figure 5.2 Relation between IRQ7/IRQ6 Interrupts and KIN15 to KIN0 Inter KMIMR, and KMIMRA (H8S/2140B Group Compatible Vector Mode: EIVS = 0)


P65/KIN5
signal
Falling-edge detection circuit
KIN
(KIN
KMIMR6 (Initial value of 1)
P66/KIN6/IRQ6
PH0/ExIRQ6
Edge-level selection enable/disable circuit
IRQ
KMIMR7 (Initial value of 1)
P67/KIN7/IRQ7
PH1/ExIRQ7
ISS7
Edge-level selection enable/disable circuit
IRQ
KMIMR8 (Initial value of 1)
PA0/KIN8
KINA internal signal
Falling-edge detection circuit
KINA
(KIN
KMIMR15 (Initial value of 1)
PA7/KIN15

Note: The ISS7 bit is an external interrupt pin switch bit. For details, see section 5.3.8, IRQ Sense Port Select Register 16 (ISSR16) IRQ Sense Port Select Register (ISSR).

Figure 5.3 Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Inter
KMIMR, and KMIMRA
(Extended Vector Mode: EIVS = 1)

In extended vector mode, the initial value of the KMIMR6 bit is 1. Accordingly, it does n of disable the $\overline{\text{IRQ6}}$ pin interrupt. The interrupt input from the $\overline{\text{ExIRQ6}}$ pin becomes the I interrupt request.

				1: PG7/$\overline{\text{ExIRQ15}}$ is selected
6	ISS14	0	R/W	0: P95/$\overline{\text{IRQ14}}$ is selected 1: PG6/$\overline{\text{ExIRQ14}}$ is selected
5	ISS13	0	R/W	0: P94/$\overline{\text{IRQ13}}$ is selected 1: PG5/$\overline{\text{ExIRQ13}}$ is selected
4	ISS12	0	R/W	0: P93/$\overline{\text{IRQ12}}$ is selected 1: PG4/$\overline{\text{ExIRQ12}}$ is selected
3	ISS11	0	R/W	0: PF3/$\overline{\text{IRQ11}}$ is selected 1: PG3/$\overline{\text{ExIRQ11}}$ is selected
2	ISS10	0	R/W	0: PF2/$\overline{\text{IRQ10}}$ is selected 1: PG2/$\overline{\text{ExIRQ10}}$ is selected
1	ISS9	0	R/W	0: PF1/$\overline{\text{IRQ9}}$ is selected 1: PG1/$\overline{\text{ExIRQ9}}$ is selected
0	ISS8	0	R/W	0: PF0/$\overline{\text{IRQ8}}$ is selected 1: PG0/$\overline{\text{ExIRQ8}}$ is selected

- ISSR

Bit	Bit Name	Initial Value	R/W	Description
7	ISS7	0	R/W	0: P67/$\overline{\text{IRQ7}}$ is selected 1: PH1/$\overline{\text{ExIRQ7}}$ is selected
6 to 0	—	0	R/W	Reserved The initial values should not be changed.

6	WUE14SC	0	R/W	These bits select the source that generates interrupt request at wake-up event interrup (WUE15 to WUE8). 0: Interrupt request generated at falling edg $\overline{\text{WUEn}}$ input 1: Interrupt request generated at rising edg $\overline{\text{WUEn}}$ input (n = 15 to 8)
5	WUE13SC	0	R/W	
4	WUE12SC	0	R/W	
3	WUE11SC	0	R/W	
2	WUE10SC	0	R/W	
1	WUE9SC	0	R/W	
0	WUE8SC	0	R/W	

- WUESR

Bit	Bit Name	Initial Value	R/W	Description
7	WUE15F	0	R/(W)*	Wake-Up Input Interrupt (WUE15 to WUE8 Request Flag Register These bits are status flags that indicate tha up input interrupts (WUE15 to WUE8) are requested. [Setting condition] • When a wake-up input interrupt is gene [Clearing condition] • When 0 is written after reading 1
6	WUE14F	0	R/(W)*	
5	WUE13F	0	R/(W)*	
4	WUE12F	0	R/(W)*	
3	WUE11F	0	R/(W)*	
2	WUE10F	0	R/(W)*	
1	WUE9F	0	R/(W)*	
0	WUE8F	0	R/(W)*	

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

5.4 Interrupt Sources

5.4.1 External Interrupt Sources

The interrupt sources of external interrupts are NMI, IRQ15 to IRQ0, KIN15 to KIN0 ar to WUE8. These interrupts can be used to restore this LSI from software standby mode.

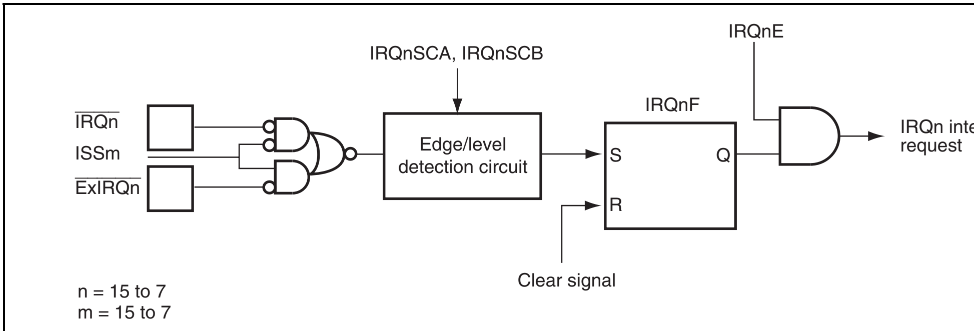
(1) NMI Interrupt

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

(2) IRQ15 to IRQ0 Interrupts:

Interrupts IRQ15 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pin $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$. Interrupts IRQ15 to IRQ0 have the following features:

- The interrupt exception handling for interrupt requests IRQ15 to IRQ0 can be startec independent vector address.
- Using ISCR, it is possible to select whether an interrupt is generated by a low level, edge, rising edge, or both edges, at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$
- Enabling or disabling of interrupt requests IRQ15 to IRQ0 can be selected with IER.
- The status of interrupt requests IRQ15 to IRQ0 is indicated in ISR. ISR flags can be 0 by software.



Note: Switching between the $\overline{\text{IRQ6}}$ and $\overline{\text{ExIRQ6}}$ pins is controlled by the EIVS bit.

Figure 5.4 Block Diagram of Interrupts IRQ15 to IRQ0

(3) KIN15 to KIN0 Interrupts

Interrupts KIN15 to KIN0 are requested by the input signals on pins $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$. Fur interrupts KIN15 to KIN0 change as follows according to the setting of the EIVS bit in sy control register 3 (SYSCR3).

- H8S/2140B Group compatible vector mode (EIVS = 0 in SYSCR3)
  - — Interrupts KIN15 to KIN8 correspond to interrupt IRQ7, and interrupts KIN7 to K correspond to interrupt IRQ6. The pin conditions for generating an interrupt reque whether the interrupt request is enabled, interrupt control level setting, and status o interrupt request for the above interrupts are in accordance with the settings and st the relevant interrupts IRQ7 and IRQ6.
  - — KIN15 to KIN0 interrupt requests can be masked by using KMIMRA and KMIMI
  - — If the $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$ pins are specified to be used as key-sensing interrupt input pi interrupt sensing condition for the corresponding interrupt source (IRQ6 or IRQ7) set to low-level sensing or falling-edge sensing.

An IRQ6 interrupt is enabled only by input to the $\overline{\text{ExIRQ6}}$ pin. The $\overline{\text{IRQ6}}$ pin is c
available for a KIN interrupt input, and functions as the $\overline{\text{KIN6}}$ pin. The initial val
KMIMR6 bit is 1. For the IRQ7 interrupt, either the $\overline{\text{IRQ7}}$ pin or $\overline{\text{ExIRQ7}}$ pin can
selected as the input pin using the ISS7 bit. The IRQ7 interrupt is not affected by
settings of bits KMIMR15 to KMIMR8. The detection of interrupts KIN15 to KI
not depend on whether the relevant pin has been set for input or output. Therefor
pin is used as an external interrupt input pin, clear the DDR bit of the correspond
0 so it is not used as an I/O pin for another function.

(4) WUE15 to WUE8 Interrupts

Interrupt requests WUE15 to WUE8 can be configured regardless of the setting of the E
system control register 3 (SYSCR3).

A block diagram of interrupts WUE15 to WUE8 is shown in figure 5.5.

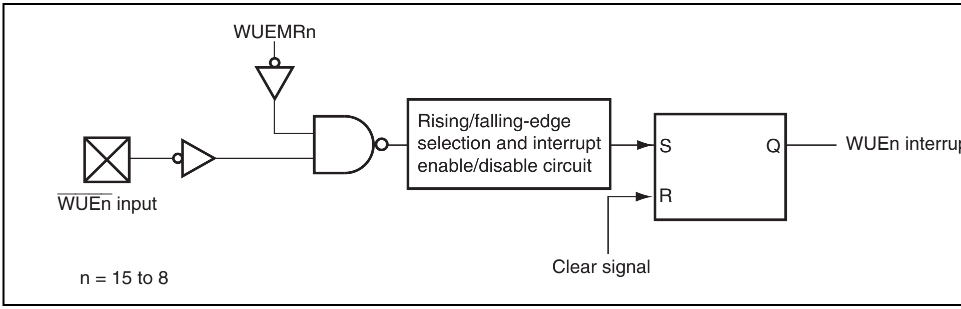

WUEMRn
Rising/falling-edge selection and interrupt enable/disable circuit
S
Q
R
WUEn interru
$\overline{\text{WUEn}}$ input
Clear signal
n = 15 to 8

Figure 5.5 Block Diagram of Interrupts WUE15 to WUE8

5.5 Interrupt Exception Handling Vector Tables

Tables 5.4 and 5.5 list interrupt exception handling sources, vector addresses, and interru
priorities. H8S/2140B Group compatible vector mode or extended vector mode can be se
the vector addresses by the EIVS bit in system control register 3 (SYSCR3).

For default priorities, the lower the vector number, the higher the priority. Modules set at
priority will conform to their default priorities. Priorities within a module are fixed.

An interrupt control level can be specified for a module to which an ICR bit is assigned. I
requests from modules that are set to interrupt control level 1 (priority) by the interrupt co
level and the I and UI bits in CCR are given priority and processed before interrupt reque
modules that are set to interrupt control level 0 (no priority).

Table 5.5 Interrupt Sources, Vector Addresses, and Interrupt Priorities (H8S/2140B Group Compatible Vector Mode)

Origin of Interrupt Source	Name	Vector Number	Vector Address	ICR
			Advanced Mode	
External pin	NMI	7	H'00001C	—
	IRQ0	16	H'000040	ICRA7
	IRQ1	17	H'000044	ICRA6
	IRQ2 IRQ3	18 19	H'000048 H'00004C	ICRA5
	IRQ4 IRQ5	20 21	H'000050 H'000054	ICRA4
	IRQ6, KIN7 to KIN0 IRQ7, KIN15 to KIN8	22 23	H'000058 H'00005C	ICRA3

External pin	WUE15 to WUE8	33	H'000084	ICRD4
TPU_0	TGI0A (TGR0A input capture/compare match) TGI0B (TGR0B input capture/compare match) TGI0C (TGR0C input capture/compare match) TGI0D (TGR0D input capture/compare match) TGI0V (Overflow 0)	34 35 36 37 38	H'000088 H'00008C H'000090 H'000094 H'000098	ICRD3
TPU_1	TGI1A (TGR1A input capture/compare match) TGI1B (TGR1B input capture/compare match) TGI1V (Overflow 1) TGI1U (Underflow 1)	39 40 41 42	H'00009C H'0000A0 H'0000A4 H'0000A8	ICRD2
TPU_2	TGI2A (TGR2A input capture/compare match) TGI2B (TGR2B input capture/compare match) TGI2V (Overflow 2) TGI2U (Underflow 2)	43 44 45 46	H'0000AC H'0000B0 H'0000B4 H'0000B8	ICRD1
—	Reserved for system use	47	H'0000BC	—
TCM_0	TICI0 (Input capture) TCMI0 (Compare match) TOVMI0 (Cycle overflow) TUDI0 (Cycle underflow) TOVI0 (Overflow)	48	H'0000C0	ICRB6
TCM_1	TICI1 (Input capture) TCMI1 (Compare match) TOVMI1 (Cycle overflow) TUDI1 (Cycle underflow) TOVI1 (Overflow)	49	H'0000C4	

TDP_0	TICI0 (Input capture) TCMI0 (Compare match) TPDMXI0 (Cycle overflow) TPDMNI0 (Cycle underflow) TWDMNI0 (Pulse width lower limit underflow) TWDMXI0 (Pulse width upper limit overflow) TOVI0 (Overflow)	52	H'0000D0	ICRB5
TDP_1	TICI1 (Input capture) TCMI1 (Compare match) TPDMXI1 (Cycle overflow) TPDMNI1 (Cycle underflow) TWDMNI1 (Pulse width lower limit underflow) TWDMXI1 (Pulse width upper limit overflow) TOVI1 (Overflow)	53	H'0000D4	
TDP_2	TICI2 (Input capture) TCMI2 (Compare match) TPDMXI2 (Cycle overflow) TPDMNI2 (Cycle underflow) TWDMNI2 (Pulse width lower limit underflow) TWDMXI2 (Pulse width upper limit overflow) TOVI2 (Overflow)	54	H'0000D8	
—	Reserved for system use	55	H'0000DC	—
External pin	IRQ8 IRQ9 IRQ10 IRQ11	56 57 58 59	H'0000E0 H'0000E4 H'0000E8 H'0000EC	ICRD7
	IRQ12 IRQ13 IRQ14 IRQ15	60 61 62 63	H'0000F0 H'0000F4 H'0000F8 H'0000FC	ICRD6

TMR_Y	CMIBY (Compare match B) OVIY (Overflow) ICIX (Input capture) CMIAX (Compare match A) CMIBX (Compare match B) OVIX (Overflow)	73 74 75 76 77 78	H'000124 H'000128 H'00012C H'000130 H'000134 H'000138	
FSI	FSII (Transmission/reception completion)	79	H'00013C	ICRC2
—	Reserved for system use	80 81	H'000140 H'000144	—
SCIF	SCIF (SCIF interrupt)	82	H'000148	ICRC7
—	Reserved for system use	83	H'00014C	—
SCI_1	ERI1 (Reception error 1) RXI1 (Reception completion 1) TXI1 (Transmission data empty 1) TEI1 (Transmission end 1)	84 85 86 87	H'000150 H'000154 H'000158 H'00015C	ICRC6
SCI_2	ERI2 (Reception error) RXI2 (Reception completion) TXI2 (Transmission data empty 2) TEI2 (Transmission end 2)	88 89 90 91	H'000160 H'000164 H'000168 H'00016C	ICRC5
IIC_0	IICI0 (1-byte transmission/reception completion)	92	H'000170	ICRC4
CIR	RENDI (Reception end) OVEI (Overrun error) REPI (Repeat detection) FREI (Framing error) ABI (Abort) HEADFI (Header detection)	93	H'000174	ICRB4
IIC_1	IICI1 (1-byte transmission/reception completion)	94	H'000178	ICRC3
IIC_2	IICI2 (1-byte transmission/reception completion)	95	H'00017C	

	KBTID (Transmission completion D)/KBCD (1st KCLKD)	103	H'00019C	
FSI	LFSII (Command reception)/(Write reception)	104	H'0001A0	ICRC1
—	Reserved for system use	105	H'0001A4	—
LPC	OBEI (ODR1 to 4 transmission completion)	106	H'0001A8	ICRC1
	IBFI4 (IDR4 reception completion)	107	H'0001AC	
	ERR1 (Transfer error, etc.)	108	H'0001B0	
	IBFI1 (IDR1 reception completion)	109	H'0001B4	
	IBFI2 (IDR2 reception completion)	110	H'0001B8	
	IBFI3 (IDR3 reception completion)	111	H'0001BC	
—	Reserved for system use	112 │ 127	H'0001C0 │ H'0001FC	—

	IRQ4 IRQ5	20 21	H'000050 H'000054	ICRA4
	IRQ6 IRQ7	22 23	H'000058 H'00005C	ICRA3
—	Reserved for system use	24	H'000060	—
WDT_0	WOVI0 (Interval timer)	25	H'000064	ICRA1
WDT_1	WOVI1 (Interval timer)	26	H'000068	ICRA0
—	Address break	27	H'00006C	—
A/D converter	ADI (A/D conversion end)	28	H'000070	ICRB7
—	Reserved for system use	29	H'000074	—
External pin	KIN7 to KIN0 KIN15 to KIN8	30 31	H'000078 H'00007C	ICRD5
—	Reserved for system use	32	H'000080	—
External pin	WUE15 to WUE8	33	H'000084	ICRD4
TPU_0	TGI0A (TGR0A input capture/compare match) TGI0B (TGR0B input capture/compare match) TGI0C (TGR0C input capture/compare match) TGI0D (TGR0D input capture/compare match) TGI0V (Overflow 0)	34 35 36 37 38	H'000088 H'00008C H'000090 H'000094 H'000098	ICRD3
TPU_1	TGI1A (TGR1A input capture/compare match) TGI1B (TGR1B input capture/compare match) TGI1V (Overflow 1) TGI1U (Underflow 1)	39 40 41 42	H'00009C H'0000A0 H'0000A4 H'0000A8	ICRD2

	TUDI0 (Cycle underflow) TOVI0 (Overflow)			
TCM_1	TICI1 (Input capture) TCMI1 (Compare match) TOVMI1 (Cycle overflow) TUDI1 (Cycle underflow) TOVI1 (Overflow)	49	H'0000C4	
TCM_2	TICI2 (Input capture) TCMI2 (Compare match) TOVMI2 (Cycle overflow) TUDI2 (Cycle underflow) TOVI2 (Overflow)	50	H'0000C8	
TCM_3	TICI3 (Input capture) TCMI3 (Compare match) TOVMI3 (Cycle overflow) TUDI3 (Cycle underflow) TOVI3 (Overflow)	51	H'0000CC	
TDP_0	TICI0 (Input capture) TCMI0 (Compare match) TPDMXI0 (Cycle overflow) TPDMNI0 (Cycle underflow) TWDMNI0 (Pulse width lower limit underflow) TWDMXI0 (Pulse width upper limit overflow) TOVI0 (Overflow)	52	H'0000D0	ICRB5
TDP_1	TICI1 (Input capture) TCMI1 (Compare match) TPDMXI1 (Cycle overflow) TPDMNI1 (Cycle underflow) TWDMNI1 (Pulse width lower limit underflow) TWDMXI1 (Pulse width upper limit overflow) TOVI1 (Overflow)	53	H'0000D4	

External pin	IRQ8	56	H'0000E0	ICRD7
	IRQ9	57	H'0000E4	
	IRQ10	58	H'0000E8	
	IRQ11	59	H'0000EC	
	IRQ12	60	H'0000F0	ICRD6
	IRQ13	61	H'0000F4	
	IRQ14	62	H'0000F8	
	IRQ15	63	H'0000FC	
TMR_0	CMIA0 (Compare match A)	64	H'000100	ICRB3
	CMIB0 (Compare match B)	65	H'000104	
	OVI0 (Overflow)	66	H'000108	
—	Reserved for system use	67	H'00010C	—
TMR_1	CMIA1 (Compare match A)	68	H'000110	ICRB2
	CMIB1 (Compare match B)	69	H'000114	
	OVI1 (Overflow)	70	H'000118	
—	Reserved for system use	71	H'00011C	—
TMR_X	CMIAY (Compare match A)	72	H'000120	ICRB1
TMR_Y	CMIBY (Compare match B)	73	H'000124	
	OVIY (Overflow)	74	H'000128	
	ICIX (Input capture)	75	H'00012C	
	CMIAX (Compare match A)	76	H'000130	
	CMIBX (Compare match B)	77	H'000134	
	OVIX (Overflow)	78	H'000138	
FSI	FSII (Transmission/reception completion)	79	H'00013C	ICRC2
—	Reserved for system use	80	H'000140	—
		81	H'000144	
SCIF	SCIF (SCIF interrupt)	82	H'000148	ICRC7
—	Reserved for system use	83	H'00014C	—
SCI_1	ERI1 (Reception error 1)	84	H'000150	ICRC6
	RXI1 (Reception completion 1)	85	H'000154	
	TXI1 (Transmission data empty 1)	86	H'000158	
	TEI1 (Transmission end 1)	87	H'00015C	

	ABI (Abort) HEADFI (Header detection)			
IIC_1	IICI1 (1-byte transmission/reception completion)	94	H'000178	ICRC3
IIC_2	IICI2 (1-byte transmission/reception completion)	95	H'00017C	
PS2	KBIA (Reception completion A) KBIB (Reception completion B) KBIC (Reception completion C) KBTIA (Transmission completion A)/ KBCA (1st KCLKA) KBTIB (Transmission completion B)/ KBCB (1st KCLKB) KBTIC (Transmission completion C)/ KBCC (1st KCLKC) KBID (Reception completion D) KBTID (Transmission completion D)/KBCD (1st KCLKD)	96 97 98 99 100 101 102 103	H'000180 H'000184 H'000188 H'00018C H'000190 H'000194 H'000198 H'00019C	ICRB0
FSI	LFSII (Command reception)/(Write reception)	104	H'0001A0	ICRC1
—	Reserved for system use	105	H'0001A4	—
LPC	OBEI (ODR1 to 4 transmission completion) IBFI4 (IDR4 reception completion) ERR1 (Transfer error, etc.) IBFI1 (IDR1 reception completion) IBFI2 (IDR2 reception completion) IBFI3 (IDR3 reception completion)	106 107 108 109 110 111	H'0001A8 H'0001AC H'0001B0 H'0001B4 H'0001B8 H'0001BC	ICRC1
—	Reserved for system use	112 | 127	H'0001C0 | H'0001FC	—

Mode	INTM1	INTM0	Registers	Mask Bits	Description
0	0	0	ICR	I	Interrupt mask control is perfo the I bit. Priority levels can be ICR.
1	0	1	ICR	I, UI	3-level interrupt mask control performed by the I and UI bits levels can be set with ICR.

Figure 5.6 shows a block diagram of the priority determination circuit.

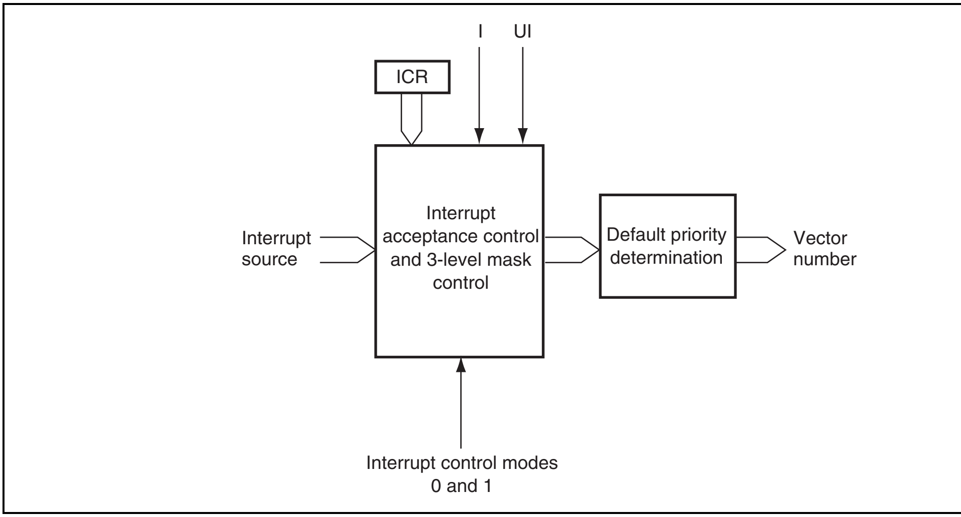

I
UI
ICR
Interrupt source
Interrupt acceptance control and 3-level mask control
Default priority determination
Vector number
Interrupt control modes 0 and 1

Figure 5.6 Block Diagram of Interrupt Control Operation

0	0		All interrupts (interrupt control level 1 ha priority)
	1	*	NMI and address break interrupts
1	0	*	All interrupts (interrupt control level 1 ha priority)
	1	0	NMI, address break, and interrupt contr interrupts
		1	NMI and address break interrupts

[Legend]

*: Don't care

(2) Default Priority Determination

The priority is determined for the selected interrupt, and a vector number is generated.

If the same value is set for ICR, acceptance of multiple interrupts is enabled, and so only interrupt source with the highest priority according to the preset default priorities is select has a vector number generated.

Interrupt sources with a lower priority than the accepted interrupt source are held pending

O: Interrupt operation control is performed

IM: Used as an interrupt mask bit

PR: Priority is set

—: Not used

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests other than NMI and address break are ma ICR and the I bit of CCR in the CPU. Figure 5.7 shows a flowchart of the interrupt acce operation.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, a interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller onl an interrupt request with interrupt control level 1 (priority), and holds pending an int request with interrupt control level 0 (no priority). If several interrupt requests are is interrupt request with the highest priority is accepted according to the priority order, interrupt handling is requested to the CPU, and other interrupt requests are held pend
3. If the I bit in CCR is set to 1, the interrupt controller holds pending interrupt request than NMI and address break. If the I bit is cleared to 0, any interrupt request is accep
4. When the CPU accepts an interrupt request, it starts interrupt exception handling aft 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 the stack shows the address of the first instruction to be executed after returning fro interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except for NMI and addre interrupts.

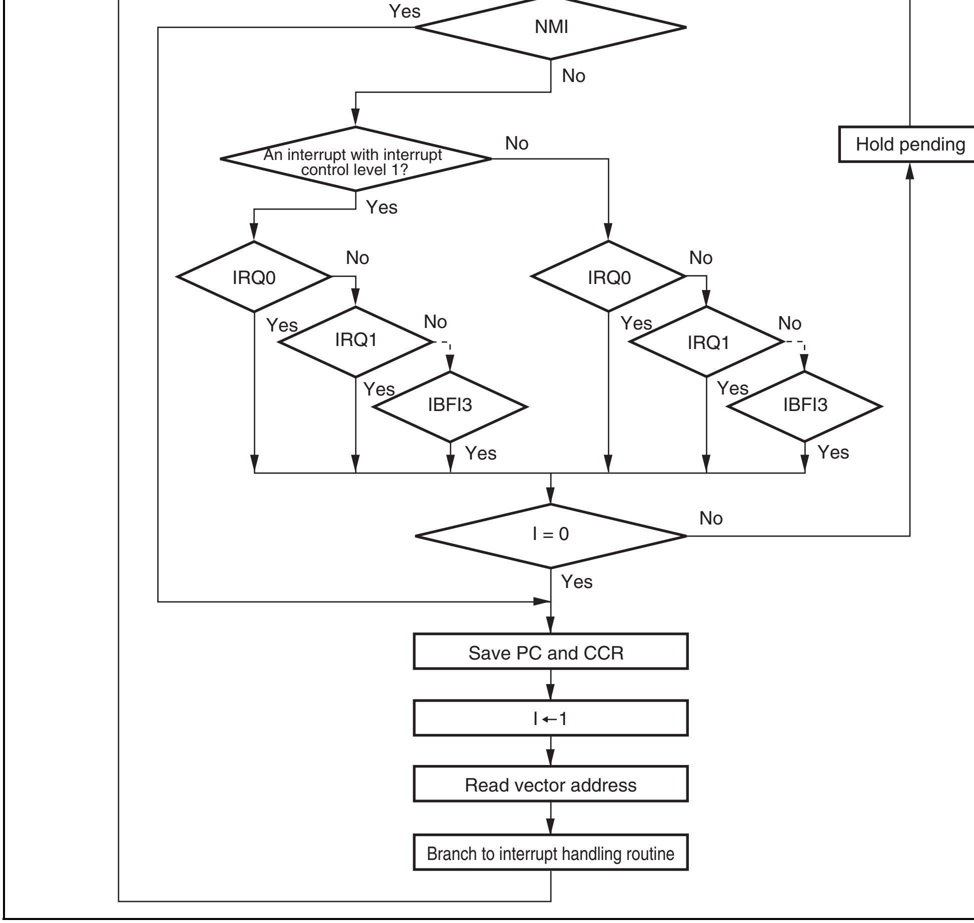


Figure 5.7 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Contro

to 1, and ICRA to ICRD are set to H'20, H'00, H'00, and H'00, respectively (IRQ2 and I
interrupts are set to interrupt control level 1, and other interrupts are set to interrupt cont
0) is shown below. Figure 5.8 shows a state transition diagram.

- All interrupt requests are accepted when I = 0. (Priority order: NMI > IRQ2 > IRQ3 break > IRQ0 > IRQ1 …)
- Only NMI, IRQ2, IRQ3, and address break interrupt requests are accepted when I = 0.
- Only NMI and address break interrupt requests are accepted when I = 1 and UI = 1.

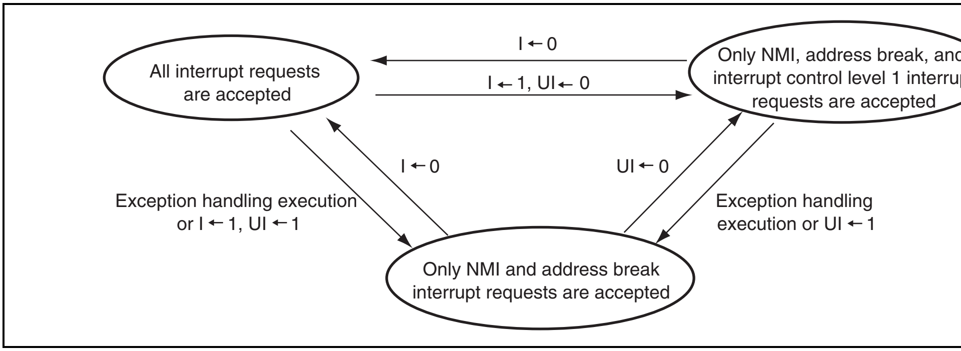

All interrupt requests are accepted
I ← 0
I ← 1, UI ← 0
Only NMI, address break, and interrupt control level 1 interru requests are accepted
I ← 0
UI ← 0
Exception handling execution or I ← 1, UI ← 1
Exception handling execution or UI ← 1
Only NMI and address break interrupt requests are accepted

Figure 5.8 State Transition in Interrupt Control Mode 1

An interrupt request with interrupt control level 0 is accepted when the I bit is cleared

When both the I and UI bits are set to 1, only NMI and address break interrupt reques accepted, and other interrupts are held pending.

When the I bit is cleared to 0, the UI bit does not affect acceptance of interrupt reques

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 the stack shows the address of the first instruction to be executed after returning from interrupt handling routine.
6. The I and UI bits in CCR are set to 1. This masks all interrupts except for NMI and ad break interrupts.
7. The CPU generates a vector address for the accepted interrupt request and starts execu the interrupt handling routine at the address indicated by the contents of the vector ad the vector table.

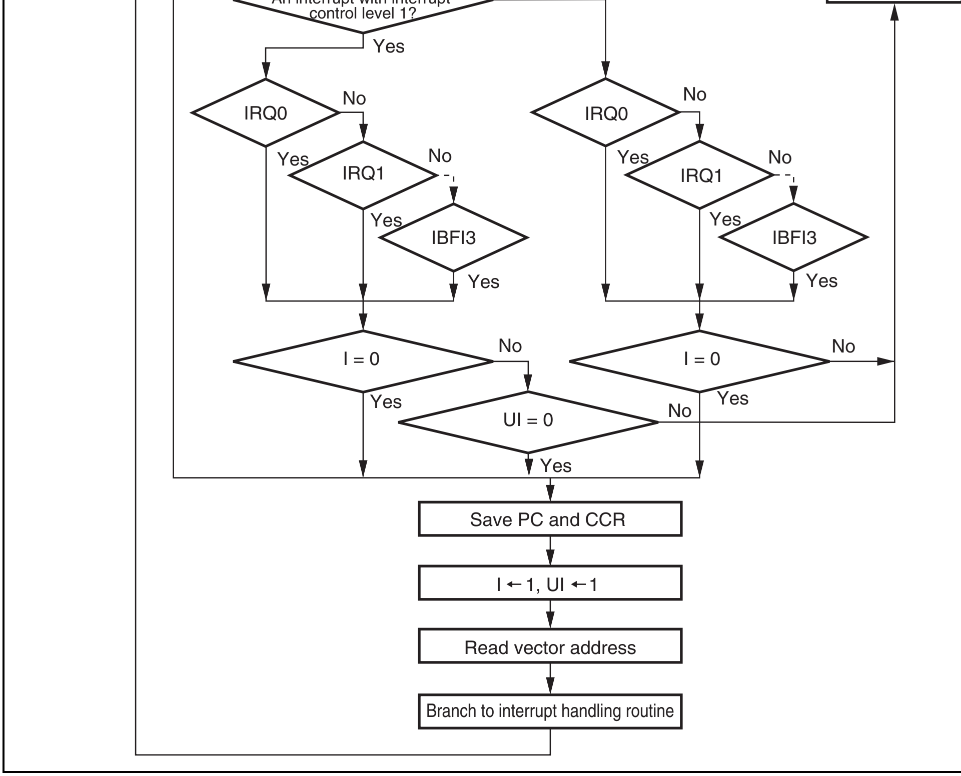
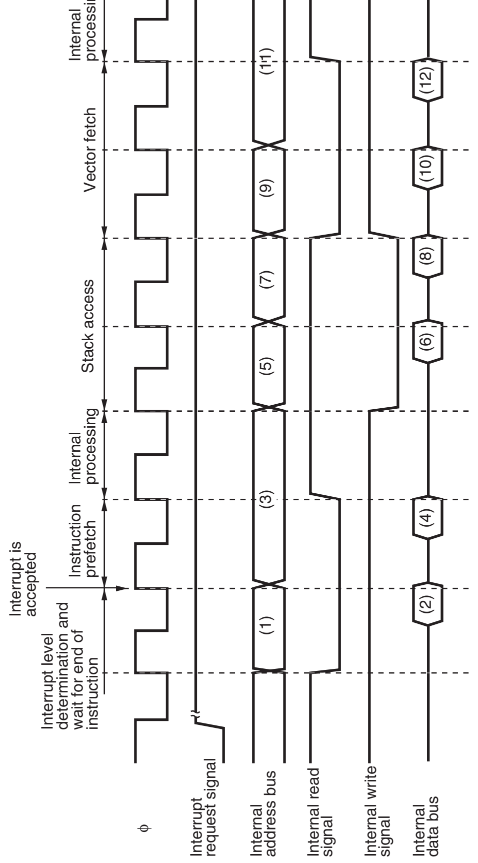


Figure 5.9 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Contr



(1) Instruction prefetch address (Not executed. Address is saved as PC contents, becoming return address.)
(2) (4) Instruction code (Not executed.)
(3) Instruction prefetch address (Not executed.)
(5) SP – 2
(7) SP – 4

(6) (8) Saved PC and CCR
(9) (11) Vector address
(10) (12) Start address of interrupt handling routine (content vector address)
(13) Start address of interrupt handling routine ((13) = (
(14) First instruction in interrupt handling routine

Figure 5.10 Interrupt Exception Handling

3	Saving of PC and CCR in stack	2
4	Vector fetch	2
5	Instruction fetch*[3]	2
6	Internal processing*[4]	2
	Total (using on-chip memory)	12 to 32

Notes: 1. Two states in case of internal interrupt.

2. Refers to MULXS and DIVXS instructions.

3. Prefetch after interrupt acceptance and prefetch of interrupt handling routine.

4. Internal processing after interrupt acceptance and internal processing after ve

5.7.2 Block Diagram

Figure 5.11 shows a block diagram of the address break function.

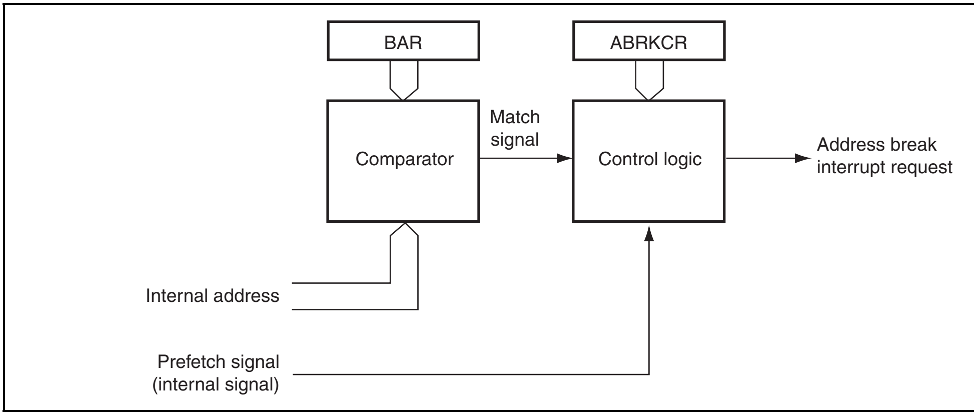

BAR
ABRKCR
Match signal
Comparator
Control logic
Address break interrupt request
Internal address
Prefetch signal (internal signal)

Figure 5.11 Block Diagram of Address Break Function

1. Set the break address in bits A23 to A1 in BAR.
2. Set the BIE bit in ABRKCR to 1 to enable address breaks. An address break will not
   requested if the BIE bit is cleared to 0.

When the setting condition occurs, the CMF flag in ABRKCR is set to 1 and an interrup
requested. If necessary, the source should be identified in the interrupt handling routine.

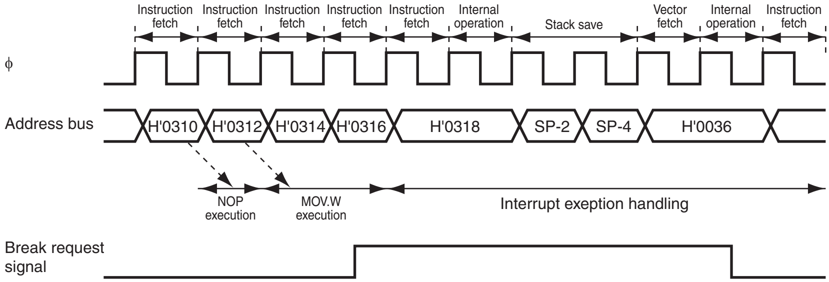
5.7.4 Usage Notes

- With the address break function, the address at which the first instruction byte is loc
  should be specified as the break address. Occurrence of the address break condition
  recognized for other addresses.
- If a branch instruction (Bcc, BSR) jump instruction (JMP, JSR), RTS instruction, or
  instruction is located immediately before the address set in BAR, execution of this in
  will output a prefetch signal for that address, and an address break may be requested
  be prevented by not making a break address setting for an address immediately follo
  of these instructions, or by determining within the interrupt handling routine whethe
  handling was initiated by a genuine condition occurrence.
- As an address break interrupt is generated by a combination of the internal prefetch s
  address, the timing of the start of interrupt exception handling depends on the conte
  execution cycle of the instruction at the set address and the preceding instruction. Fi
  shows some address timing examples.

H'0314 NOP
H'0316 NOP

next address, H'0314; fetch from address H'0316 starts after end of exception handling.

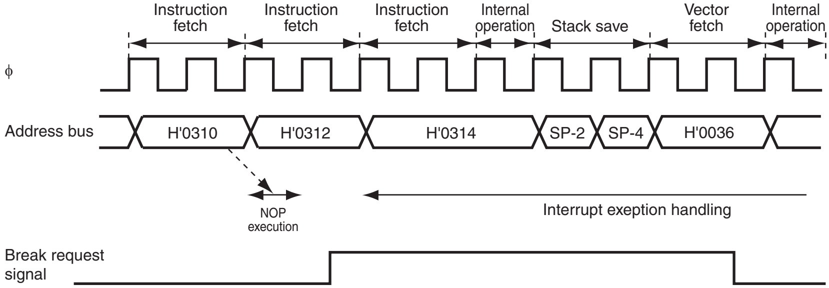
- Program area in on-chip memory, 2-state execution instruction at specified break address



H'0310 NOP
H'0312 MOV.W #xx : 16,Rd ◄——Breakpoint MOV instruction is executed at breakpoint address H'0312,
H'0314 NOP
H'0316 NOP

NOP instruction at next address, H'0316, is not executed;
fetch from address H'0316 starts after end of exception handlin

- Program area in external memory (2-state access, 16-bit-bus access), 1-state execution instruction at specified break address (Not available in this LSI)



H'0310 NOP
H'0312 NOP ◄——Breakpoint NOP instruction at breakpoint address H'0312 is not executed;
H'0314 NOP
H'0316 NOP

fetch from address H'0312 starts after end of exception handling.

Figure 5.12 Examples of Address Break Timing

handling will be executed for the higher-priority interrupt, and the lower-priority interru ignored. The same rule is also applied when an interrupt source flag is cleared to 0. Figu shows an example where the CMIEA bit in TCR of the TMR is cleared to 0. The above will not occur if an interrupt enable bit or interrupt source flag is cleared to 0 while the i disabled.

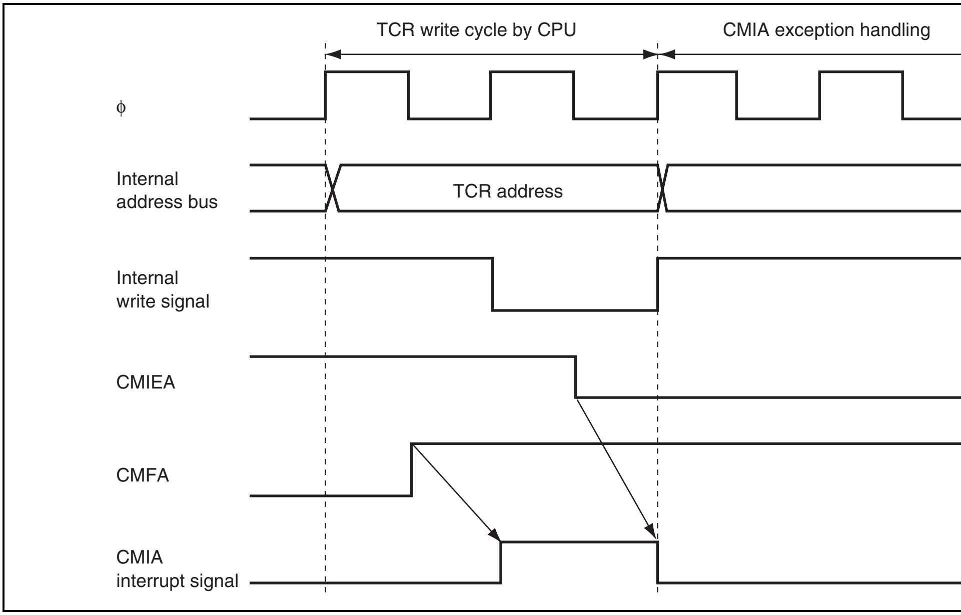


Figure 5.13 Conflict between Interrupt Generation and Disabling

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

With the EEPMOV.W instruction, if an interrupt request is issued during data transfer, in
exception handling starts at a break in the transfer cycles. The PC value saved on the stac
case is the address of the next instruction. Therefore, if an interrupt is generated during ex
of an EEPMOV.W instruction, the following coding should be used.

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

5.8.4 Vector Address Switching

Switching between H8S/2140B Group compatible vector mode and extended vector mod
done in a state with no interrupts occurring.

If the EIVS bit in SYSCR3 is changed from 0 to 1 when interrupt input is enabled becaus
$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$ pins are set at low level, a falling edge is detected
causing an interrupt to be generated. The vector mode must be changed when interrupt in
disabled, that is the $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$ pins are set at high level.

The noise canceller should be switched when the external input pins ($\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$, $\overline{\text{ExIR}}$
$\overline{\text{ExIRQ8}}$, $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$, and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) are high.

5.8.7 IRQ Status Register (ISR)

Since IRQnF may be set to 1 according to the pin state after reset, the ISR should be rea
reset, and then write 0 in IRQnF (n = 15 to 0).

Table 6.1 Register Configuration

					Da
Register Name	**Abbreviation**	**R/W**	**Initial Value**	**Address**	**W**
Bus control register	BCR	R/W	H'D3	H'FFC6	8
Wait state control register	WSCR	R/W	H'F3	H'FFC7	8

6.1.1 Bus Control Register (BCR)

		Initial		
Bit	**Bit Name**	**Value**	**R/W**	**Description**
7	—	1	R/W	Reserved The initial value should not be changed.
6	ICIS0	1	R/W	Idle Cycle Insertion The initial value should not be changed.
5	BRSTRM	0	R/W	Burst ROM Enable The initial value should not be changed.
4	BRSTS1	1	R/W	Burst Cycle Select 1 The initial value should not be changed.
3	BRSTS0	0	R/W	Burst Cycle Select 0 The initial value should not be changed.
2	—	0	R/W	Reserved The initial value should not be changed.
1	IOS1	1	R/W	IOS Select 1 and 0
0	IOS0	1	R/W	The initial value should not be changed.

				The initial value should not be changed.
3	WMS1	0	R/W	Wait Mode Select 1 and 0
2	WMS0	0	R/W	The initial value should not be changed.
1	WC1	1	R/W	Wait Count 1 and 0
0	WC0	1	R/W	The initial value should not be changed.

In addition, ports 1 to 3, C, and D can drive a LED (5 mA sink current). P52, P97, P86, A, G, and I are NMOS push-pull outputs and 5-V tolerant inputs. PE4 and PE2 to PE0 a tolerant inputs.

Ports I and J are not supported by TFP-144V and TLP-145V.

		2	P12	—	—		
		1	P11	—	—		
		0	P10	—	—		
Port 2	General I/O port	7	P27	—	—	O	O
		6	P26	—	—		
		5	P25	—	—		
		4	P24	—	—		
		3	P23	—	—		
		2	P22	—	—		
		1	P21	—	—		
		0	P20	—	—		
Port 3	General I/O port also functioning as LPC input/output	7	P37/SERIRQ	—	—	O	O
		6	P36	LCLK	—		
		5	P35	$\overline{\text{LRESET}}$	—		
		4	P34	$\overline{\text{LFRAME}}$	—		
		3	P33/LAD3	—	—		
		2	P32/LAD2	—	—		
		1	P31/LAD1	—	—		
		0	P30/LAD0	—	—		

	TMR_1, IIC_1, and SCI_2 inputs/outputs			TCMMCI1			
		2	P42/SDA1	TCMCYI1			
		1	P41	RxD2/TCMCKI0/ TCMMCI0	TMO0		
		0	P40	TMI0/TCMCYI0	TxD2		
Port 5	General I/O port also functioning as IIC_0 and SCIF inputs/outputs	2	P52/SCL0	—	—	—	—
		1	P51	FRxD	—		
		0	P50	—	FTxD		
Port 6	General I/O port also functioning as interrupt input and keyboard input	7	P67	$\overline{\text{KIN7}}$/$\overline{\text{IRQ7}}$	—	O	—
		6	P66	$\overline{\text{KIN6}}$/$\overline{\text{IRQ6}}$	—		
		5	P65	$\overline{\text{KIN5}}$	—		
		4	P64	$\overline{\text{KIN4}}$	—		
		3	P63	$\overline{\text{KIN3}}$	—		
		2	P62	$\overline{\text{KIN2}}$	—		
		1	P61	$\overline{\text{KIN1}}$	—		
		0	P60	$\overline{\text{KIN0}}$	—		

		1	—	P71/AN1	—		
		0	—	P70/AN0	—		
Port 8	General I/O port also functioning as interrupt input, IIC_1, SCI_1, and LPC inputs/outputs	6	P86/SCK1/ SCL1	$\overline{\text{IRQ5}}$	—	—	—
		5	P85	RxD1/$\overline{\text{IRQ4}}$	—		
		4	P84	$\overline{\text{IRQ3}}$	TxD1		
		3	P83	$\overline{\text{LPCPD}}$	—		
		2	P82/$\overline{\text{CLKRUN}}$	—	—		
		1	P81/GA20	—	—		
		0	P80/$\overline{\text{PME}}$	—	—		
Port 9	General I/O port also functioning as external sub-clock, interrupt input, IIC_0 input/output, and system clock output	7	P97/SDA0	$\overline{\text{IRQ15}}$	—	O (P95 to P90)	—
		6	P96	EXCL	ϕ		
		5	P95	$\overline{\text{IRQ14}}$	—		
		4	P94	$\overline{\text{IRQ13}}$	—		
		3	P93	$\overline{\text{IRQ12}}$	—		
		2	P92	$\overline{\text{IRQ0}}$	—		
		1	P91	$\overline{\text{IRQ1}}$	—		
		0	P90	$\overline{\text{IRQ2}}$	—		

		1	PA1/PS2DD	$\overline{\text{KIN9}}$	—		
		0	PA0/PS2DC	$\overline{\text{KIN8}}$	—		
Port B	General I/O port also functioning as LPC, SCIF and FSI inputs/outputs and PWMU output	7	PB7	—	$\overline{\text{RTS}}$/FSISS	O	—
		6	PB6	$\overline{\text{CTS}}$	FSICK		
		5	PB5	FSIDI	$\overline{\text{DTR}}$		
		4	PB4	$\overline{\text{DSR}}$	FSIDO		
		3	PB3	$\overline{\text{DCD}}$	PWMU1B		
		2	PB2	$\overline{\text{RI}}$	PWMU0B		
		1	PB1/LSCI	—	—		
		0	PB0/$\overline{\text{LSMI}}$	—	—		
Port C	General I/O port also functioning as wake-up input and TPU input/output	7	PC7/TIOCB2	TCLKD/$\overline{\text{WUE15}}$	—	O	O
		6	PC6/TIOCA2	$\overline{\text{WUE14}}$	—		
		5	PC5/TIOCB1	TCLKC/$\overline{\text{WUE13}}$	—		
		4	PC4/TIOCA1	$\overline{\text{WUE12}}$	—		
		3	PC3/TIOCD0	TCLKB/$\overline{\text{WUE11}}$	—		
		2	PC2/TIOCC0	TCLKA/$\overline{\text{WUE10}}$	—		
		1	PC1/TIOCB0	$\overline{\text{WUE9}}$	—		
		0	PC0/TIOCA0	$\overline{\text{WUE8}}$	—		

		1	PD1	AN9	—		
		0	PD0	AN8	—		
Port E	General input port also functioning as external sub-clock input, emulator input/output	4	—	PE4*^{1}/ETMS	—	—	—
		3	—	PE3*^{1}	ETDO		
		2	—	PE2*^{1}/ETDI	—		
		1	—	PE1*^{1} /ETCK	—		
		0	—	PE0/ExEXCL	—		
Port F	General I/O port also functioning as interrupt and TDP inputs, TMR_X, TMR_Y, and PWM outputs	7	PF7	—	PWMU5A	O	—
		6	PF6	—	PWMU4A		
		5	PF5	—	PWMU3A		
		4	PF4	—	PWMU2A		
		3	PF3	TDPCKI0/ TDPMCI0/$\overline{\text{IRQ11}}$	TMOX		
		2	PF2	TDPCYI0//$\overline{\text{IRQ10}}$	TMOY		
		1	PF1	$\overline{\text{IRQ9}}$	PWMU1A		
		0	PF0	$\overline{\text{IRQ8}}$	PWMU0A		

	IIC0 to IIC2 inputs/outputs	1	PG1	TMIY1/ TDPCKI1/TDPMCI1/ $\overline{\text{ExIRQ9}}$	—		
		0	PG0	TMIX/TDPCYI1 $\overline{\text{ExIRQ8}}$	—		
Port H	General I/O port also functioning as interrupt and TDP and CIR inputs	5	PH5	—	—	O	—
		4	PH4	—	—		
		3	PH3	—	—		
		2	PH2	CIRI	—		
		1	PH1	TDPCKI2/ TDPMCI1/ $\overline{\text{ExIRQ7}}$	—		
		0	PH0	TDPCYI2/ $\overline{\text{ExIRQ6}}$	—		
Port I	General I/O port	7	PI7*[2]	—	—	—	—
		6	PI6*[2]	—	—		
		5	PI5*[2]	—	—		
		4	PI4*[2]	—	—		
		3	PI3*[2]	—	—		
		2	PI2*[2]	—	—		
		1	PI1*[2]	—	—		
		0	PI0*[2]	—	—		

1	PJ1*[2]	—	—
0	PJ0*[2]	—	—

Notes: 1. Not supported by the system development tool (emulator).

2. Not supported by TFP-144V and TLP-145V.

Port 3	8	O	O	O*	O	—	—	—	—	—
Port 4	8	O	O	O*[2]	—	—	—	—	—	—
Port 5	3	O	O	O*[2]	—	—	—	—	—	—
Port 6	8	O	O	O*[2]	—	O	—	O	O	O
Port 7	8	—	—	O	—	—	—	—	—	—
Port 8	7	O	O	O*[2]	—	—	—	—	—	—
Port 9	8	O	O	O*[2]	O	—	—	—	—	—
Port A	8	O	—	O	—	—	O	—	—	—
Port B	8	O	—	O	O*[2]	—	O	—	—	—
Port C	8	O	—	O	O*[2]	—	O	O	O	O
Port D	8	O	—	O	O*[2]	—	O	—	—	—
Port E	5	—	—	O	—	—	—	—	—	—
Port F	8	O	—	O	O*[2]	—	O	—	—	—
Port G	8	O	—	O	—	—	O	O	O	O
Port H	6	O	—	O	O*[2]	—	O	—	—	—
Port I	8*[1]	O	—	O	—	—	O	—	—	—
Port J	8*[1]	O	—	O	O	—	O	—	—	—

[Legend]

O: Register exists

—: No register exists

Notes: 1. Not supported by TFP-144V and TLP-145V.

2. Valid only when the PORTS bit in the port control register 2 (PTCNT2) is 1.

6	Pn6DDR	0	W	these bits are set to 1 and act as input port cleared to 0. Note: These bits cannot be set with bit mar instructions such as BSET and BCLF
5	Pn5DDR	0	W	
4	Pn4DDR	0	W	
3	Pn3DDR	0	W	
2	Pn2DDR	0	W	
1	Pn1DDR	0	W	
0	Pn0DDR	0	W	

(2) PORTS = 1

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7DDR	0	R/W	The corresponding pins act as output ports these bits are set to 1 and act as input port cleared to 0.
6	Pn6DDR	0	R/W	
5	Pn5DDR	0	R/W	
4	Pn4DDR	0	R/W	
3	Pn3DDR	0	R/W	
2	Pn2DDR	0	R/W	
1	Pn1DDR	0	R/W	
0	Pn0DDR	0	R/W	

4	Pn4DR	0	R/W	for pins corresponding to PnDDR bits set reads out the states of pins correspondin PnDDR bits cleared to 0.
3	Pn3DR	0	R/W	
2	Pn2DR	0	R/W	
1	Pn1DR	0	R/W	
0	Pn0DR	0	R/W	

7.1.3 Input Data Register (PnPIN) (n = 1 to 9 and A to J)

PIN is an 8-bit read-only register that reflects the port pin state. A write to PIN is invalic upper five bits in P5PIN, the upper one bit in P8PIN, the upper three bits in PEPIN, and two bits in PHPIN are reserved.

Bits P1PIN to P9PIN are valid only when PORTS in PTCNT2 is 1.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7PIN	Undefined*	R	When this register is read, the pin states a returned.
6	Pn6PIN	Undefined*	R	
5	Pn5PIN	Undefined*	R	
4	Pn4PIN	Undefined*	R	
3	Pn3PIN	Undefined*	R	
2	Pn2PIN	Undefined*	R	
1	Pn1PIN	Undefined*	R	
0	Pn0PIN	Undefined*	R	

Note: * The initial values of these pins are determined in accordance with the states c to Pn0.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7PCR	0	R/W	For pins in the input state corresponding t this register that have been set to 1, the ir up MOSs are turned on.
6	Pn6PCR	0	R/W	
5	Pn5PCR	0	R/W	
4	Pn4PCR	0	R/W	
3	Pn3PCR	0	R/W	
2	Pn2PCR	0	R/W	
1	Pn1PCR	0	R/W	
0	Pn0PCR	0	R/W	

	Port input	Off	On/Off
Port 6 (KMPCR)	Port output	Off	
	Port input	Off	On/Off
Port 9	Port output	Off	
	Port input	Off	On/Off
Port J	Port output	Off	
	Port input	Off	On/Off

[Legend]

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

On/Off: On when PnDDR = 0 and PnPCR = 1; otherwise off.

	Port input	Off	On/Off
Port F	Port output	Off	
	Port input	Off	On/Off
Port H	Port output	Off	
	Port input	Off	On/Off

[Legend]

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

On/Off: On when the pin is in the input state, PnDDR = 0, and PnODR = 1; otherwise off (w
PORTS = 0).
On when the pin is in the input state, PnDDR = 0, and PnPCR = 1; otherwise off (w
PORTS = 1).

2	Pn2ODR	0	R/W
1	Pn1ODR	0	R/W
0	Pn0ODR	0	R/W

7.1.6 Noise Canceler Enable Register (PnNCE) (n = 6, C, and G)

NCE enables or disables the noise cancel circuit at port n pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7NCE	0	R/W	Noise cancel circuit is enabled when a bi register is set to 1, and the pin setting sta fetched in P6DR or PnPIN in the samplin set by the PnNCCS.
6	Pn6NCE	0	R/W	
5	Pn5NCE	0	R/W	
4	Pn4NCE	0	R/W	
3	Pn3NCE	0	R/W	
2	Pn2NCE	0	R/W	
1	Pn1NCE	0	R/W	
0	Pn0NCE	0	R/W	

2	Pn2NCMC	0	R/W
1	Pn1NCMC	0	R/W
0	Pn0NCMC	0	R/W

7.1.8 Noise Cancel Cycle Setting Register (PnNCCS) (n = 6, C, and G)

NCCS controls the sampling cycles of the noise canceler.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read value is undefined. The write val should always be 0.
2	PnNCCK2	0	R/W	These bits set the sampling cycles of the r canceler. When ϕ is 10 MHz 000: 0.80 μs ϕ/2 001: 12.8 μs ϕ/32 010: 3.3 ms ϕ/8192 011: 6.6 ms ϕ/16384 100: 13.1 ms ϕ/32768 101: 26.2 ms ϕ/65536 110: 52.4 ms ϕ/131072 111: 104.9 ms ϕ/262144
1	PnNCCK1	0	R/W	
0	PnNCCK0	0	R/W	

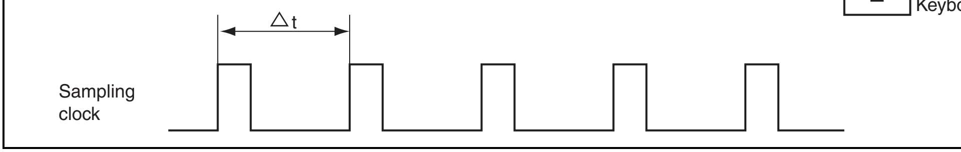


Figure 7.1   Noise Cancel Circuit

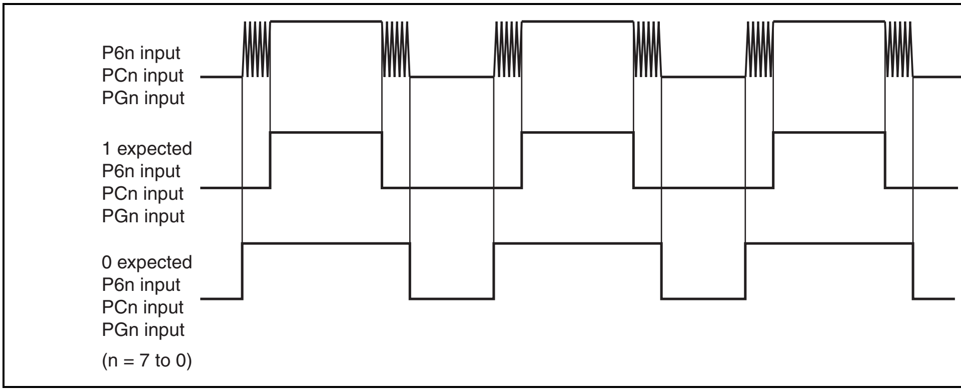


Figure 7.2   Schematic View of Noise Cancel Operation

3	Pn3NOCR	0	R/W	(High level driver is disabled)
2	Pn2NOCR	0	R/W	
1	Pn1NOCR	0	R/W	
0	Pn0NOCR	0	R/W	

N-ch. driver	Off		On	Off	On	
P-ch. driver	Off		Off	On	Off	
Input pull-up MOS	Off	On	Off			
Pin function	Input pin		Output pin			

(2) PORTS = 1

DDR	0		1			
NOCR	—		0		1	
ODR	—		0	1	0	
PCR	0	1	—			
N-ch. Driver	Off		On	Off	On	
P-ch. Driver	Off		Off	On	Off	
Input pull-up MOS	Off	On	Off			
Pin Function	Input pin		Output pin			

(1) P17 to P10

The pin function is switched as shown below according to the P1nDDR bit setting.

		Setting
		I/O Port
Module Name	**Pin Function**	**P1nDDR**
I/O port	P1n output	1
	P1n input (initial setting)	0

(

7.2.2 Port 2

(1) P27 to P20

The pin function is switched as shown below according to the P2nDDR bit setting.

		Setting
		I/O Port
Module Name	**Pin Function**	**P2nDDR**
I/O port	P2n output	1
	P2n input (initial setting)	0

(

		Setting	
		Logical expression	I/O Port
Module Name	Pin Function	LPCENABLE	P3nDDR
LPC	LPC output	1	—
I/O port	P3n output	0	1
	P3n input (initial setting)	0	0

7.2.4 Port 4

(1) P47/PWX1/PWMU5B/TCMCKI3/TCMMCI3

The pin function is switched as shown below according to the combination of the PWM
PWMU and the P47DDR bit.

		Setting		
		PWMX	PWMU	I/O Port
Module Name	Pin Function	PWX1_OE	PWMU5B_OE	P47DDR
PWMX	PWX1 output	1	—	—
PWMU	PWMU5B output	0	1	1
I/O port	P47 output	0	0	1
	P47 input (initial setting)	0	—	0

I/O port	P46 output	0	0	1
	P46 input (initial setting)	0	—	0

(3) P45/PWMU3B/TCMCKI2/TCMMCI2

The pin function is switched as shown below according to the combination of the PWMX P45DDR bit.

		Setting	
		PWMU	I/O Port
Module Name	Pin Function	PWMU3B_OE	P45DDR
PWMU	PWMU3B output	1	1
I/O port	P45 output	0	1
	P45 input (initial setting)	—	0

I/O port	P44 output	0	0	1
	P44 input (initial setting)	0	—	0

(5) P43/TMI1/SCK2/TCMCKI1/TCMMCI1

The pin function is switched as shown below according to the combination of the SCI ar
P43DDR bit.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	SCK2_OE	P43DDR
SCI	SCK2 input/output	1	—
I/O port	P43 output	0	1
	P43 input (initial setting)	0	0

I/O port	P42 output	0	1
	P42 input (initial setting)	0	0

Note: To use this pin as SDA1, clear the IIC1AS and IIC1BS bits in PTCNT1 to 0. The ou format for SDA1 is NMOS output only and direct bus drive is possible. When this p as the P42 output pin, the output format is NMOS push-pull.

(7) P41/TMO0/RxD2/TCMCKI0/TCMMCI0

The pin function is switched as shown below according to the combination of the TMR a P41DDR bit.

		Setting		
		TMR	SCI	I/O Port
Module Name	Pin Function	TMO0_OE	RE	P41DDR
TMR	TMO0 output	1	0	—
SCI	RxD2 input	0	1	—
I/O port	P41 output	0	0	1
	P41 input (initial setting)	0	0	0

Note: To use this pin as TMO0 output, clear the RE bit in SCR of the SCI2 to 0.

	P40 input (initial setting)	0	0

7.2.5 Port 5

(1) P52/SCL0

The pin function is switched as shown below according to the combination of the IIC0A
IIC0BS bits in PTCNT1, ICE bit in ICCR of IIC_0, and the P52DDR bit.

		Setting	
		IIC_0	I/O Port
Module Name	Pin Function	SCL0_OE	P52DDR
IIC	SCL0 output	1	—
I/O port	P52 output	0	1
	P52 input (initial setting)	0	0

Note: To use this pin as SCL0, clear the IIC0AS and IIC0BS bits in PTCNT1 to 0. The o
format for SCL0 is NMOS output only and direct bus drive is possible. When this
as the P52 output pin, the output format is NMOS push-pull.

I/O port	P51 output	0	1
	P51 input (initial setting)	0	0

(3) P50/FTxD

The pin function is switched as shown below according to the SCIFOE1 bit in SCIFCR o SCIF, the SCIFE bit in HICR5, and the P50DDR bit.

SCIFENABLE = 1: SCIFOE1 + SCIFE

		Setting	
		Logical Expression	**I/O Port**
Module Name	**Pin Function**	**SCIFENABLE**	**P50DDR**
SCIF	FTxD output	1	—
I/O port	P50 output	0	1
	P50 input (initial setting)	0	0

Module Name	Pin Function	I/O Port
		P67DDR
I/O port	P67 output	1
	P67 input (initial setting)	0

(2) P66/$\overline{\text{KIN6}}$/$\overline{\text{IRQ6}}$

When the KMIM6 bit in KMIMR of the interrupt controller is cleared to 0, this pin can the $\overline{\text{KIN6}}$ input pin. When the EIVS bit in SYSCR3 is cleared to 0 and the IRQ6E bit in interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ6}}$ input pin.

The pin function is switched as shown below according to the state of the P66DDR bit.

Module Name	Pin Function	Setting
		I/O Port
		P66DDR
I/O port	P66 output	1
	P66 input (initial setting)	0

P6n input (initial setting)	0

(r

7.2.7 Port 7

(1) P77/AN7, P76/AN6, P75/AN5, P74/AN4, P73/AN3, P72/AN2, P71/AN1, P70/AN

Module Name	Pin Function
A/D converter	ANn/P7n input

(r

Module Name	Pin Function	SCK1_OE	SCL1_OE	P86DDR
SCI	SCK1 input/output	1	0	—
IIC	SCL1 input/output	0	1	—
I/O port	P86 output	0	0	1
	P86 input (initial setting)	0	0	0

Note: To use this pin as SCL1 input/output, be sure that SCK1_OE is 0. To use this pin the IIC1AS and IIC1BS bits in PTCNT1 must be cleared to 0. The output format fo NMOS output only and direct bus drive is possible. When this pin is used as the P pin or SCK1 output pin, the output format is NMOS push-pull.

(2) P85/$\overline{\text{IRQ4}}$/RxD1

The pin function is switched as shown below according to the state of the P85DDR bit.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	RE	P85DDR
SCI	RxD1 input	1	—
I/O port	P85 output	0	1
	P85 input (initial setting)	0	0

	P84 input (initial setting)	0	0

(4) P83/$\overline{\text{LPCPD}}$

The pin function is switched as shown below according to the combination of the FSILIE SLCR of FSI, the SCIFE bit in HICR5 and the LPC4E bit in HICR4 of the LPC, LPC3E t bits in HICR0, and the P83DDR bit. LPCENABLE in the following table is expressed by following logical expression.

LPCENABLE = 1 : FSILIE + SCIFE + LPC4E + LPC3E + LPC2E + LPC1E

		Setting	
		Logical Expression	**I/O Port**
Module Name	**Pin Function**	**LPCENABLE**	**P83DDR**
LPC	$\overline{\text{LPCPD}}$ input	1	—
I/O port	P83 output	0	1
	P83 input (initial setting)	0	0

Name	Pin Function	LPCENABLE	P82DDR
LPC	$\overline{\text{CLKRUN}}$ output	1	—
I/O port	P82 output	0	1
	P82 input (initial setting)	0	0

(6) P81/GA20

The pin function is switched as shown below according to the combination of the registe of the LPC and the P81DDR bit.

		Setting	
		LPC	I/O Port
Module Name	Pin Function	GA20_OE	P81DDR
LPC	GA20 output	1	—
I/O port	P81 output	0	1
	P81 input (initial setting)	0	0

	P80 input (initial setting)	0	0

7.2.9 Port 9

(1) P97/$\overline{\text{IRQ15}}$/SDA0

The pin function is switched as shown below according to the combination of the IIC0AS IIC0BS bits in PTCNT1, ICE bit in ICCR of IIC_0, and the P97DDR bit. When the ISS1 ISSR16 is cleared to 0 and the IRQ15E bit in IER16 of the interrupt controller is set to 1, can be used as the $\overline{\text{IRQ15}}$ input pin.

		Setting	
		IIC_0	**I/O Port**
Module Name	**Pin Function**	**SDA0_OE**	**P97DDR**
IIC	SDA0 input/output	1	—
I/O port	P97 output	0	1
	P97 input (initial setting)	0	0

Note: To use this pin as SDA0, clear the IIC0AS and IIC0BS bits in PTCNT1 to 0. The ou format for SDA0 is NMOS output only and direct bus drive is possible. When this p as the P97 output pin, the output format is NMOS push-pull.

(3) P95/$\overline{\text{IRQ14}}$, P94/$\overline{\text{IRQ13}}$, P93/$\overline{\text{IRQ12}}$, P92/$\overline{\text{IRQ0}}$, P91/$\overline{\text{IRQ1}}$, P90/$\overline{\text{IRQ2}}$

The pin function is switched as shown below according to the state of the P9nDDR bit. V
ISSm bit in ISSR (ISSR16) is cleared to 0 and the IRQmE bit in IER (IER16) of the inte
controller is set to 1, this pin can be used as the $\overline{\text{IRQm}}$ input pin.

		Setting
		I/O Port
Module Name	**Pin Function**	**P9nDDR**
I/O port	P9n output	1
	P9n input (initial setting)	0

(m = 14 to

Module Name	Pin Function	PS2_OE	PAnDDR
PS2	PS2 input/output	1	—
I/O port	PAn output	0	1
	PAn input (initial setting)	0	0

(n = 7 to 0, m

Note: When the KBIOE bit is set to 1, this pin functions as an NMOS open-drain output, a bus drive is possible.

When the IICS bit in STCR is set to 1, the output format for PA7 to PA4 is NMOS o drain, and direct bus drive is possible.

FSI	FSISS output	1	—	—
SCIF	$\overline{\text{RTS}}$ output	0	1	—
I/O port	PB7 output	0	0	1
	PB7 input (initial setting)	0	0	0

(2) PB6/$\overline{\text{CTS}}$/FSICK

The pin function is switched as shown below according to the FSIE bit in FSICR1 of FS PB6DDR bit.

		Setting	
		FSI	I/O Port
Module Name	Pin Function	FSICK_OE	PB6DDR
FSI	FSICK output	1	—
I/O port	PB6 output	0	1
	PB6 input (initial setting)	0	0

I/O port	PB5 output	0	0	1
	PB5 input (initial setting)	0	0	0

(4) PB4/$\overline{\text{DSR}}$/FSIDO

The pin function is switched as shown below according to the state of the FSIE bit in FSI
FSI and the PB4DDR bit.

		Setting	
		FSI	I/O Port
Module Name	Pin Function	FSIDO_OE	PB4DDR
FSI	FSIDO output	1	—
I/O port	PB4 output	0	1
	PB4 input (initial setting)	0	0

	PB3 input (initial setting)	—	0

(6) PB2/$\overline{\text{RI}}$/PWMU0B

The pin function is switched as shown below according to the combination of the registe of the PWMU and the PB2DDR bit.

		Setting	
		PWMU	**I/O Port**
Module Name	**Pin Function**	**PWMU0B_OE**	**PB2DDR**
PWMU	PWMU0B output	1	1
I/O port	PB2 output	0	1
	PB2 input (initial setting)	—	0

	PB1 input (initial setting)	0	0

(8) PB0/$\overline{\text{LSMI}}$

The pin function is switched as shown below according to the combination of the register of the LPC and the PB0DDR bit.

		Setting	
		LPC	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{LSMI}}$_OE**	**PB0DDR**
LPC	$\overline{\text{LSMI}}$ output	1	—
I/O port	PB0 output	0	1
	PB0 input (initial setting)	0	0

counting mode and IOB3 in TIOR_2 is set to 1.

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCB2_OE	PC7DDR
TPU	TIOCB2 output	1	—
I/O port	PC7 output	0	1
	PC7 input (initial setting)	0	0

(2) PC6/$\overline{\text{WUE14}}$/TIOCA2

The pin function is switched as shown below according to the combination of the registe
of the TPU and the PC6DDR bit. When the WUEMR14 bit in WUEMR of the interrupt
is cleared to 0, this pin can be used as the $\overline{\text{WUE14}}$ input pin.

This pin functions as TIOCA2 input when TPU channel 2 timer operating mode is set to
operation or phase counting mode and IOA3 in TIOR_2 is set to 1.

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCA2_OE	PC6DDR
TPU	TIOCA2 output	1	—
I/O port	PC6 output	0	1
	PC6 input (initial setting)	0	0

Module Name	Pin Function	Setting	
		TPU	I/O Port
		TIOCB1_OE	PC5DDR
TPU	TIOCB1 output	1	—
I/O port	PC5 output	0	1
	PC5 input (initial setting)	0	0

(4) PC4/$\overline{\text{WUE12}}$/TIOCA1

The pin function is switched as shown below according to the combination of the register
of the TPU and the PC4DDR bit. When the WUEMR12 bit in WUEMR of the interrupt c
is cleared to 0, this pin can be used as the $\overline{\text{WUE12}}$ input pin.

This pin functions as TIOCA1 input when TPU channel 1 timer operating mode is set to n
operation or phase counting mode and IOA3 to IOA0 in TIOR_1 are set to B'10xx. (x: De

Module Name	Pin Function	Setting	
		TPU	I/O Port
		TIOCA1_OE	PC4DDR
TPU	TIOCA1 output	1	—
I/O port	PC4 output	0	1
	PC4 input (initial setting)	0	0

Module Name	Pin Function	Setting	
		TPU	I/O Port
		TIOCD0_OE	PC3DDR
TPU	TIOCD0 output	1	—
I/O port	PC3 output	0	1
	PC3 input (initial setting)	0	0

(6) PC2/$\overline{\text{WUE10}}$/TIOCC0/TCLKA

The pin function is switched as shown below according to the combination of the registe of the TPU and the PC2DDR bit. When the WUEMR10 bit in WUEMR of the interrupt is cleared to 0, this pin can be used as the $\overline{\text{WUE10}}$ input pin.

This pin functions as TCLKA input when TPSC2 to TPSC0 in any of TCR_0 to TCR_2 B'100 or when channel 1 is set to phase counting mode.

This pin functions as TIOCC0 input when TPU channel 0 timer operating mode is set to operation or phase counting mode and IOC3 to IOC0 in TIOR_0 are set to B'10xx. (x: D

Module Name	Pin Function	Setting	
		TPU	I/O Port
		TIOCC0_OE	PC2DDR
TPU	TIOCC0 output	1	—
I/O port	PC2 output	0	1
	PC2 input (initial setting)	0	0

Module Name	Pin Function	TPU	I/O Port
		TIOCB0_OE	PC1DDR
TPU	TIOCB0 output	1	—
I/O port	PC1 output	0	1
	PC1 input (initial setting)	0	0

(8) PC0/$\overline{\text{WUE8}}$/TIOCA0

The pin function is switched as shown below according to the combination of the register of the TPU and the PC0DDR bit. When the WUEMR8 bit in WUEMR of the interrupt co is cleared to 0, this pin can be used as the $\overline{\text{WUE8}}$ input pin.

This pin functions as TIOCA0 input when TPU channel 0 timer operating mode is set to r operation or phase counting mode and IOA3 to IOA0 in TIORH_0 are set to B'10xx. (x: D care.)

		Setting	
Module Name	Pin Function	TPU	I/O Port
		TIOCA0_OE	PC0DDR
TPU	TIOCA0 output	1	—
I/O port	PC0 output	0	1
	PC0 input (initial setting)	0	0

Name	Pin Function	PDnDDR
I/O port	PDn output	1
	PDn input (initial setting)	0

7.2.14 Port E

(1) PE4/ETMS, PE3/ETDO, PE2/ETDI, PE1/ETCK

The pin function is switched as shown below according to the operating mode.

		Setting	
		On-Chip Emulation Mode	**Single-Chip Mode**
Module Name	**Pin Function**	**Emulator Input/Output**	**PEn input**
Operating mode	On-chip emulation mode	1	—
	Single-chip mode	0	1

Note: These pins are not supported by the system development tool (emulator).

7.2.15 Port F

(1) PF7/PWMU5A, PF6/PWMU4A, PF5/PWMU3A, PF4/PWMU2A

The pin function is switched as shown below according to the combination of the register of the PWMU and the PFnDDR bit.

		Setting	
		PWMU	**I/O Port**
Module Name	**Pin Function**	**PWMUmA_OE**	**PFnDDR**
PWMU	PWMUmA output	1	1
I/O port	PFn output	0	1
	PFn input (initial setting)	—	0

(n = 5 to 2, m

Module Name	Pin Function	TMR	I/O Port
		TMOX_OE	**PF3DDR**
TMR	TMOX output	1	—
I/O port	PF3 output	0	1
	PF3 input (initial setting)	0	0

(3) PF2/TMOY/$\overline{\text{IRQ10}}$/TDPCYI0

The pin function is switched as shown below according to the combination of the registe of the TMR and the PF2DDR bit. When the TDPIPE bit in TDPIER_0 of TDP0 is set to can be used as the TDPCYI0 input pin. When the ISS10 bit in ISSR16 is cleared to 0 an IRQ10E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IR}}$ pin.

Module Name	Pin Function	Setting	
		TMR	**I/O Port**
		TMOY_OE	**PF2DDR**
TMR	TMOY output	1	—
I/O port	PF2 output	0	1
	PF2 input (initial setting)	0	0

I/O port	PF1 output	0	1
	PF1 input (initial setting)	—	0

(5) PF0/$\overline{\text{IRQ8}}$/PWMU0A

The pin function is switched as shown below according to the combination of the register of the PWMU and the PF1DDR bit. When the ISS8 bit in ISSR16 is cleared to 0 and the bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ8}}$ input p

		Setting	
		PWMU	I/O Port
Module Name	Pin Function	PWMU0A_OE	PF0DDR
PWMU	PWMU0A output	1	1
I/O port	PF0 output	0	1
	PF0 input (initial setting)	—	0

Name	Pin Function	ExSCLB_OE	PG7DDR
PTCNT1	ExSCLB input/output	1	—
I/O port	PG7 output	0	1
	PG7 input (initial setting)	0	0

Note: The output format for ExSCLB is NMOS output only, and direct bus drive is possi
this pin is used as the PG7 output pin, the output format is NMOS push-pull.

(2) PG6/ExSDAB/$\overline{\text{ExIRQ14}}$

The pin function is switched as shown below according to the combination of the registe
of PTCNT1 and the PG6DDR bit. When the ISS14 bit in ISSR16 is set to 1 and the IRQ
IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ14}}$ input pi

		Setting	
		PTCNT1	I/O Port
Module Name	Pin Function	ExSDAB_OE	PG6DDR
PTCNT1	ExSDAB input/output	1	—
I/O port	PG6 output	0	1
	PG6 input (initial setting)	0	0

Note: The output format for ExSDAB is NMOS output only, and direct bus drive is possi
this pin is used as the PG6 output pin, the output format is NMOS push-pull.

I/O port	PG5 output	0	1
	PG5 input (initial setting)	0	0

Note: The output format for ExSCLA is NMOS output only, and direct bus drive is possib this pin is used as the PG5 output pin, the output format is NMOS push-pull.

(4) PG4/ExSDAA/$\overline{\text{ExIRQ12}}$

The pin function is switched as shown below according to the combination of the register of PTCNT1 and the PG4DDR bit. When the ISS12 bit in ISSR16 is set to 1 and the IRQ1 IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ12}}$ input pin

		Setting	
		PTCNT1	I/O Port
Module Name	Pin Function	ExSDAA_OE	PG4DDR
PTCNT1	ExSDAA input/output	1	—
I/O port	PG4 output	0	1
	PG4 input (initial setting)	0	0

Note: The output format for ExSDAA is NMOS output only, and direct bus drive is possib this pin is used as the PG4 output pin, the output format is NMOS push-pull.

I/O port	PG3 output	0	1
	PG3 input (initial setting)	0	0

Note: The output format for SCL2 is NMOS output only, and direct bus drive is possible
this pin is used as the PG3 output pin, the output format is NMOS push-pull.

(6) PG2/SDA2/$\overline{\text{ExIRQ10}}$

The pin function is switched as shown below according to the combination of the registe of the IIC and the PG2DDR bit. When the ISS10 bit in ISSR16 is set to 1 and the IRQ1 IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ10}}$ input p

		Setting	
		IIC	**I/O Port**
Module Name	**Pin Function**	**SDA2_OE**	**PG2DDR**
IIC	SDA2 input/output	1	—
I/O port	PG2 output	0	1
	PG2 input (initial setting)	0	0

Note: The output format for SDA2 is NMOS output only, and direct bus drive is possible
this pin is used as the PG2 output pin, the output format is NMOS push-pull.

Name	Pin Function	PG1DDR
I/O port	PG1 output	1
	PG1 input (initial setting)	0

(8) PG0/$\overline{\text{ExIRQ8}}$/TMIX/TDPCYI1

The pin function is switched as shown below according to the state of the PG0DDR bit. V TDPIPE bit in TDPIER_1 of the TDP is set to 1, this pin is used as the TDPCYI1 input p the ISS8 bit in ISSR16 is set to 1 and the IRQ8E bit in IER16 of the interrupt controller i this pin can be used as the $\overline{\text{ExIRQ8}}$ input pin.

		Setting
		I/O Port
Module Name	Pin Function	PG0DDR
I/O port	PG0 output	1
	PG0 input (initial setting)	0

	PHn input (initial setting)	0

(2) PH2/CIRI

The pin function is switched as shown below according to the combination of the registe
of CIR and the PH2DDR bit.

		Setting	
		CIR	I/O Port
Module Name	Pin Function	CIRI	PH2DDR
CIR	CIRI input	1	—
I/O port	PH2 output	0	1
	PH2 input (initial setting)	0	0

Name	Pin Function	PH1DDR
I/O port	PH1 output	1
	PH1 input (initial setting)	0

(4) PH0/$\overline{\text{ExIRQ6}}$/TDPCYI2

The pin function is switched as shown below according to the state of the PH0DDR bit. V
TDPIPE bit in TDPIER_2 of the TDP is set to 1, this pin is used as the TDPCYI2 input p
the EIVS bit in SYSCR3 is set to 1 and the IRQ6E bit in IER of the interrupt controller is
this pin can be used as the $\overline{\text{ExIRQ6}}$ input pin.

		Setting
		I/O Port
Module Name	Pin Function	PH0DDR
I/O port	PH0 output	1
	PH0 input (initial setting)	0

	PIn input (initial setting)	0

Note: The output format for PIn is NMOS push-pull.

7.2.19 Port J

(1) PJ7, PJ6, PJ5, PJ4, PJ3, PJ2, PJ1, PJ0

The pin function is switched as shown below according to the state of the PJnDDR bit.

		Setting
		I/O Port
Module Name	Pin Function	PJnDDR
I/O port	PJn output	1
	PJn input (initial setting)	0

	2	P12_OE	P12	
	1	P11_OE	P11	
	0	P10_OE	P10	
P2	7	P27_OE	P27	
	6	P26_OE	P26	
	5	P25_OE	P25	
	4	P24_OE	P24	
	3	P23_OE	P23	
	2	P22_OE	P22	
	1	P21_OE	P21	
	0	P20_OE	P20	
P3	7	SERIRQ_OE	SERIRQ	FSI.SLCR.FSILIE, LPC.HICR5.SCIFE, HICR4.LPC4E HICR0.LPC[3E:1E] LPCENABLE = 1: FSILIE + SCIFE + LPC3E + LPC2E + LPC1E
	6	P36_OE	P36	
	5	P35_OE	P35	
	4	P34_OE	P34	
	3	LAD3_OE	LAD3	
	2	LAD2_OE	LAD2	
	1	LAD1_OE	LAD1	
	0	LAD0_OE	LAD0	

		PWMU2B_OE	PWMU2B		PWMU_B.PWMCONB.PWM2E = PWMU_B.PWMCOND.CNTMD23
	3	SCK2_OE	SCK2		SCI_2.SCR.CKE[1:0] = 01/10/11 = 1
	2	SDA1_OE	SDA1	PTCNT1.IIC1AS PTCNT1.IIC1BS	ICE•$\overline{\text{IIC1AS}}$•$\overline{\text{IIC1BS}}$ = 1
	1	TMO0_OE	TMO0		Except TMR_0.TCSR.OS[3:0] = 0
	0	TxD2_OE	TxD2		SCI_2.SCR.TE = 1
P5	2	SCL0_OE	SCL0	PTCNT1.IIC1AS PTCNT1.IIC1BS	ICE•$\overline{\text{IIC0AS}}$•$\overline{\text{IIC0BS}}$ = 1
	1	P51_OE	P51		
	0	FTxD_OE	FTxD		SCIF.SCIFCR.SCIFOE1, LPC.HIC SCIFENABLE = 1: SCIFOE1 + SC
P6	7	P67_OE	P67		
	6	P66_OE	P66		
	5	P65_OE	P65		
	4	P64_OE	P64		
	3	P63_OE	P63		
	2	P62_OE	P62		
	1	P61_OE	P61		
	0	P60_OE	P60		

	2	$\overline{\text{CLKRUN}}$_OE	$\overline{\text{CLKRUN}}$		FSI.SLCR.FSILIE, LPC.HICR5.SCIFE, HICR4.LPC4E, HICR0.LPC[3E:1E] LPCENABLE = 1: FSILIE + SCIFE + LPC3E + LPC2E + LPC1E
	1	GA20_OE	GA20		LPC.HICR0.FGA20E = 1
	0	$\overline{\text{PME}}$_OE	$\overline{\text{PME}}$		LPC.HICR0.PMEE = 1
P9	7	SDA0_OE	SDA	PTCNT1.IIC0AS PTCNT1.IIC0BS	ICE•$\overline{\text{IIC0AS}}$•$\overline{\text{IIC0BS}}$ = 1
	6	ϕ_OE	ϕ		
	5	P95_OE	P95		
	4	P94_OE	P94		
	3	P93_OE	P93		
	2	P92_OE	P92		
	1	P91_OE	P91		
	0	P90_OE	P90		
PA	7	PS2CD_OE	PS2CD		PS2_2.KBCRH.KBIOE = 1
	6	PS2CC_OE	PS2CC		PS2_2.KBCRH.KBIOE = 1
	5	PS2BD_OE	PS2BD		PS2_1.KBCRH.KBIOE = 1
	4	PS2BC_OE	PS2BC		PS2_1.KBCRH.KBIOE = 1
	3	PS2AD_OE	PS2AD		PS2_0.KBCRH.KBIOE = 1
	2	PS2AC_OE	PS2AC		PS2_0.KBCRH.KBIOE = 1
	1	PS2DD_OE	PS2DD		PS2_3.KBCRH.KBIOE = 1
	0	PS2DC_OE	PS2DC		PS2_3.KBCRH.KBIOE = 1

				SCIFOE=1: ($\overline{\text{SCIFE}}$ • SCIFOE1 • $\overline{\text{S}}$ SCIFE • $\overline{\text{SCIFOE0}}$)
	4	FSIDO_OE	FSIDO	FSI.FSICR1.FSIE = 1
	3	PWMU1B_OE	PWMU1B	PWMU_B.PWMCONB.PWM1E = 1
	2	PWMU0B_OE	PWMU0B	PWMU_B.PWMCONB.PWM0E = 1 PWMU_B.PWMCONC.CNTMD01 =
	1	LSCI_OE	LSCI	LPC.HICR0.LSCIE = 1
	0	$\overline{\text{LSMI}}$_OE	$\overline{\text{LSMI}}$	LPC.HICR0.LSMIE = 1
PC	7	TIOCB2_OE	TIOCB2	TPU.TIOR2.IOB3 = 0, TPU.TIOR2.IOB[1:0] = 01/10/11
	6	TIOCA2_OE	TIOCA2	TPU.TIOR2.IOA3 = 0, TPU.TIOR2.IOA[1:0] = 01/10/11
	5	TIOCB1_OE	TIOCB1	TPU.TIOR1.IOB3 = 0, TPU.TIOR1.IOB[1:0] = 01/10/11
	4	TIOCA1_OE	TIOCA1	TPU.TIOR1.IOA3 = 0, TPU.TIOR1.IOA[1:0] = 01/10/11
	3	TIOCD0_OE	TIOCD0	TPU.TIOR0.IOD3 = 0, TPU.TIOR0.IOD[1:0] = 01/10/11
	2	TIOCC0_OE	TIOCC0	TPU.TIOR0.IOC3 = 0, TPU.TIOR0.IOC[1:0] = 01/10/11
	1	TIOCB0_OE	TIOCB0	TPU.TIOR0.IOB3 = 0, TPU.TIOR0.IOB[1:0] = 01/10/11
	0	TIOCA0_OE	TIOCA0	TPU.TIOR0.IOA3 = 0, TPU.TIOR0.IOA[1:0] = 01/10/11

	0	PD0_OE	PD0	
PF	7	PWMU5A_OE	PWMU5A	PWMU_A.PWMCONB.PWM5E = 1
	6	PWMU4A_OE	PWMU4A	PWMU_A.PWMCONB.PWM4E = 1 PWMU_A.PWMCOND.CNTMD45 =
	5	PWMU3A_OE	PWMU3A	PWMU_A.PWMCONB.PWM3E = 1
	4	PWMU2A_OE	PWMU2A	PWMU_A.PWMCONB.PWM2E = 1 PWMU_A.PWMCOND.CNTMD23 =
	3	TMOX_OE	TMOX	Except TMR_X.TCSR.OS[3:0] = 00
	2	TMOY_OE	TMOY	Except TMR_Y.TCSR.OS[3:0] = 00
	1	PWMU1A_OE	PWMU1A	PWMU_A.PWMCONB.PWM1E = 1
	0	PWMU0A_OE	PWMU0A	PWMU_A.PWMCONB.PWM0E = 1 PWMU_A.PWMCONC.CNTMD01 =
PG	7	ExSCLB_OE	ExSCLB	PTCNT1.IIC1BS or PTCNT1.IIC0BS
	6	ExSDAB_OE	ExSDAB	PTCNT1.IIC1BS or PTCNT1.IIC0BS
	5	ExSCLA_OE	ExSCLA	PTCNT1.IIC1AS or PTCNT1.IIC0AS
	4	ExSDAA_OE	ExSDAA	PTCNT1.IIC1AS or PTCNT1.IIC0AS
	3	SCL2_OE	SCL2	IIC_2.ICCR.ICE = 1
	2	SDA2_OE	SDA2	IIC_2.ICCR.ICE = 1
	1	PG1_OE	PG1	
	0	PG0_OE	PG0	

	6	PI6_OE	PI6		
	5	PI5_OE	PI5		
	4	PI4_OE	PI4		
	3	PI3_OE	PI3		
	2	PI2_OE	PI2		
	1	PI1_OE	PI1		
	0	PI0_OE	PI0		
PJ	7	PJ7_OE	PJ7		
	6	PJ6_OE	PJ6		
	5	PJ5_OE	PJ5		
	4	PJ4_OE	PJ4		
	3	PJ3_OE	PJ3		
	2	PJ2_OE	PJ2		
	1	PJ1_OE	PJ1		
	0	PJ0_OE	PJ0		

- Port control register 0 (PTCNT0)
- Port control register 1 (PTCNT1)
- Port control register 2 (PTCNT2)

7.3.1 Port Control Register 0 (PTCNT0)

PTCNT0 selects ports that also function as the external sub-clock input pin.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	EXCLS	0	R/W	0: P96/EXCL is selected 1: PH0/ExEXCL is selected

				1	0:	Selects PG7/ExSCLB ar PG6/ExSDAB
				1	1:	Setting prohibited
4	—	0	R/W	Reserved		
5	—	0		The initial value should not be changed.		
3	IIC0BS	0	R/W	These bits select input/output pins for IIC_0		
2	IIC0AS	0	R/W	IIC0BS	IIC0AS	
				0	0:	Selects P52/SCL0 and F
				0	1:	Selects PG5/ExSCLA ar PG4/ExSDAA
				1	0:	Selects PG7/ExSCLB ar PG6/ExSDAB
				1	1:	Setting prohibited
1	—	0	R/W	Reserved		
0	—	0	R/W	The initial value should not be changed.		

Note: Do not set input/output of IIC_0 and IIC_1 for one pin at the same time.

5	RxD2RS	0	R/W	0: RxD2 direct input 1: RxD2 inverted input
4	TxD1RS	0	R/W	0: TxD1 direct output 1: TxD1 inverted output
3	RxD1RS	0	R/W	0: RxD1 direct input 1: RxD1 inverted input
2	—	0	R/W	Reserved The initial value should not be changed.
1	PORTS	0	R/W	0: Existing port specification 1: New port specification
0	—	0	R/W	Reserved The initial value should not be changed.

- Selectable from four types of counter input clock

  Selection of four internal clock signals (ϕ, ϕ/2, ϕ/4, and ϕ/8)
- Independent operation and variable cycle for each channel

  Cascaded connection of two channels is possible.

  Operation of channel 1 (higher order) and channel 0 (lower order) as a 16-bit single-PWM timer

  Operation of channel 3 (higher order) and channel 2 (lower order) as a 16-bit single-PWM timer

  Operation of channel 5 (higher order) and channel 4 (lower order) as a 16-bit single-PWM timer
- 8-bit single pulse mode

  Operates at a maximum carrier frequency of 78.1 kHz (at 20 MHz operation)

  Pulse output settable with a duty cycle from 0/255 to 255/255

  PWM output enable/disable control, and selection of direct or inverted PWM output
- 16-bit single pulse mode

  Two channels are cascade-connected for operation in this mode.

  Operates at a maximum carrier frequency of 305.1 Hz (at 20 MHz operation)

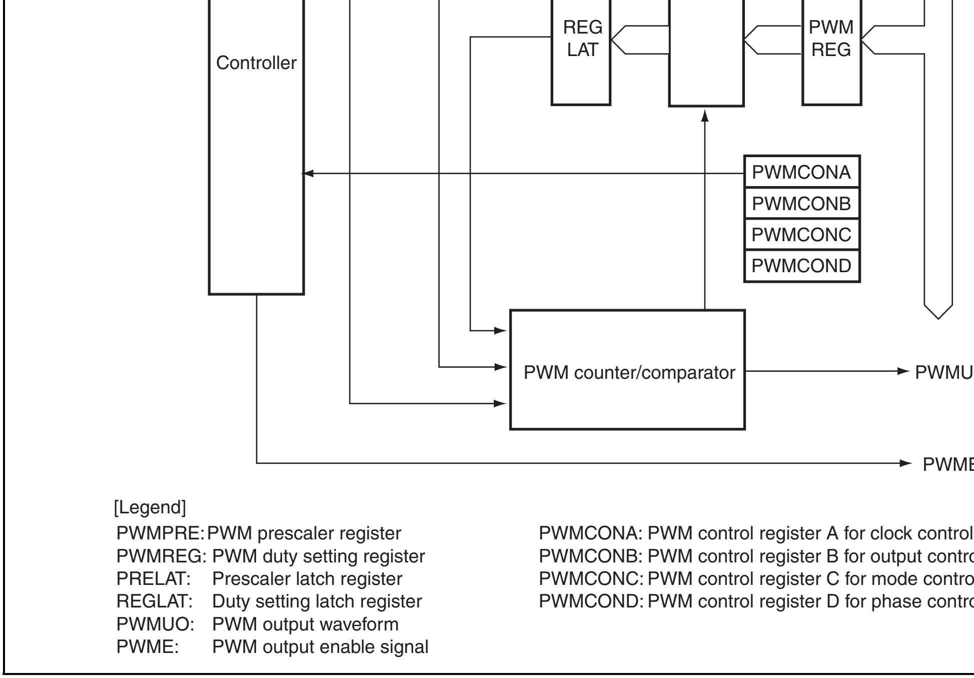
  Pulse output settable with a duty cycle from 0/65535 to 65535/65535

  PWM output enable/disable control, and selection of direct or inverted PWM output
- 8-bit pulse division mode

  Operable at a maximum carrier frequency of 1.25 MHz (at 20 MHz operation)

  Pulse output settable with a duty cycle from 0/16 to 15/16

  PWM output enable/disable control, and selection of direct or inverted PWM output



[Legend]

PWMPRE: PWM prescaler register
PWMREG: PWM duty setting register
PRELAT: Prescaler latch register
REGLAT: Duty setting latch register
PWMUO: PWM output waveform
PWME: PWM output enable signal

PWMCONA: PWM control register A for clock control
PWMCONB: PWM control register B for output contro
PWMCONC: PWM control register C for mode contro
PWMCOND: PWM control register D for phase contro

Figure 8.1 Block Diagram of PWMU Timer

				division)
	2	PWMU2A	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	3	PWMU3A	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit division)
	4	PWMU4A	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	5	PWMU5A	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit division)
Channel B	0	PWMU0B	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	1	PWMU1B	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit division)
	2	PWMU2B	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	3	PWMU3B	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit division)
	4	PWMU4B	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	5	PWMU5B	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit division)

	PWM control register B_A (for output control)	PWMCONB_A	R/W	H'00	H'FD0D
	PWM control register C_A (for mode control)	PWMCONC_A	R/W	H'00	H'FD0E
	PWM control register D_A (for phase control)	PWMCOND_A	R/W	H'00	H'FD0F
	PWM prescaler register 0_A	PWMPRE0_A	R/W	H'00	H'FD01
	PWM prescaler register 1_A	PWMPRE1_A	R/W	H'00	H'FD03
	PWM prescaler register 2_A	PWMPRE2_A	R/W	H'00	H'FD05
	PWM prescaler register 3_A	PWMPRE3_A	R/W	H'00	H'FD07
	PWM prescaler register 4_A	PWMPRE4_A	R/W	H'00	H'FD09
	PWM prescaler register 5_A	PWMPRE5_A	R/W	H'00	H'FD0B
	PWM duty setting register 0_A	PWMREG0_A	R/W	H'00	H'FD00
	PWM duty setting register 1_A	PWMREG1_A	R/W	H'00	H'FD02
	PWM duty setting register 2_A	PWMREG2_A	R/W	H'00	H'FD04
	PWM duty setting register 3_A	PWMREG3_A	R/W	H'00	H'FD06
	PWM duty setting register 4_A	PWMREG4_A	R/W	H'00	H'FD08
	PWM duty setting register 5_A	PWMREG5_A	R/W	H'00	H'FD0A

	PWM prescaler register 0_B	PWMPRE0_B	R/W	H'00	H'FD11
	PWM prescaler register 1_B	PWMPRE1_B	R/W	H'00	H'FD13
	PWM prescaler register 2_B	PWMPRE2_B	R/W	H'00	H'FD15
	PWM prescaler register 3_B	PWMPRE3_B	R/W	H'00	H'FD17
	PWM prescaler register 4_B	PWMPRE4_B	R/W	H'00	H'FD19
	PWM prescaler register 5_B	PWMPRE5_B	R/W	H'00	H'FD1B
	PWM duty setting register 0_B	PWMREG0_B	R/W	H'00	H'FD10
	PWM duty setting register 1_B	PWMREG1_B	R/W	H'00	H'FD12
	PWM duty setting register 2_B	PWMREG2_B	R/W	H'00	H'FD14
	PWM duty setting register 3_B	PWMREG3_B	R/W	H'00	H'FD16
	PWM duty setting register 4_B	PWMREG4_B	R/W	H'00	H'FD18
	PWM duty setting register 5_B	PWMREG5_B	R/W	H'00	H'FD1A

				0 1: Internal clock φ/2 is selected
				1 0: Internal clock φ/4 is selected
				1 1: Internal clock φ/8 is selected
5 to 0	–	All 0	R	Reserved These bits are always read as 0 and canno modified.

8.3.2 PWM Control Register B (PWMCONB)

PWMCONB controls enabling and disabling of the PWM output and counter operation o channel.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	PWM5E	0	R/W	PWMU5 Output Enable 0: PWMU5 output and counter operation a disabled. 1: PWMU5 output and counter operation a enabled.

				disabled. 1: PWMU4 output and counter operation enabled.
3	PWM3E	0	R/W	PWMU3 Output Enable 0: PWMU3 output and counter operation disabled. 1: PWMU3 output and counter operation enabled.
2	PWM2E	0	R/W	PWMU2 Output Enable • 8-bit single-pulse/pulse division mode 0: PWMU2 output and counter operation disabled. 1: PWMU2 output and counter operation enabled. • 16-bit single-pulse mode 0: PWMU2 output and counter operation disabled. 1: PWMU2 output and counter operation enabled.
1	PWM1E	0	R/W	PWMU1 Output Enable 0: PWMU1 output and counter operation disabled. 1: PWMU1 output and counter operation enabled.

disabled.

1: PWMU0 output and counter operation a enabled.

				1: Channels 0 and 1 are in 16-bit counter c (Upper: channel 1, lower: channel 0). Note: When the 16-bit counter is selected, s single pulse mode.
5	PWMSL5	0	R/W	Channel 5 Operating Mode Select 0: Single-pulse mode 1: Pulse division mode (Specify 8-bit count
4	PWMSL4	0	R/W	Channel 4 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit count
3	PWMSL3	0	R/W	Channel 3 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit count
2	PWMSL2	0	R/W	Channel 2 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit count
1	PWMSL1	0	R/W	Channel 1 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit count
0	PWMSL0	0	R/W	Channel 0 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit count

				0: PWMU4 direct output 1: PWMU4 inverted output
5	PH3S	0	R/W	Channel 3 Output Phase Select 0: PWMU3 direct output 1: PWMU3 inverted output
4	PH2S	0	R/W	Channel 2 Output Phase Select 0: PWMU2 direct output 1: PWMU2 inverted output
3	PH1S	0	R/W	Channel 1 Output Phase Select 0: PWMU1 direct output 1: PWMU1 inverted output
2	PH0S	0	R/W	Channel 0 Output Phase Select 0: PWMU0 direct output 1: PWMU0 inverted output
1	CNTMD45	0	R/W	Channels 4 and 5 Counter Select 0: Channels 4 and 5 are in 8-bit counter op 1: Channels 4 and 5 are in 16-bit counter o (Upper: channel 5, lower: channel 4). Note: When the 16-bit counter is selected, s single pulse mode.
0	CNTMD23	0	R/W	Channels 2 and 3 Counter Select 0: Channels 2 and 3 are in 8-bit counter op 1: Channels 2 and 3 are in 16-bit counter o (Upper: channel 2, lower: channel 3). Note: When the 16-bit counter is selected, s single pulse mode.

Operation) when φ = 20 MHz

				Carrier Frequency	
Internal Clock Frequency	Resolution	PWM Conversion Period		Single Pulse Mode	
		Min.	Max.	Min.	Max
φ	50 ns	12.8 µs	3.3 ms	306.4 Hz	78.4
φ/2	100 ns	25.5 µs	6.5 ms	153.2 Hz	39.2
φ/4	200 ns	51.2 µs	13.1 ms	76.6 Hz	19.6
φ/8	400 ns	102 µs	26.1 ms	38.3 Hz	9.8

Internal Clock Frequency	Resolution	Min.	Max.	Min.	Max.
ϕ	50 ns	3.3 ms	838.9 ms	1.2 Hz	305.1
ϕ/2	100 ns	6.5 ms	1.7 s	0.6 Hz	152.6
ϕ/4	200 ns	13.1 ms	3.4 s	0.3 Hz	76.3
ϕ/8	400 ns	26.2 ms	6.7 s	0.15 Hz	38.1

(3) 8-Bit Pulse Division Mode

PWM cycle = [16 × (n + 1)] / internal clock frequency (0 ≤ n ≤ 255)

PWM conversion cycle = [256 × (n + 1)] / internal clock frequency (0 ≤ n ≤ 255)

Table 8.5 Resolution, PWM Conversion Period, and Carrier Frequency when ϕ = (at 8-bit counter operation)

Internal Clock		PWM Conversion Period		Carrier Frequency (1/PWM	
Frequency	Resolution	Min.	Max.	Min.	Max.
ϕ	50 ns	12.8 µs	3.3ms	4882.8Hz	1250.0 k
ϕ/2	100 ns	25.6 µs	6.6ms	2441.4Hz	625.0 kH
ϕ/4	200 ns	51.2 µs	13.1ms	1220.7Hz	312.5 kH
ϕ/8	400 ns	102.4 µs	26.2ms	610.4Hz	156.3 kH

When the PWMREG value is m, the high period of the output pulse is calculated as foll

$$\text{Output pulse high period} = (\text{PWM cycle} \times m) / 255 \quad (0 \leq m \leq 255)$$

(2) 16-Bit Single Pulse Mode

Directly set the high period of the pulse for PWM output. With cascade-connected PWM registers, the duty cycle of the PWM output pulse is specified as a value from 0/65535 t 65535/65535.

When the PWMREG value is m, the high period of the output pulse is calculated as foll

$$\text{Output pulse high period} = (\text{PWM cycle} \times m) / 65535 \quad (0 \leq m \leq 65535)$$

Set the respective high-level pulse periods by using the following register combinations connection): PWMREG1 (higher order) and PWMREG0 (lower order), PWMREG3 (hi and PWMREG2 (lower order), and PWMREG5 (higher order) and PWMREG4 (lower o

(3) 8-Bit Pulse Division Mode

Specify the basic pulse duty cycle and the number of additional pulses for PWM output. higher-order four bits of the PWMREG setting specify the duty cycle of the basic pulse 15/16 with a resolution of 1/16, and the lower-order four bits specify the number of puls added within the conversion period comprising the basic pulses.

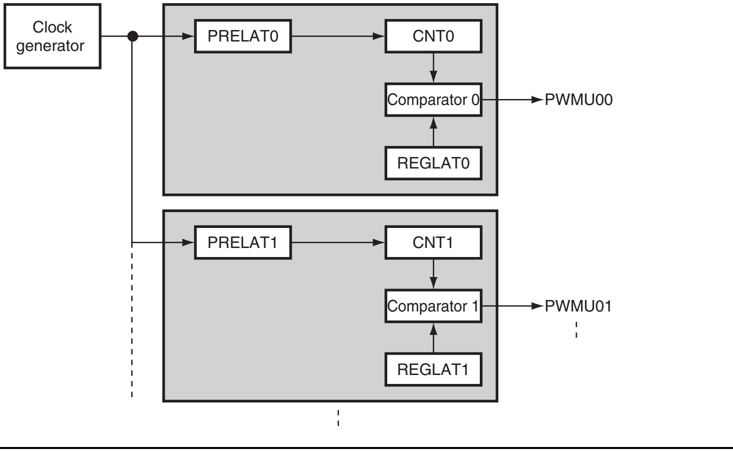


Figure 8.2 Block Diagram of 8-Bit Single Pulse Mode

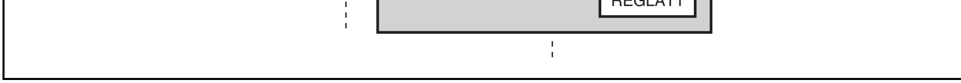


Figure 8.3 Block Diagram of 16-bit Single Pulse Mode

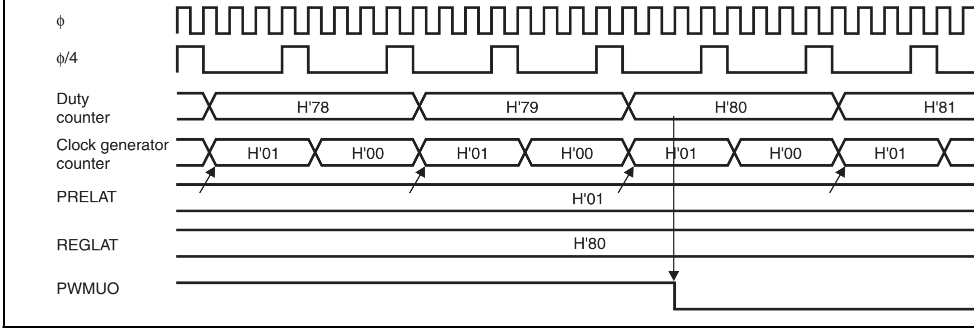


Figure 8.4 Example of Duty Counter and Clock Generator Counter Operati
(When PWMPRE = H'01 and PWMREG = H'80 with φ/4 Selected as Count Clock

The following shows the duty counter value and PWMU output timing.

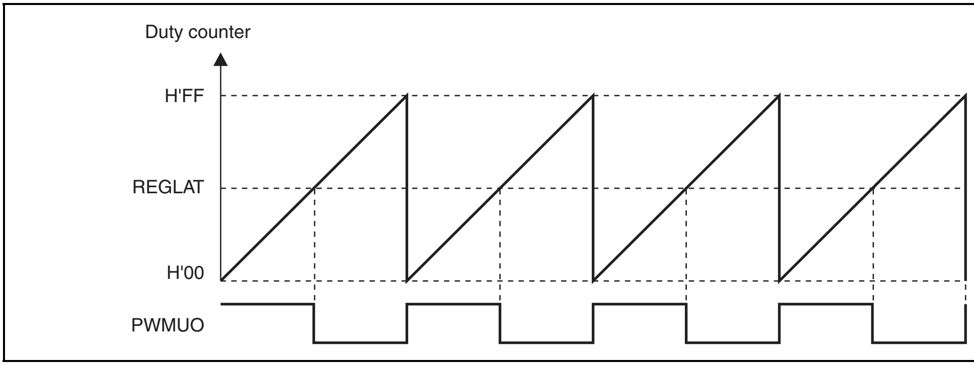


Figure 8.5 Duty Counter Value and PWMU Output Timing

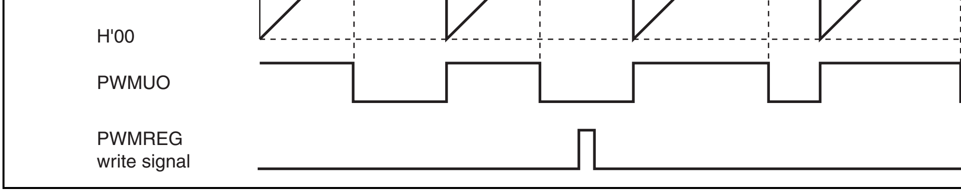


Figure 8.6 PWMU Output Waveform When PWMREG Value is Change

When the PWMPRE value is changed during PWM output, the PWM cycle changes fro
cycle. When the clock generator counter underflows, the PWMPRE value is loaded into
The following shows the PRELAT update timing when the PWMPRE value is changed.

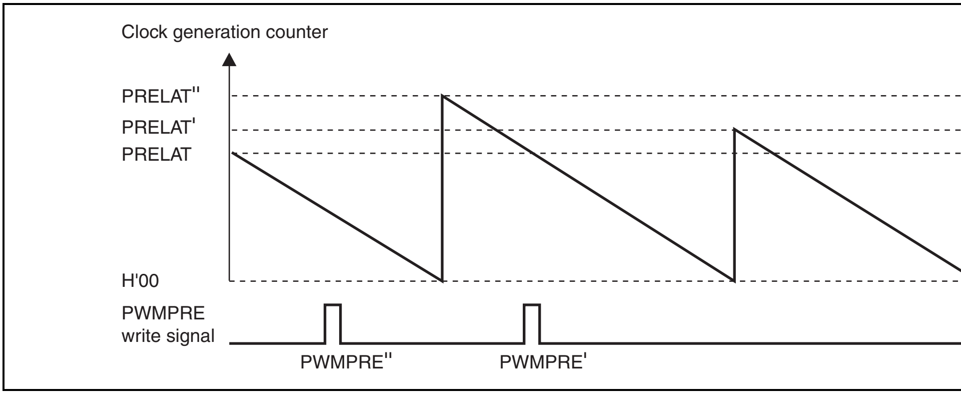
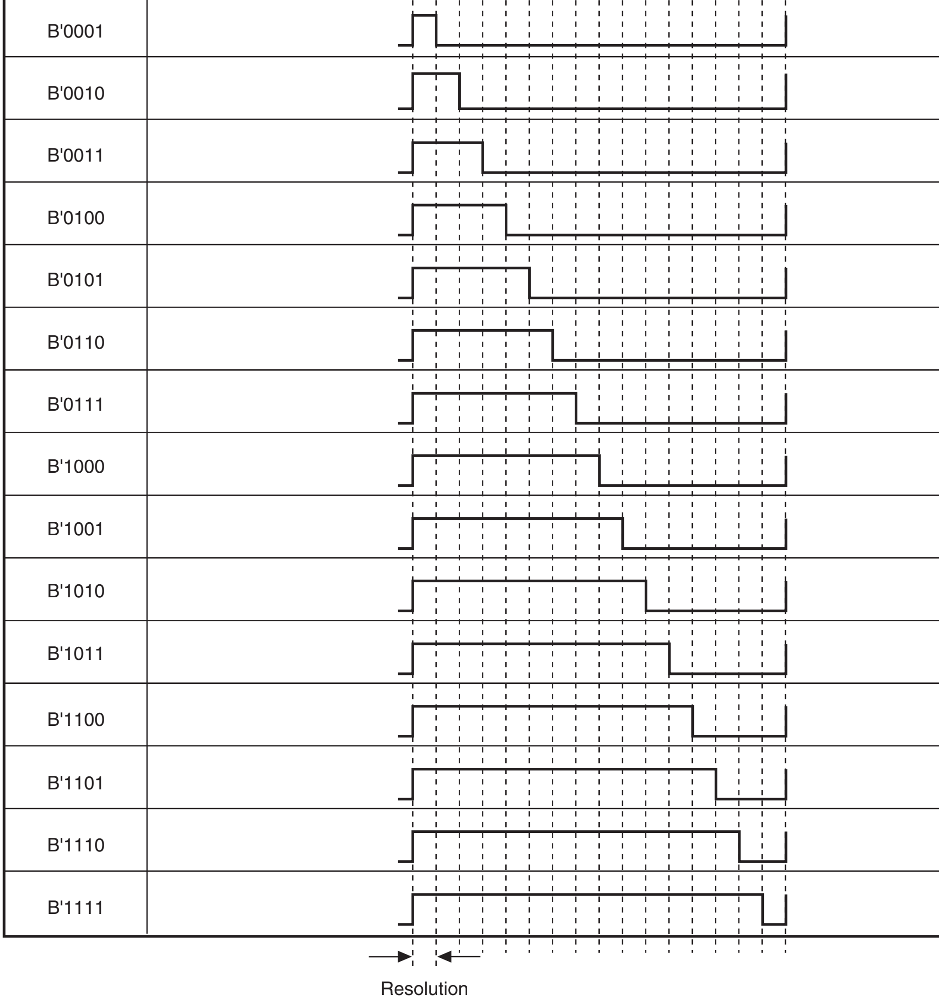


Figure 8.7 PRELAT Update Timing When PWMPRE Value is Changed


B'0001
B'0010
B'0011
B'0100
B'0101
B'0110
B'0111
B'1000
B'1001
B'1010
B'1011
B'1100
B'1101
B'1110
B'1111
Resolution

B'0000													
B'0001													
B'0010							O						
B'0011							O				O		
B'0100			O				O				O		
B'0101			O				O				O		O
B'0110			O		O		O				O		O
B'0111			O		O		O		O		O		O
B'1000	O		O		O		O		O		O		O
B'1001	O		O		O		O		O		O		O
B'1010	O		O		O	O	O		O		O		O
B'1011	O		O		O	O	O		O	O	O		O
B'1100	O	O	O		O	O	O		O	O	O		O
B'1101	O	O	O		O	O	O		O	O	O	O	O
B'1110	O	O	O	O	O	O	O		O	O	O	O	O
B'1111	O	O	O	O	O	O	O	O	O	O	O	O	O

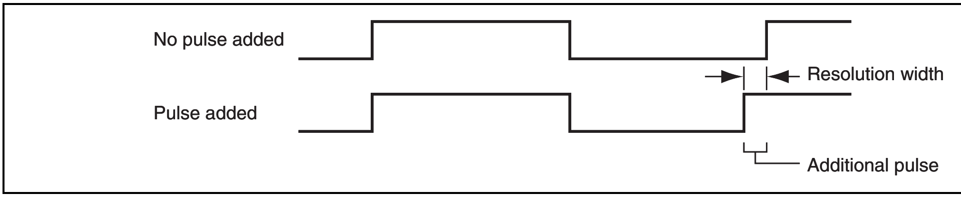
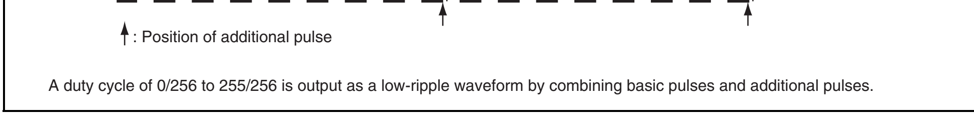


Figure 8.8 Example of Additional Pulse Timing (Upper 4 Bits in PWMREG = I



A duty cycle of 0/256 to 255/256 is output as a low-ripple waveform by combining basic pulses and additional pulses.

Figure 8.9 Example of WMU Setting

(2) Example of Circuit for Use as D/A Converter

The following shows an example of a circuit in which PWMU output pulses are used as a converter. When a low-pass filter is connected externally to the LSI, low-ripple analog ou be generated. If pulse division mode is used, a D/A output with even less ripple is availab

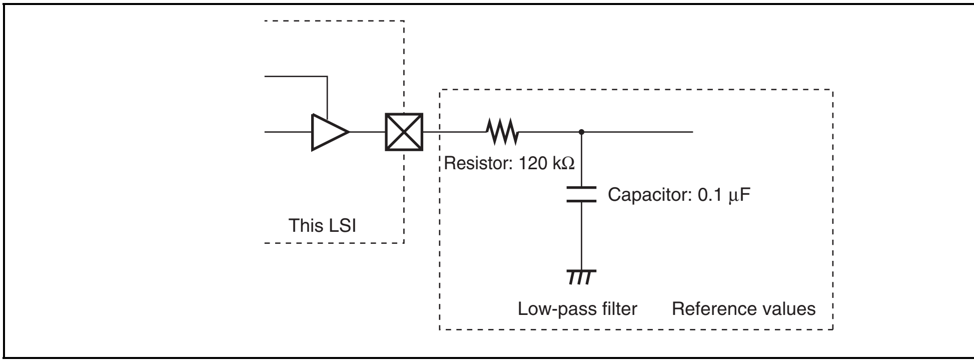


Figure 8.10 Example of Circuit for Use as a D/A Converter

When the duty cycle is to be changed in usage of a 16-bit single-pulse PWM timer, the h lower-order eight bits must be individually written to the respective PWMREGn (n = 0 t registers. There will thus be a time lag between the write operations, and this may lead t output of a pulse waveform with a duty cycle other than the intended one during the corr period.

Also, care must be taken to ensure that there are no interrupts while writing to PWMREG progress, since interrupt processing can lead to the continued output of pulses with a dut other than the intended one.

- Two base cycle settings

  The base cycle can be set equal to T × 64 or T × 256, where T is the resolution.
- Sixteen operation clocks (by combination of eight resolution settings and two base c settings)

Figure 9.1 shows a block diagram of the PWM (D/A) module.

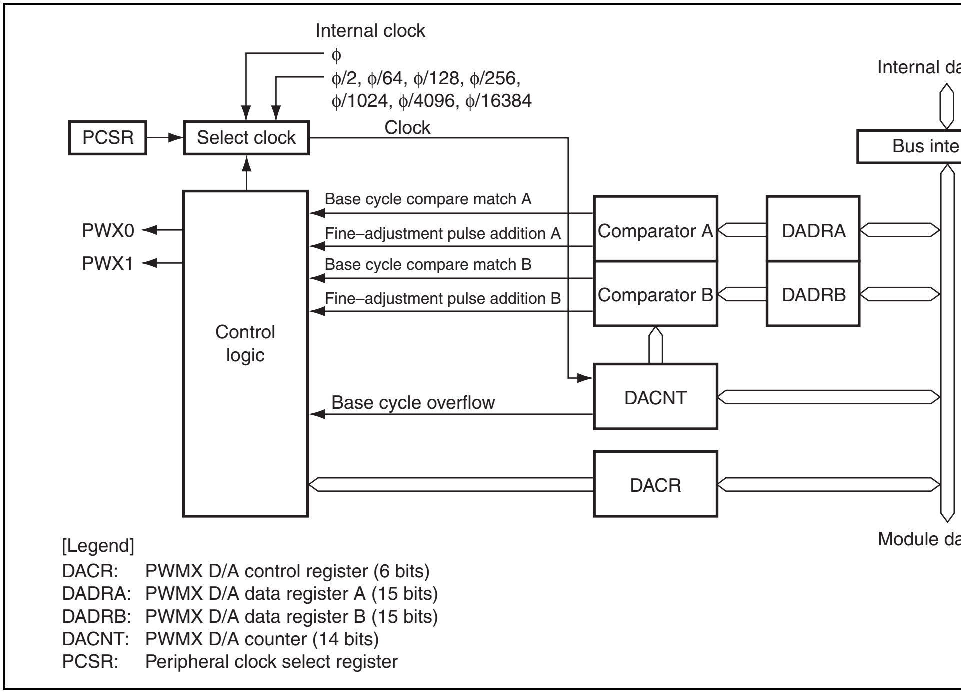

Internal clock
ϕ
ϕ/2, ϕ/64, ϕ/128, ϕ/256,
ϕ/1024, ϕ/4096, ϕ/16384
Internal da
PCSR
Select clock
Clock
Bus inte
Base cycle compare match A
Fine–adjustment pulse addition A
Base cycle compare match B
Fine–adjustment pulse addition B
PWX0
PWX1
Comparator A
Comparator B
DADRA
DADRB
Control logic
Base cycle overflow
DACNT
DACR
Module da
[Legend]
DACR: PWMX D/A control register (6 bits)
DADRA: PWMX D/A data register A (15 bits)
DADRB: PWMX D/A data register B (15 bits)
DACNT: PWMX D/A counter (14 bits)
PCSR: Peripheral clock select register

Figure 9.1 PWMX (D/A) Block Diagram

9.3 Register Descriptions

The PWMX (D/A) module has the following registers. The PWMX (D/A) registers are as
the same addresses with other registers. The registers are selected by the IICE bit in the se
timer control register (STCR). For details on the module stop control register, see section
Module Stop Control Registers H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA,
MSTPCRB).

Table 9.2 Register Configuration

					Da
Register Name	**Abbreviation**	**R/W**	**Initial Value**	**Address**	**Wi**
PWMX (D/A) counter H	DACNTH	R/W	H'00	H'FFA6 H'FEA6*	8
PWMX (D/A) counter L	DACNTL	R/W	H'03	H'FFA7 H'FEA7*	8
PWMX (D/A) data register AH	DADRAH	R/W	H'FF	H'FFA0 H'FEA0*	8
PWMX (D/A) data register AL	DADRAL	R/W	H'FF	H'FFA1 H'FEA1*	8
PWMX (D/A) data register BH	DADRBH	R/W	H'FF	H'FFA6 H'FEA6*	8
PWMX (D/A) data register BL	DADRBL	R/W	H'FF	H'FFA7 H'FEA7*	8
PWMX (D/A) control register	DACR	R/W	H'30	H'FFA0 H'FEA0*	8
Peripheral clock select register	PCSR	R/W	H'00	H'FF82	8

Notes: The same addresses are shared by DADRA and DACR, and by DADRB and DAC
Switching is performed by the REGS bit in DACNT or DADRB.

* Upper address: when RELOCATE = 0
Lower address: when RELOCATE = 1

Bit (CPU):	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Bit (counter):	7	6	5	4	3	2	1	0	8	9	10	11	12	13

- DACNTH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DACNT7 to DACNT0	All 0	R/W	Upper Up-Counter

- DACNTL

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	DACNT 8 to DACNT 13	All 0	R/W	Lower Up-Counter
1	—	1	R	Reserved Always read as 1 and cannot be modified.
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT located at the same addresses. The REGS bi which registers can be accessed. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

14	DA12	1	R/W	These bits set a digital value to be converted to analog value. In each base cycle, the DACNT value is contin compared with the DADR value to determine th cycle of the output waveform, and to decide wh output a fine-adjustment pulse equal in width to resolution. To enable this operation, this regist be set within a range that depends on the CFS the DADR value is outside this range, the PWM is held constant. A channel can be operated with 12-bit precisio fixing DA0 and DA1 to 0. The two data bits are compared with DACNT12 and DACNT13 of DA
13	DA11	1	R/W	
12	DA10	1	R/W	
11	DA9	1	R/W	
10	DA8	1	R/W	
9	DA7	1	R/W	
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 The range of DA13 to DA0: H'0100 to H'3FF 1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FF
0	—	1	R	Reserved Always read as 1 and cannot be modified.

9	DA7	1	R/W	resolution. To enable this operation, this regis be set within a range that depends on the CF the DADR value is outside this range, the PW is held constant. A channel can be operated with 12-bit precisi fixing DA0 and DA1 to 0. The two data bits ar compared with DACNT12 and DACNT13 of D
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 DA13 to DA0 range = H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 DA13 to DA0 range = H'0040 to H'3FFF
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT located at the same addresses. The REGS bi which registers can be accessed. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

				0: DACNT operates as a 14-bit up-counter 1: DACNT halts at H′0003
5	—	1	R	Reserved
4	—	1	R	Always read as 1 and cannot be modified.
3	OEB	0	R/W	Output Enable B Enables or disables output on PWMX (D/A) ch 0: PWMX (D/A) channel B output (at the PWX1 pin) is disabled 1: PWMX (D/A) channel B output (at the PWX1 pin) is enabled
2	OEA	0	R/W	Output Enable A Enables or disables output on PWMX (D/A) ch 0: PWMX (D/A) channel A output (at the PWX0 pin) is disabled 1: PWMX (D/A) channel A output (at the PWX0 pin) is enabled
1	OS	0	R/W	Output Select Selects the phase of the PWMX (D/A) output. 0: Direct PWMX (D/A) output 1: Inverted PWMX (D/A) output
0	CKS	0	R/W	Clock Select Selects the PWMX (D/A) resolution. Eight kind resolution can be selected. 0: Operates at resolution (T) = system clock cy (t_{cyc}) 1: Operates at resolution (T) = system clock cy (t_{cyc}) × 2, × 64, × 128, × 256, × 1024, × 4096, 16384.

				DACR of PWMX being 1. See table 9.3.
3 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	PWCKXC	0	R/W	PWMX clock select This bit selects a clock cycle with the CKS bit of PWMX being 1. See table 9.3.

Table 9.3 Clock Select of PWMX

PWCKXC	PWCKXB	PWCKXA	Resolution (T)
0	0	0	Operates on the system clock cycle (t_{cyc})
0	0	1	Operates on the system clock cycle (t_{cyc})
0	1	0	Operates on the system clock cycle (t_{cyc})
0	1	1	Operates on the system clock cycle (t_{cyc})
1	0	0	Operates on the system clock cycle (t_{cyc})
1	0	1	Operates on the system clock cycle (t_{cyc})
1	1	0	Operates on the system clock cycle (t_{cyc})
1	1	1	Setting prohibited

combined 16-bit value is written in the register.

- Read

  When the upper byte is read from, the upper-byte value is transferred to the CPU and lower-byte value is transferred to TEMP. Next, when the lower byte is read from, the byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time with a MOV instruction, and byte should always be accessed before the lower byte. Correct data will not be transferred the upper byte or only the lower byte is accessed. Also note that a bit manipulation instru cannot be used to access these registers.

Example 1: Write to DACNT

```
MOV.W R0, @DACNT      ; Write R0 contents to DACNT
```

Example 2: Read DADRA

```
MOV.W @DADRA, R0      ; Copy contents of DADRA to R0
```

×: The result of the access in the unit cannot be guaranteed.

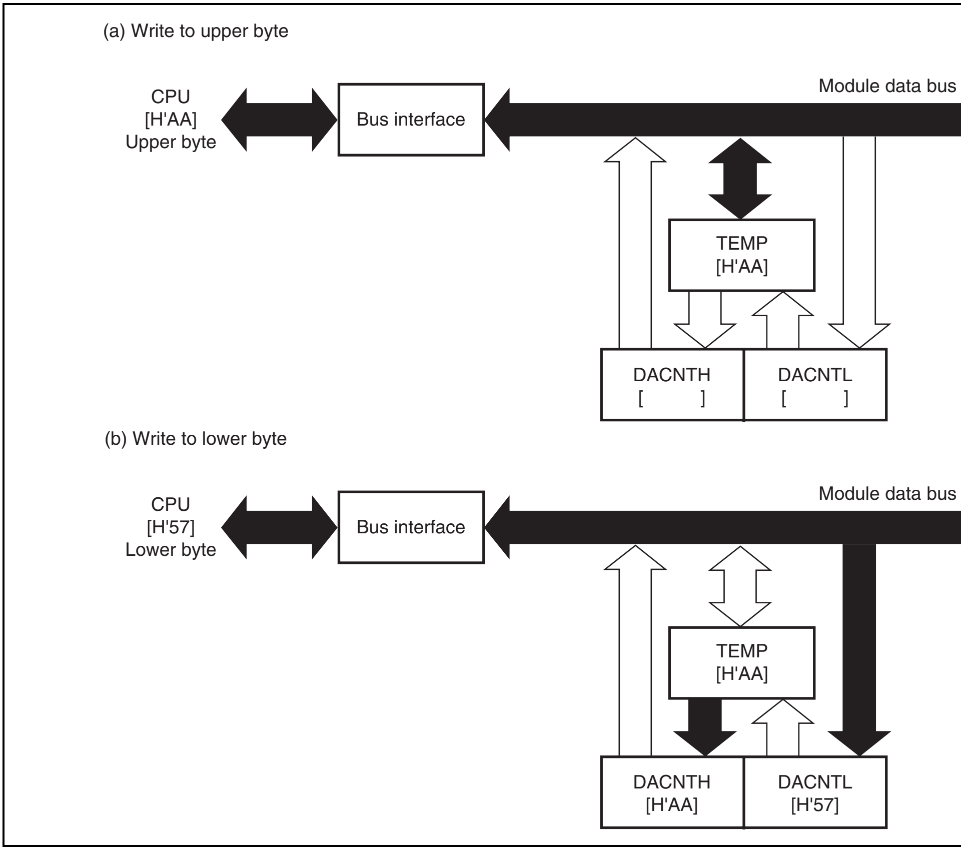


Figure 9.2 DACNT Access Operation (1) [CPU → DACNT (H'AA57) Writi

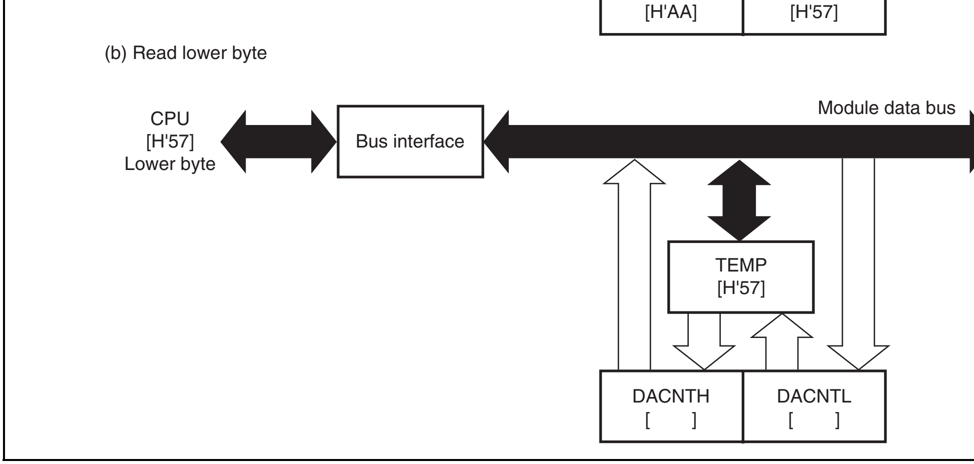


Figure 9.2 DACNT Access Operation (2) [DACNT → CPU (H'AA57) Readin

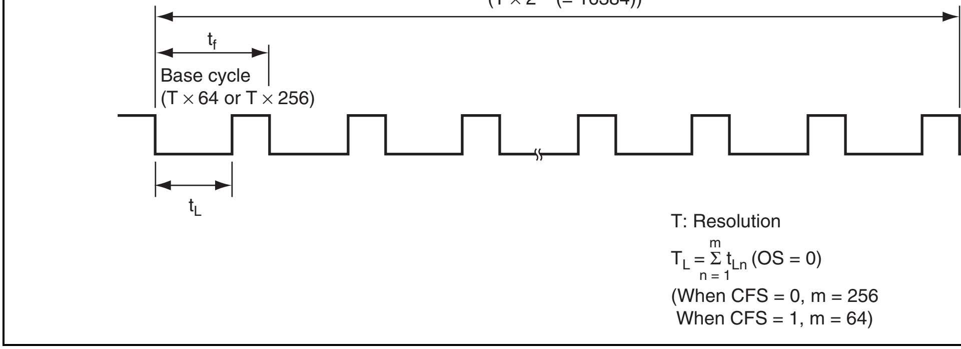


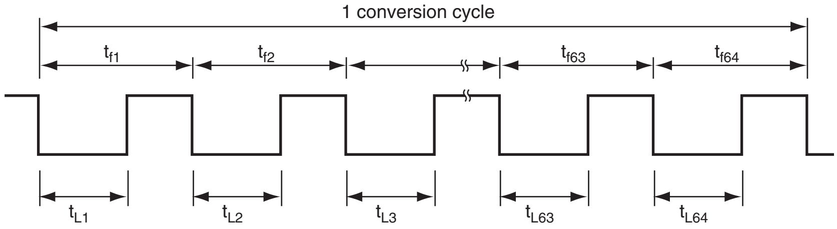
Figure 9.3 PWMX (D/A) Operation

Table 9.5 summarizes the relationships between the CKS and CFS bit settings and the re base cycle, and conversion cycle. The PWM output remains fixed unless DA13 to DA0 contain at least a certain minimum value. The relationship between the OS bit and the ou waveform is shown in figures 9.4 and 9.5.

						(µs)		DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	2
				(ϕ)		/78.1kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	5
0	0	0	1	0.1	0	6.4	1.64	Always low/high output	14					1
						(µs)	(ms)	DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0	4
						/156.2kHz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	1
					1	25.6		Always low/high output	14					1
						(µs)		DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	4
				(ϕ/2)		/39.1kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	1
0	0	1	1	3.2	0	204.8	52.4	Always low/high output	14					5
						(µs)	(ms)	DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0	1
						/4.9kHz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	3
					1	819.2		Always low/high output	14					5
						(µs)		DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	1
				(ϕ/64)		/1.2kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	3
0	1	0	1	6.4	0	409.6	104.9	Always low/high output	14					1
						(µs)	(ms)	DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0	2
						/2.4kHz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	6
					1	1638.4		Always low/high output	14					1
						(µs)		DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	2
				(ϕ/128)		/610.4kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	6

				(ϕ/256)		/305.2kHz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	0	1	51.2	0	3.3	838.9	Always low/high output	14				
						(ms)	(ms)	DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0
						/305.2Hz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	13.1		Always low/high output	14				
						(ms)		DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0
				(ϕ/1024)		/76.3Hz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	1	1	204.8	0	13.1	3.4	Always low/high output	14				
						(ms)	(s)	DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0
						/76.3Hz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	52.4		Always low/high output	14				
						(ms)		DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0
				(ϕ/4096)		/19.1Hz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	1	0	1	819.2	0	52.4	13.4	Always low/high output	14				
						(ms)	(s)	DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0
						/19.1Hz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	209.7		Always low/high output	14				
						(ms)		DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0
				(ϕ/16384)		/4.8Hz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	1	1	1	Setting prohibited	—	—	—	—	—	—	—	—	—

Note: * Indicates the conversion cycle when specific DA3 to DA0 bits are fixed.

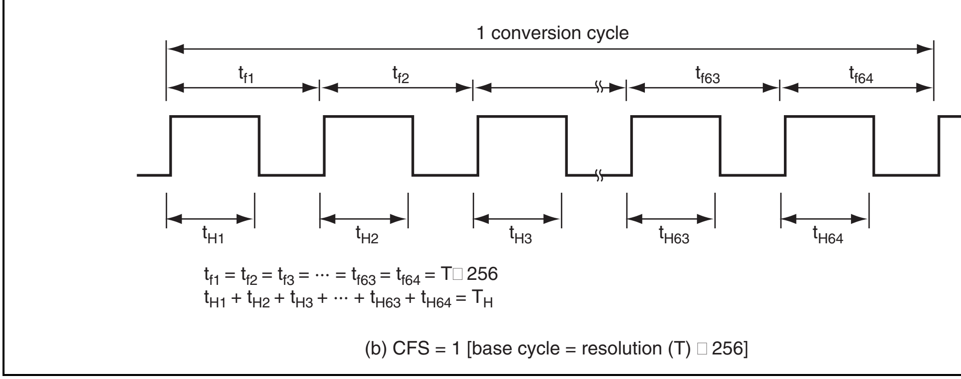


$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T$□256

$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L63} + t_{L64} = T_L$

(b) CFS = 1 [base cycle = resolution (T) □ 256]

Figure 9.4 Output Waveform (OS = 0, DADR corresponds to T_L)

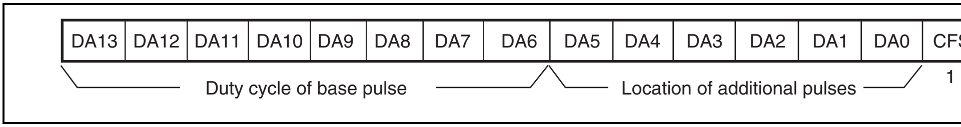

1 conversion cycle
t_{f1}
t_{f2}
t_{f63}
t_{f64}
t_{H1}
t_{H2}
t_{H3}
t_{H63}
t_{H64}
$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T$□256
$t_{H1} + t_{H2} + t_{H3} + \cdots + t_{H63} + t_{H64} = T_H$

(b) CFS = 1 [base cycle = resolution (T) □ 256]

Figure 9.5 Output Waveform (OS = 1, DADR corresponds to T_H)

An example of the additional pulses when CFS = 1 (base cycle = resolution (T) × 256) a (inverted PWM output) is described below. When CFS = 1, the upper eight bits (DA13 t DADR determine the duty cycle of the base pulse while the subsequent six bits (DA5 to determine the locations of the additional pulses as shown in figure 9.6.

Table 9.6 lists the locations of the additional pulses.



DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	CF
														1

Figure 9.6 D/A Data Register Configuration when CFS = 1

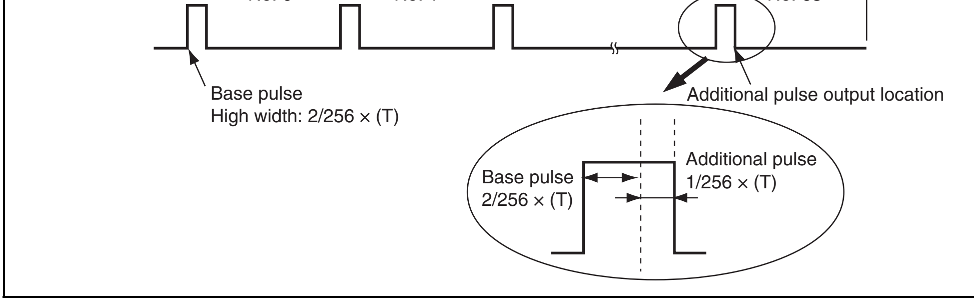


Figure 9.7 Output Waveform when DADR = H'0207 (OS = 1)

However, when CFS = 0 (base cycle = resolution (T) × 64), the duty cycle of the base pul determined by the upper six bits and the locations of the additional pulses by the subsequ bits with a method similar to as above.

Lower 6 bits							Base pulse No.																																										
							0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
0	0	0	0	0	0	0																																															
0	0	0	0	0	1	1																																															
0	0	0	0	1	0	2																																○															
0	0	0	0	1	1	3																																○															
0	0	0	1	0	0	4																○																○															
0	0	0	1	0	1	5																○																○															
0	0	0	1	1	0	6																○								○								○															
0	0	0	1	1	1	7																○								○								○								○							
0	0	1	0	0	0	8								○								○								○								○								○							
0	0	1	0	0	1	9								○								○								○								○								○							
0	0	1	0	1	0	10								○								○								○				○				○								○							
0	0	1	0	1	1	11								○								○								○				○				○								○				○			
0	0	1	1	0	0	12								○				○				○								○				○				○								○				○			
0	0	1	1	0	1	13								○				○				○								○				○				○								○				○			
0	0	1	1	1	0	14								○				○				○				○				○				○				○								○				○			
0	0	1	1	1	1	15								○				○				○				○				○				○				○				○				○				○			
0	1	0	0	0	0	16				○				○				○				○				○				○				○				○				○				○				○			
0	1	0	0	0	1	17				○				○				○				○				○				○				○				○				○				○				○			
0	1	0	0	1	0	18				○				○				○				○				○				○				○		○		○				○				○				○			
0	1	0	0	1	1	19				○				○				○				○				○				○				○		○		○				○				○				○		○	
0	1	0	1	0	0	20				○				○				○		○		○				○				○				○		○		○				○				○				○		○	
0	1	0	1	0	1	21				○				○				○		○		○				○				○				○		○		○				○				○				○		○	
0	1	0	1	1	0	22				○				○				○		○		○				○		○		○				○		○		○				○				○				○		○	
0	1	0	1	1	1	23				○				○				○		○		○				○		○		○				○		○		○				○		○		○				○		○	
0	1	1	0	0	0	24				○		○		○				○		○		○				○		○		○				○		○		○				○		○		○				○		○	
0	1	1	0	0	1	25				○		○		○				○		○		○				○		○		○				○		○		○				○		○		○				○		○	
0	1	1	0	1	0	26				○		○		○				○		○		○				○		○		○		○		○		○		○				○		○		○				○		○	
0	1	1	0	1	1	27				○		○		○				○		○		○				○		○		○		○		○		○		○				○		○		○		○		○		○	
0	1	1	1	0	0	28				○		○		○		○		○		○		○				○		○		○		○		○		○		○				○		○		○		○		○		○	
0	1	1	1	0	1	29				○		○		○		○		○		○		○				○		○		○		○		○		○		○				○		○		○		○		○		○	
0	1	1	1	1	0	30				○		○		○		○		○		○		○		○		○		○		○		○		○		○		○				○		○		○		○		○		○	
0	1	1	1	1	1	31				○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○	
1	0	0	0	0	0	32		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○	
1	0	0	0	0	1	33		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○	
1	0	0	0	1	0	34		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○	○	○		○		○		○		○		○		○	
1	0	0	0	1	1	35		○		○		○		○		○		○		○		○		○		○		○		○		○		○		○	○	○		○		○		○		○		○		○	○
1	0	0	1	0	0	36		○		○		○		○		○		○		○	○	○		○		○		○		○		○		○		○	○	○		○		○		○		○		○		○	○
1	0	0	1	0	1	37		○		○		○		○		○		○		○	○	○		○		○		○		○		○		○		○	○	○		○		○		○		○		○		○	○
1	0	0	1	1	0	38		○		○		○		○		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○		○		○		○	○
1	0	0	1	1	1	39		○		○		○		○		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○
1	0	1	0	0	0	40		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○
1	0	1	0	0	1	41		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○		○		○	○
1	0	1	0	1	0	42		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○	○	○		○	○	○		○		○		○	○	○		○		○		○	○
1	0	1	0	1	1	43		○		○		○	○	○		○		○		○	○	○		○		○		○	○	○		○	○	○		○	○	○		○		○		○	○	○		○	○	○		○	○
1	0	1	1	0	0	44		○		○		○	○	○		○	○	○		○	○	○		○		○		○	○	○		○	○	○		○	○	○		○		○		○	○	○		○	○	○		○	○
1	0	1	1	0	1	45		○		○		○	○	○		○	○	○		○	○	○		○		○		○	○	○		○	○	○		○	○	○		○		○		○	○	○		○	○	○		○	○
1	0	1	1	1	0	46		○		○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○		○		○	○	○		○	○	○		○	○
1	0	1	1	1	1	47		○		○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○
1	1	0	0	0	0	48		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○
1	1	0	0	0	1	49		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○
1	1	0	0	1	0	50		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○		○	○	○		○	○	○		○	○
1	1	0	0	1	1	51		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○		○	○	○		○	○	○	○	○	○
1	1	0	1	0	0	52		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○		○	○	○		○	○	○	○	○	○
1	1	0	1	0	1	53		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○		○	○	○		○	○	○	○	○	○
1	1	0	1	1	0	54		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○		○	○	○		○	○	○	○	○	○
1	1	0	1	1	1	55		○	○	○		○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○
1	1	1	0	0	0	56		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○
1	1	1	0	0	1	57		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○
1	1	1	0	1	0	58		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○
1	1	1	0	1	1	59		○	○	○	○	○	○	○		○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○
1	1	1	1	0	0	60		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○
1	1	1	1	0	1	61		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○

channel 1

- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - 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 7-phase PWM output possible by combination with synchronous op
- Buffer operation settable for channel 0
- Phase counting mode settable independently for each of channels 1 and 2
- Fast access via internal 16-bit bus
- 13 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated

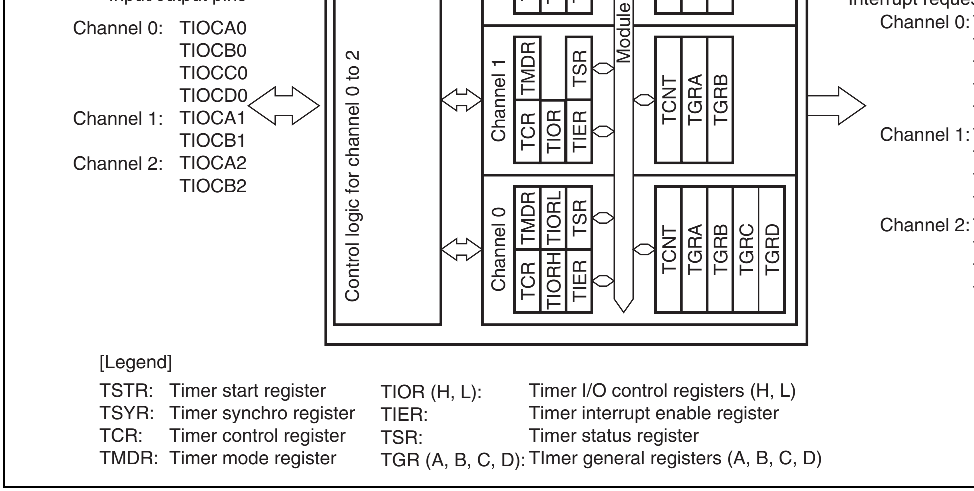

Channel 0: TIOCA0
TIOCB0
TIOCC0
TIOCD0
Channel 1: TIOCA1
TIOCB1
Channel 2: TIOCA2
TIOCB2
Control logic for channel 0 to 2
Module
Channel 1
TCR
TMDR
TIOR
TIER
TSR
TCNT
TGRA
TGRB
Channel 0
TCR
TMDR
TIORH
TIORL
TIER
TSR
TCNT
TGRA
TGRB
TGRC
TGRD
Channel 0:
Channel 1:
Channel 2:
[Legend]
TSTR: Timer start register
TSYR: Timer synchro register
TCR: Timer control register
TMDR: Timer mode register
TIOR (H, L): Timer I/O control registers (H, L)
TIER: Timer interrupt enable register
TSR: Timer status register
TGR (A, B, C, D): TImer general registers (A, B, C, D)

Figure 10.1 Block Diagram of TPU

		TCLKD		TCLKC
General registers (TGR)		TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2
General registers/buffer registers		TGRC_0 TGRC_0	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare m input capture
Compare match output	0 output	O	O	O
	1 output	O	O	O
	Toggle output	O	O	O
Input capture function		O	O	O
Synchronous operation		O	O	O
PWM mode		O	O	O
Phase counting mode		—	O	O
Buffer operation		O	—	—

- Compare match or input capture 0D
- Overflow

[Legend]

O: Enable

—: Disable

	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase inp
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	TGRA_2 input capture input/output compare output/PWM output pin

	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FE52	8
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FE53	8
	Timer interrupt enable register_0	TIER_0	R/W	H'40	H'FE54	8
	Timer status register_0	TSR_0	R/W	H'C0	H'FE55	8
	Timer counter_0	TCNT_0	R/W	H'0000	H'FE56	16
	Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FE58	16
	Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FE5A	16
	Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FE5C	16
	Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FE5E	16
Channel 1	Timer control register_1	TCR_1	R/W	H'00	H'FD40	8
	Timer mode register_1	TMDR_1	R/W	H'C0	H'FD41	8
	Timer I/O control register _1	TIOR_1	R/W	H'00	H'FD42	8
	Timer interrupt enable register_1	TIER_1	R/W	H'40	H'FD44	8
	Timer status register_1	TSR_1	R/W	H'C0	H'FD45	8
	Timer counter_1	TCNT_1	R/W	H'0000	H'FD46	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FD48	16
	Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FD4A	16
Channel 2	Timer control register_2	TCR_2	R/W	H'00	H'FE70	8
	Timer mode register_2	TMDR_2	R/W	H'C0	H'FE71	8
	Timer I/O control register_2	TIOR_2	R/W	H'00	H'FE72	8
	Timer interrupt enable register_2	TIER_2	R/W	H'40	H'FE74	8
	Timer status register_2	TSR_2	R/W	H'C0	H'FE75	8
	Timer counter_2	TCNT_2	R/W	H'0000	H'FE76	16

The TCR registers control the TCNT operation for each channel. The TPU has a total of TCR registers, one for each channel (channel 0 to 2). TCR register settings should be ma 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 s See tables 10.4 and 10.5 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 clock is counted using both edges, the input c is divided in 2 (ϕ/4 both edges = ϕ/2 rising edg phase counting mode is used on channels 1, 5, this setting is ignored and the phase counti setting has priority. Internal clock edge select when the input clock is ϕ/4 or slower. This set ignored if the input clock is ϕ/1 and rising edg selected. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. Th source can be selected independently for eac See tables 10.6 to 10.8 for details.
0	TPSC0	0	R/W	

[Legend]

x: Don't care

				synchronous/clearing synchronou 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 another channel performing sync clearing/synchronous operation*[1]

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR t

2. When TGRC or TGRD is used as a buffer register. TCNT is not cleared becau buffer register setting has priority, and compare match/input capture dose not c

Table 10.5 CCLR2 to CCLR0 (channels 1 and 2)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	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 another channel performing sync clearing/synchronous operation*[1]

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR t

2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be mod

		1	0	External clock: counts on TCLKC
			1	External clock: counts on TCLKD

Table 10.7 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	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
			1	External clock: counts on TCLKB
		1	0	Internal clock: counts on ϕ/256
			1	Setting prohibited

Note: This setting is ignored when channel 1 is in phase counting mode.

1	0	External clock: counts on TCLKC
	1	Internal clock: counts on φ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the r way, or TGRB and TGRD are to be used toge buffer operation. When TGRD is used as a bu register. TGRD input capture/output compare generation. Because channels 1 and 2 have r bit 5 is reserved. It is always read as 0 and ca modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer c
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the r way, or TGRA and TGRC are to be used toge buffer operation. When TGRC is used as a bu register, TGRC input capture/output compare generated. Because channels 1 and 2 have n bit 4 is reserved. It is always read as 0 and ca modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer c
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	These bits are used to set the timer operating
1	MD1	0	R/W	MD3 is a reserved bit. In a write, the write valu always be 0. See table 10.9 for details.
0	MD0	0	R/W	

		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	×	×	×	Setting prohibited

[Legend]

x: Don't care

Notes: 1. MD3 is reserved bit. In a write, it should be written with 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 shoul be written to MD2.

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.
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.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- TIORL_0

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.
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.
1	IOC1	0	R/W	
0	IOC0	0	R/W	

			1		Initial output is 0 output Toggle output at compare mat
	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 mat
1	0	0	0	Input capture register	Capture input source is TIOCE Input capture at rising edge
			1		Capture input source is TIOCE Input capture at falling edge
		1	×		Capture input source is TIOCE Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

			1		Initial output is 0 output Toggle output at compare ma
	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 ma
1	0	0	0	Input capture register	Capture input source is TIOC Input capture at rising edge
			1		Capture input source is TIOC Input capture at falling edge
		1	×		Capture input source is TIOC Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

			1		Initial output is 0 output Toggle output at compare mat
	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 mat
1	0	0	0	Input capture register*	Capture input source is TIOCD Input capture at rising edge
			1		Capture input source is TIOCD Input capture at falling edge
		1	×		Capture input source is TIOCD Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

Note: When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, setting is invalid and input capture/output compare is not generated.

			1		Initial output is 0 output Toggle output at compare ma
	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 ma
1	0	0	0	Input capture register*	Capture input source is TIOC Input capture at rising edge
			1		Capture input source is TIOC Input capture at falling edge
		1	×		Capture input source is TIOC Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

Note: * When the BFA bit in TMDR_0 is set to 1and TGRC_0 is used as a buffer regi setting is invalid and input capture/output compare is not generated.

			1		Initial output is 0 output Toggle output at compare mat
	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 mat
1	0	0	0	Input capture register	Capture input source is TIOCE Input capture at rising edge
			1		Capture input source is TIOCE Input capture at falling edge
		1	×		Capture input source is TIOCE Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

			1		Initial output is 0 output Toggle output at compare ma
	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 ma
1	0	0	0	Input capture register	Capture input source is TIOC Input capture at rising edge
			1		Capture input source is TIOC Input capture at falling edge
		1	×		Capture input source is TIOC Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

			1		Initial output is 0 output Toggle output at compare mat
	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 mat
1	×	0	0	Input capture register	Capture input source is TIOCB Input capture at rising edge
			1		Capture input source is TIOCB Input capture at falling edge
		1	×		Capture input source is TIOCB Input capture at both edges

[Legend]

×: Don't care

			1		Initial output is 0 output Toggle output at compare ma
	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 ma
1	×	0	0	Input capture register	Capture input source is TIOC Input capture at rising edge
			1		Capture input source is TIOC Input capture at falling edge
		1	×		Capture input source is TIOC Input capture at both edges

[Legend]

×: Don't care

				0: A/D conversion start request generation dis 1: A/D conversion start request generation ena
6	—	1	R	Reserved This bit is always read as 1 and cannot be mod
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) b TCFU flag when the TCFU flag in TSR is set to channels 1 and 2. In channel 0, bit 5 is reserve 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) b TCFV flag when the TCFV flag in TSR is set to 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) b TGFD bit when the TGFD bit in TSR is set to 1 channel 0. In channels 1 and 2, bit 3 is reserve always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD disabled 1: Interrupt requests (TGID) by TGFD enabled.

				Enables or disables interrupt requests (TGIB) TGFB bit when the TGFB bit in TSR is set to 0: Interrupt requests (TGIB) by TGFB disable 1: Interrupt requests (TGIB) by TGFB enable
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) TGFA bit when the TGFA bit in TSR is set to 0: Interrupt requests (TGIA) by TGFA disable 1: Interrupt requests (TGIA) by TGFA enable

				0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1 and cannot be mod
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow occurred when channels 1 and 2 are set to pha counting mode. In channel 0, bit 5 is reserved. It is always read and cannot be modified. [Setting condition] When the TCNT value underflows (change fror to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU
4	TCFV	0	R/(W) *	Overflow Flag Status flag that indicates that TCNT overflow h occurred. [Setting condition] When the TCNT value overflows (change from to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV

				• When TCNT value is transferred to TGRD capture signal while TGRD is functioning a capture register [Clearing condition] When 0 is written to TGFD after reading TGF
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TC capture or compare match in channel 0. In channels 1 and 2, bit 2 is reserved. It is alw as 0 and cannot be modified. [Setting conditions] • When the TCNT = TGRC while TGRC is f as output compare register • When TCNT value is transferred to TGRC capture signal while TGRC is functioning a capture register [Clearing condition] When 0 is written to TGFC after reading TGF

				capture register [Clearing condition] When 0 is written to TGFB after reading TGFB
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TG capture or compare match. The write value sho always be 0 to clear this flag. [Setting conditions] • When TCNT = TGRA while TGRA is functio output compare register • When TCNT value is transferred to TGRA b capture signal while TGRA is functioning as capture register [Clearing condition] When 0 is written to TGFA after reading TGFA

Note: * The write value should always be 0 to clear the flag.

TGRC and TGRD for channel 0 can also be designated for operation as buffer registers. registers are initialized to H'FFFF by a reset. The TGR registers cannot be accessed in 8 they must always be accessed as a 16-bit unit. TGR buffer register combinations are TG TGRC and TGRB—TGRD.

10.3.8 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels TCNT of a channel performs counting when the corresponding bit in TSTR is set to 1. V setting the operating mode in TMDR or setting the count clock in TCR, first stop the TC counter.

Bit	Bit Name	Initial value	R/W	Description
7 to 3	—	0	R	Reserved The initial value should not be changed.
2	CST2	0	R/W	Counter Start 2 to 0 (CST2 to CST0)
1	CST1	0	R/W	These bits select operation or stoppage for T
0	CST0	0	R/W	If 0 is written to the CST bit during operation v TIOC pin designated for output, the counter s the TIOC pin output compare output level is r If TIOR is written to when the CST bit is clear the pin output level will be changed to the set output value. 0: TCNT_n count operation is stopped 1: TCNT_n performs count operation (n = 2 to 0)

2	SYNC2	0	R/W	Timer Synchro 2 to 0 These bits select whether operation is indepen or synchronized with other channels. When synchronous operation is selected, sync presetting of multiple channels, and synchrono clearing through counter clearing on another cl are possible. To set synchronous operation, the SYNC bits f least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC TCNT clearing source must also be set by mea bits CCLR2 to CCLR0 in TCR. 0: TCNT_n operates independently (TCNT presetting /clearing is unrelated to ot channels) 1: TCNT_n performs synchronous operation TCNT synchronous presetting/synchronous is possible (n = 2 to 0)
1	SYNC 1	0	R/W	
0	SYNC 0	0	R/W	

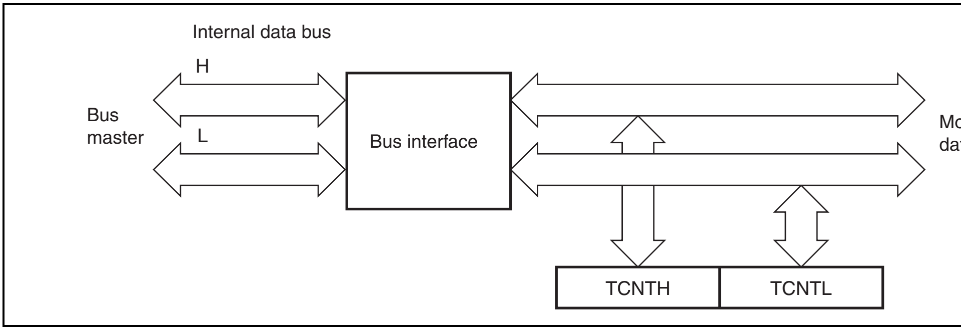


Figure 10.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bi

10.4.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wid
registers can be read and written to in 16-bit units. They can also be read and written to
units.

Examples of 8-bit register access operation are shown in figures 10.3, 10.4, and 10.5.

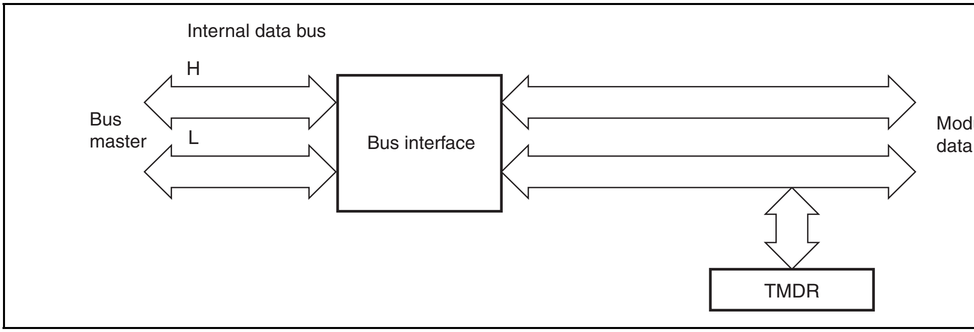


Figure 10.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8

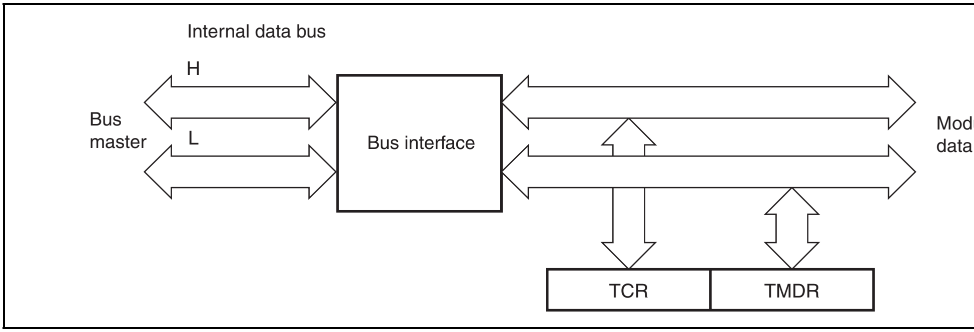
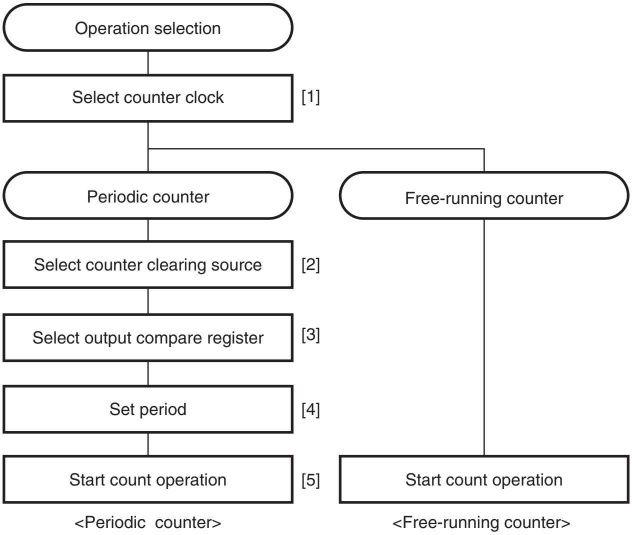


Figure 10.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (1

When one of bits CST0 to CST2 is set to 1 in TSTR, the TCNT counter for the correspo channel starts counting. TCNT can operate as a free-running counter, periodic counter, a

(a) Example of count operation setting procedure

Figure 10.6 shows an example of the count operation setting procedure.



[1] Select the counter clock wit TPSC2 to TPSC0 inTCR. At the same time, select the clock edge with bits CKEG1 CKEG0 in TCR.

[2] For periodic counter operati select the TGR to be used a TCNT clearing source with l CCLR2 to CCLR0 in TCR.

[3] Designate the TGR selected as an output compare regis means of TIOR.

[4] Set the periodic counter cyc the TGR selected in [2].

[5] Set the CST bit in TSTR to start the counter operation.

Figure 10.6 Example of Counter Operation Setting Procedure

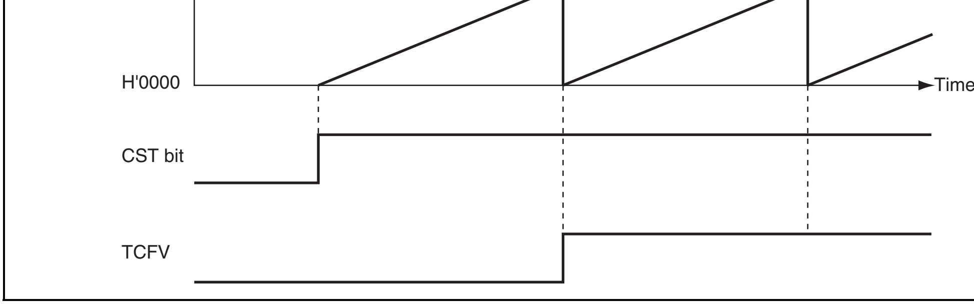


Figure 10.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the
channel performs periodic count operation. The TGR register for setting the period is des
as an output compare register, and counter clearing by compare match is selected by mea
CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count ope
periodic counter when the corresponding bit in TSTR is set to 1. When the count value m
the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000. If the va
the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. Afte
compare match, TCNT starts counting up again from H'0000. Figure 10.8 illustrates perio
counter operation.

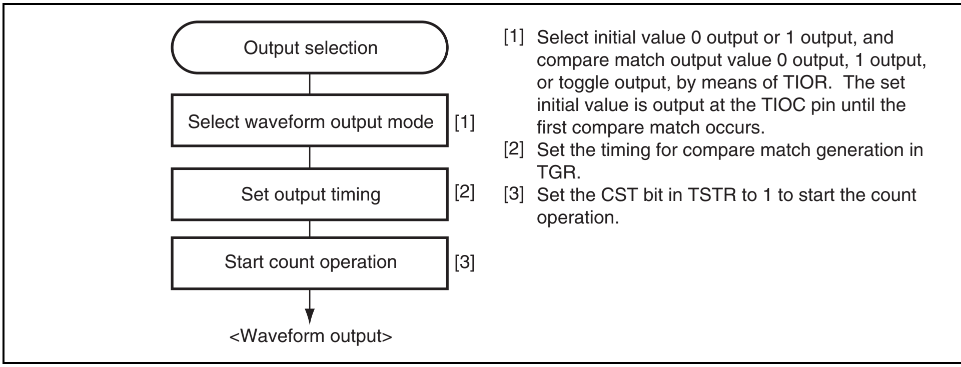
Figure 10.8 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using co
match.

(a) Example of setting procedure for waveform output by compare match

Figure 10.9 shows an example of the setting procedure for waveform output by compare



[1] Select initial value 0 output or 1 output, and compare match output value 0 output, 1 output, or toggle output, by means of TIOR. The set initial value is output at the TIOC pin until the first compare match occurs.

[2] Set the timing for compare match generation in TGR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.9 Example of Setting Procedure for Waveform Output by Compare

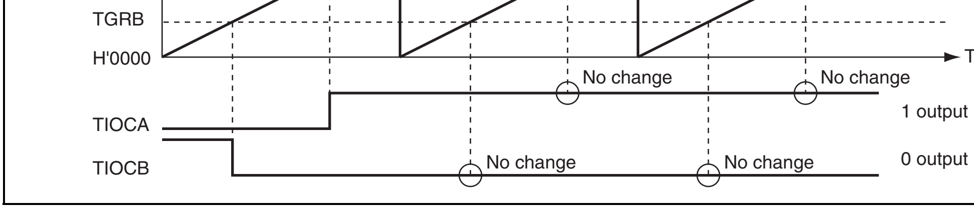


Figure 10.10 Example of 0 Output/1 Output Operation

Figure 10.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing p by compare match B), and settings have been made so that output is toggled by both com match A and compare match B.

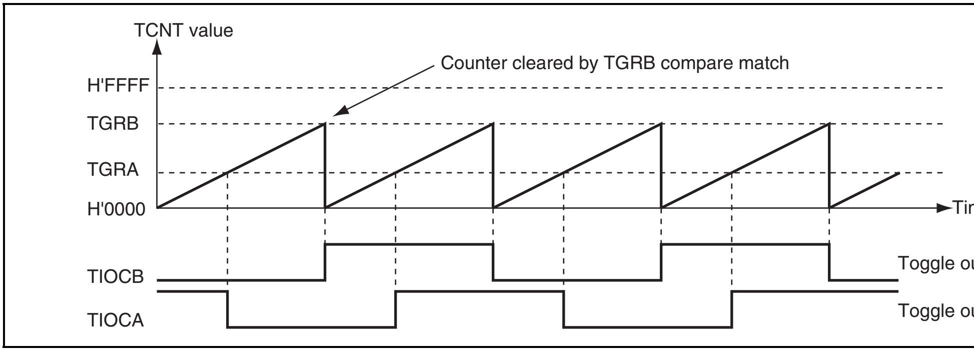
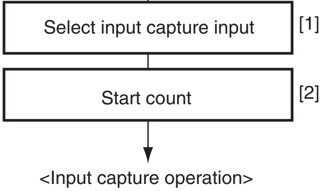


Figure 10.11 Example of Toggle Output Operation



edge, or both edges as the input capture source and input signal edge.

[2] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.12 Example of Input Capture Operation Setting Procedure

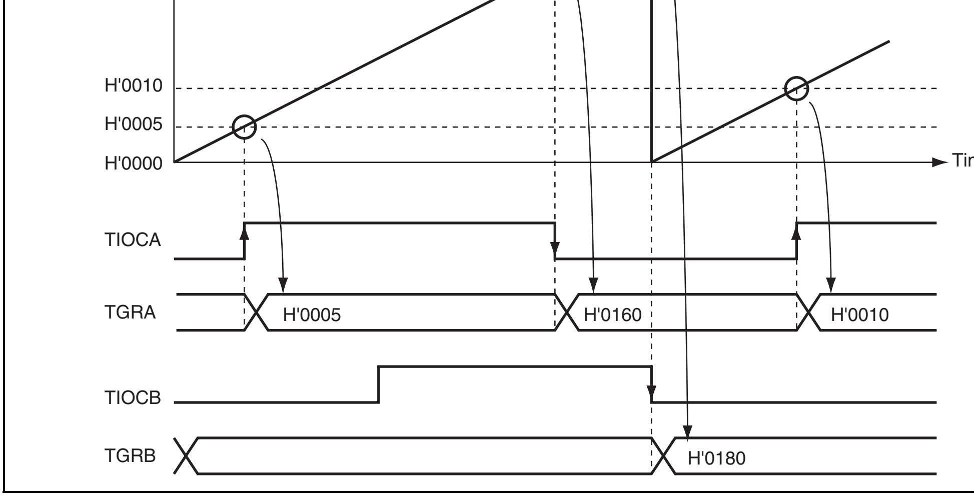
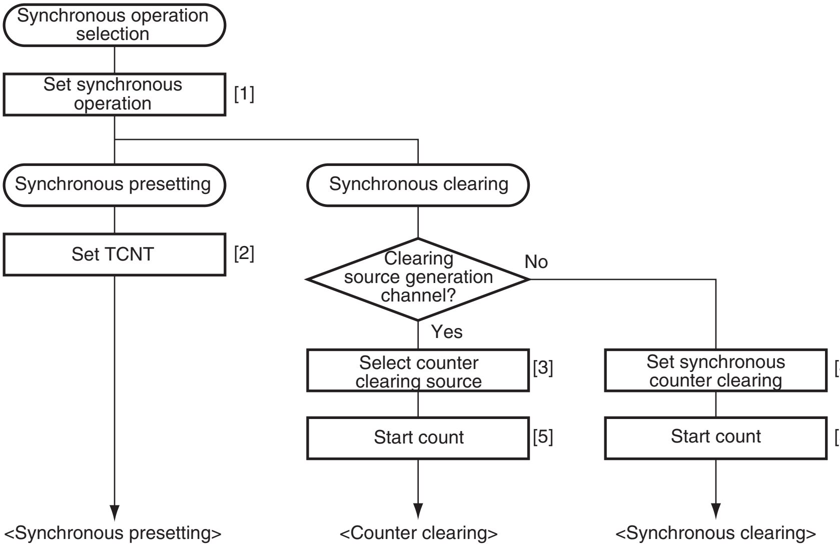


Figure 10.13 Example of Input Capture Operation



[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 10.14 Example of Synchronous Operation Setting Procedure

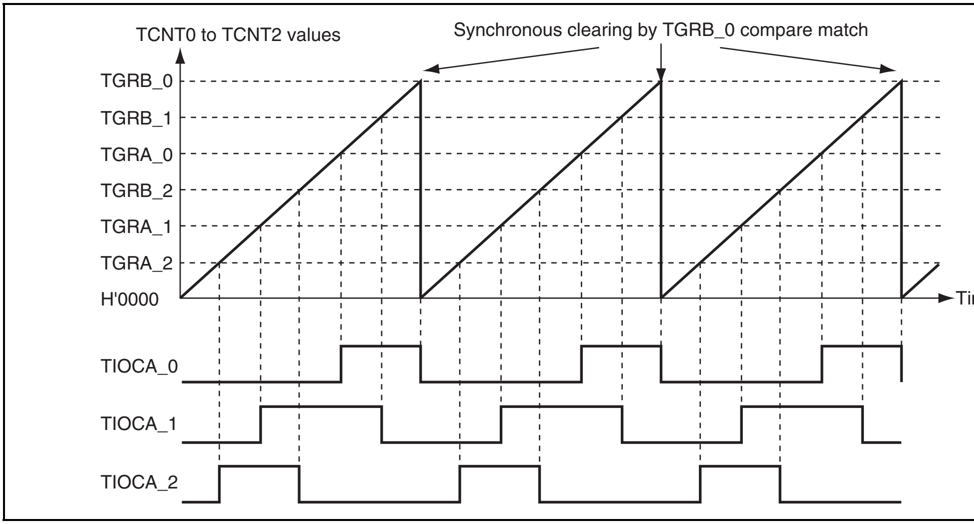


Figure 10.15 Example of Synchronous Operation

TGRB_0	TGRD_0

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding transferred to the timer general register. This operation is illustrated in figure 10.16.

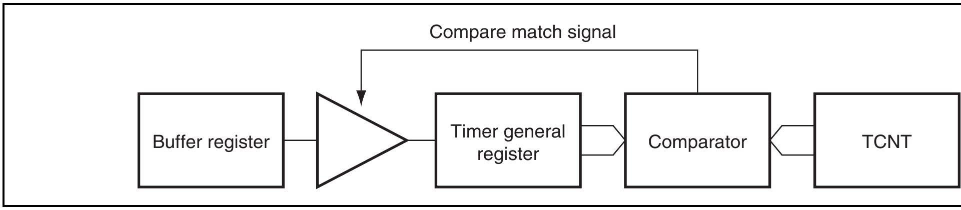


Figure 10.16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value p held in the timer general register is transferred to the buffer register. This operation i illustrated in figure 10.17.

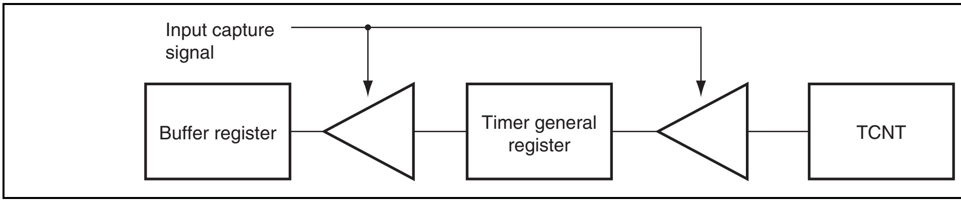


Figure 10.17 Input Capture Buffer Operation

Figure 10.18   Example of Buffer Operation Setting Procedure

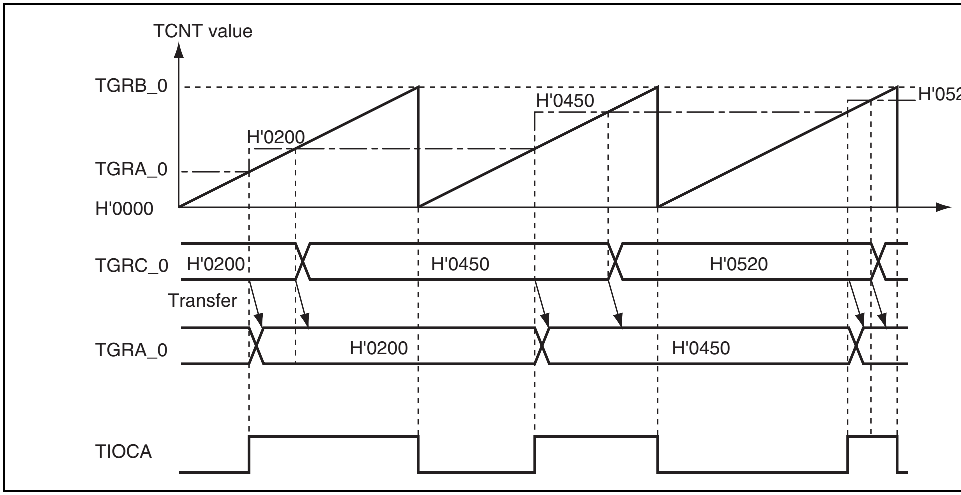


Figure 10.19 Example of Buffer Operation (1)

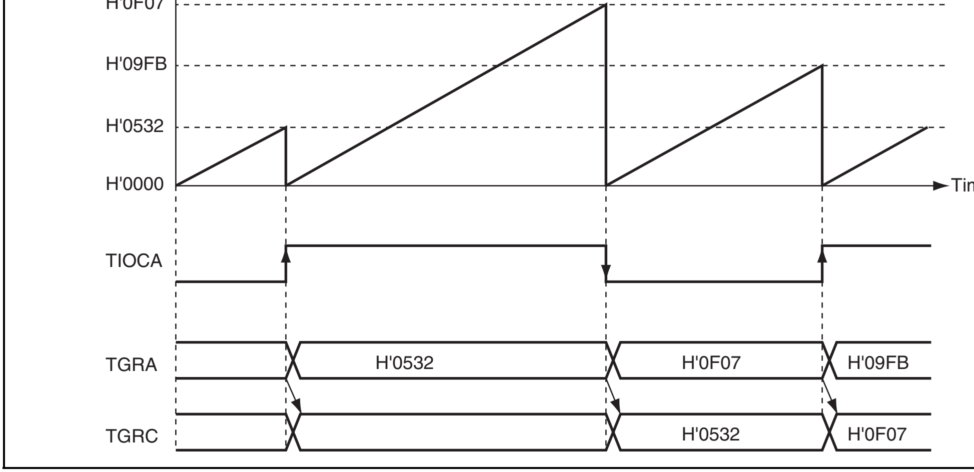


Figure 10.20   Example of Buffer Operation (2)

PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with
TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 i
output from the TIOCA and TIOCC pins at compare matches A and C, and the outpu
specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare ma
and D. The initial output value is the value set in TGRA or TGRC. If the set values o
TGRs are identical, the output value does not change when a compare match occurs.
mode 1, a maximum 4-phase PWM output is possible.

- PWM mode 2

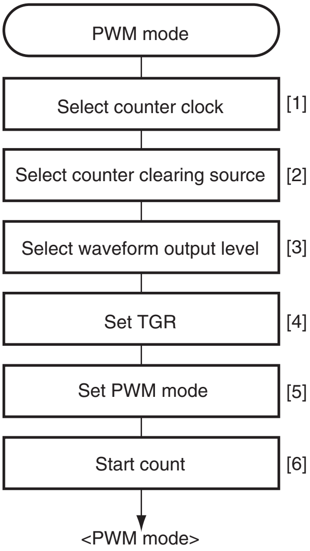
  PWM output is generated using one TGR as the cycle register and the others as duty
  The output specified in TIOR is performed by means of compare matches. Upon cou
  clearing by a synchronization register compare match, the output value of each pin is
  value set in TIOR. If the set values of the cycle and duty registers are identical, the o
  value does not change when a compare match occurs. In PWM mode 2, a maximum
  PWM output is possible by combined use with synchronous operation. The correspo
  between PWM output pins and registers is shown in table 10.19.

2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the peri

(1) Example of PWM Mode Setting Procedure

Figure 10.21 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.
[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.
[3] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.
[4] Set the cycle in the TGR selected in [2], and set the duty in the other the TGR.
[5] Select the PWM mode with bits MD3 to MD0 in TMDR.
[6] Set the CST bit in TSTR to 1 start the count operation.

Figure 10.21 Example of PWM Mode Setting Procedure

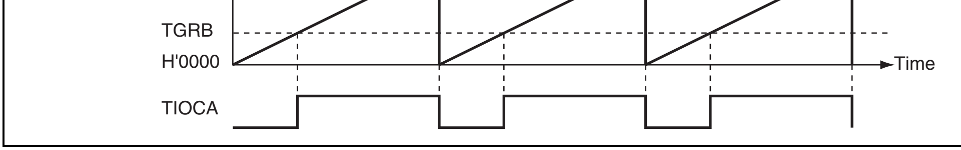


Figure 10.22 Example of PWM Mode Operation (1)

Figure 10.23 shows an example of PWM mode 2 operation. In this example, synchronou operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCN source, and 0 is set for the initial output value and 1 for the output value of the other TG (TGRA_0 to TGRD_0, TGRA_1), to output a 5-phase PWM waveform. In this case, the in TGRB_1 is used as the cycle, and the values set in the other TGRs as the duty.

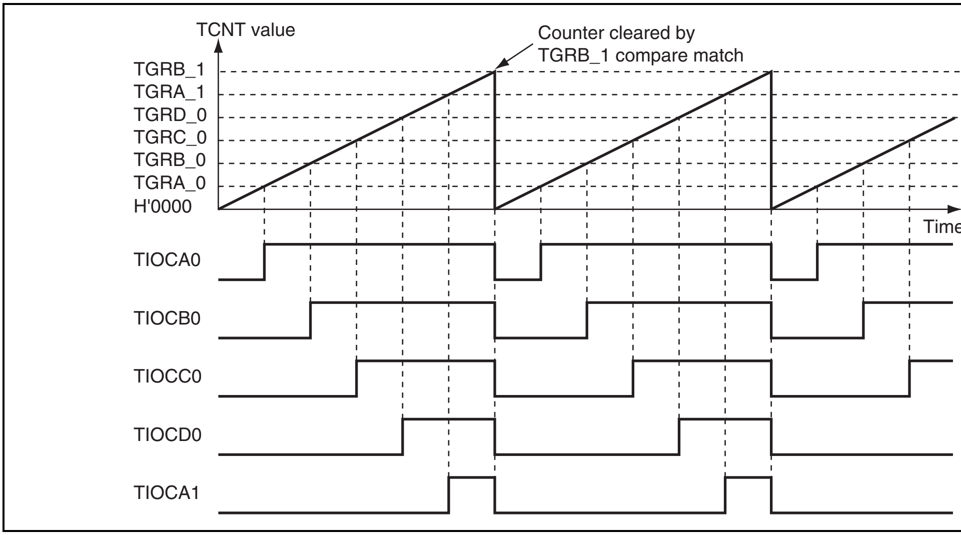


Figure 10.23 Example of PWM Mode Operation (2)

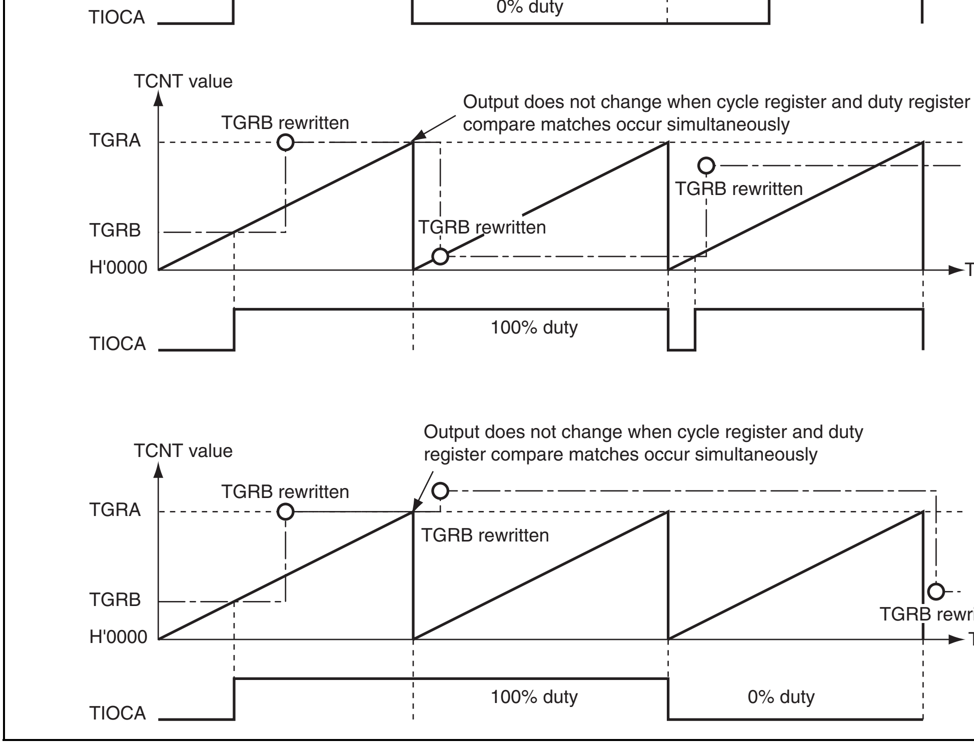

0% duty
TIOCA
TCNT value
Output does not change when cycle register and duty register compare matches occur simultaneously
TGRB rewritten
TGRA
TGRB rewritten
TGRB
TGRB rewritten
H'0000
T
100% duty
TIOCA
TCNT value
Output does not change when cycle register and duty register compare matches occur simultaneously
TGRB rewritten
TGRA
TGRB rewritten
TGRB
TGRB rewri
H'0000
T
100% duty
0% duty
TIOCA

Figure 10.24 Example of PWM Mode Operation (3)

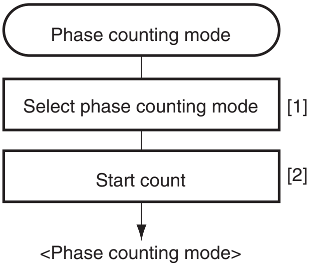
the TCFU flag is set. The TCFD bit in TSR is the count direction flag. Reading the TCF provides an indication of whether TCNT is counting up or down. Table 10.20 shows the correspondence between external clock pins and channels.

Table 10.20 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 10.25 shows an example of the phase counting mode setting procedure.



[1] Select phase counting mode with bits MD3 to MD0 in TMDR.

[2] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.25 Example of Phase Counting Mode Setting Procedure

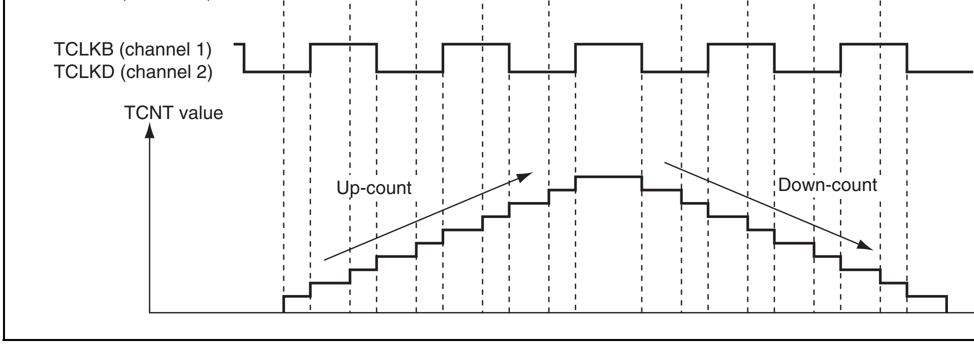


Figure 10.26 Example of Phase Counting Mode 1 Operation

Table 10.21 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	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

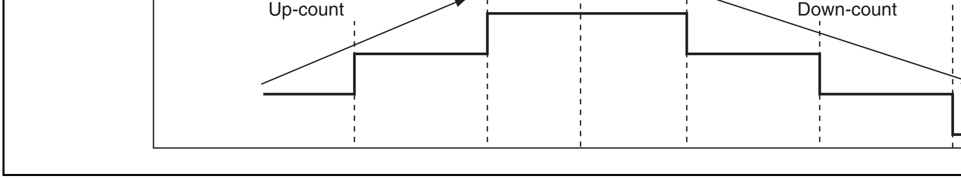


Figure 10.27 Example of Phase Counting Mode 2 Operation

Table 10.22 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) **TCLKC (Channel 2)**	**TCLKB (Channel 1)** **TCLKD (Channel 2)**	**Operation**
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

[Legend]

↑: Rising edge

↓: Falling edge

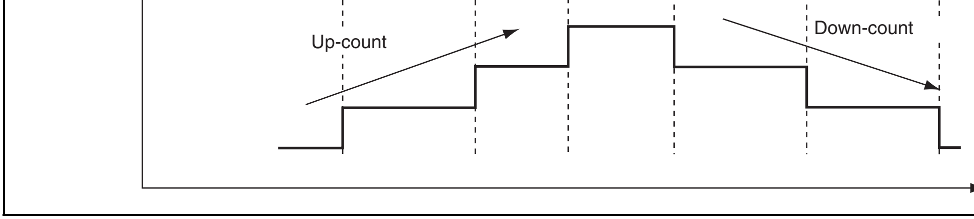


Figure 10.28 Example of Phase Counting Mode 3 Operation

Table 10.23 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) **TCLKC (Channel 2)**	**TCLKB (Channel 1)** **TCLKD (Channel 2)**	**Operation**
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]

↑:	Rising edge
↓:	Falling edge

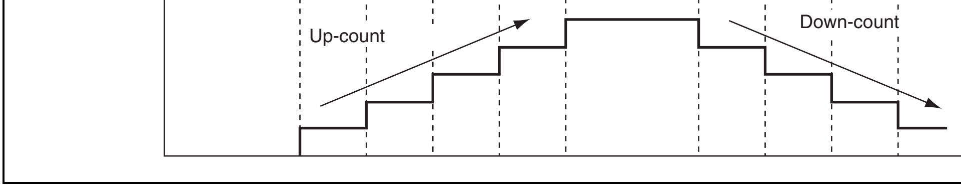


Figure 10.29 Example of Phase Counting Mode 4 Operation

Table 10.24 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	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

changed by the interrupt controller, but the priority order within a channel is fixed. For de section 5, Interrupt Controller. Table 10.25 lists the TPU interrupt sources.

Table 10.25 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	Pr
0	TGI0A	TGRA_0 input capture/compare match	TGFA	Hi
	TGI0B	TGRB_0 input capture/compare match	TGFB	
	TGI0C	TGRC_0 input capture/compare match	TGFC	
	TGI0D	TGRD_0 input capture/compare match	TGFD	
	TCI0V	TCNT_0 overflow	TCFV	
1	TGI1A	TGRA_1 input capture/compare match	TGFA	
	TGI1B	TGRB_1 input capture/compare match	TGFB	
	TCI1V	TCNT_1 overflow	TCFV	
	TCI1U	TCNT_1 underflow	TCFU	
2	TGI2A	TGRA_2 input capture/compare match	TGFA	
	TGI2B	TGRB_2 input capture/compare match	TGFB	
	TCI2V	TCNT_2 overflow	TCFV	
	TCI2U	TCNT_2 underflow	TCFU	Lo

Note: * This table shows the initial state immediately after a reset. The relative channe priorities can be changed by the interrupt controller.

the TCFV flag to 0. The TPU has three overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TS 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared b the TCFU flag to 0. The TPU has two underflow interrupts, one each for channels 1 and

10.6.2 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a ch the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurren TGRA input capture/compare match on a particular channel, a request to start A/D conv sent to the A/D converter. If the TPU conversion start trigger has been selected on the A converter side at this time, A/D conversion is started. In the TPU, a total of three TGRA capture/compare match interrupts can be used as A/D converter conversion start sources each channel.

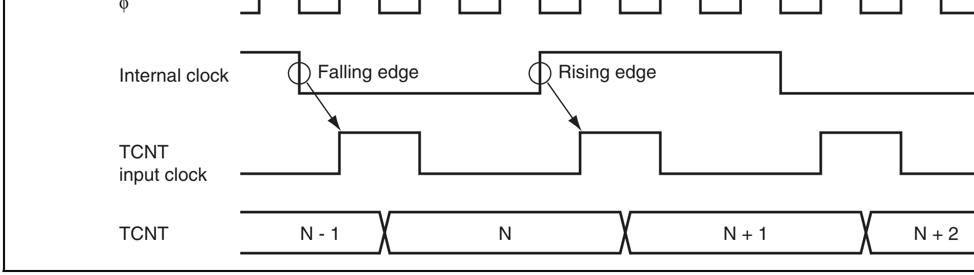


Figure 10.30 Count Timing in Internal Clock Operation

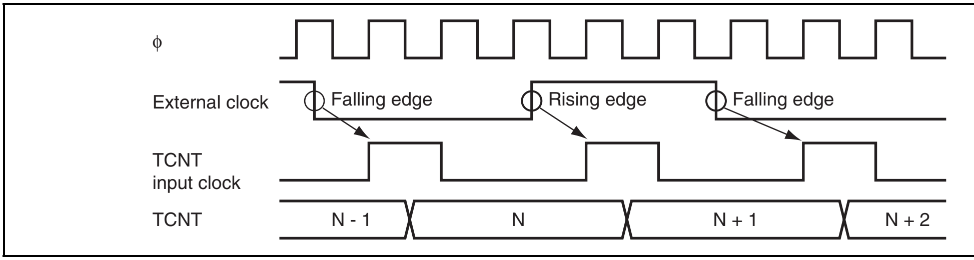


Figure 10.31 Count Timing in External Clock Operation

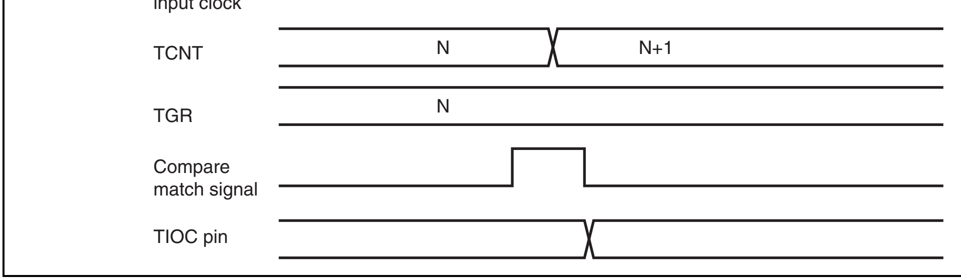


Figure 10.32   Output Compare Output Timing

(3)   Input Capture Signal Timing

Figure 10.33 shows input capture signal timing.

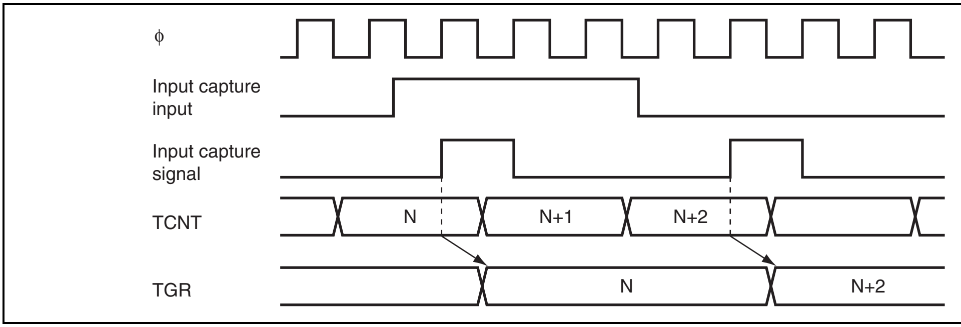


Figure 10.33   Input Capture Input Signal Timing

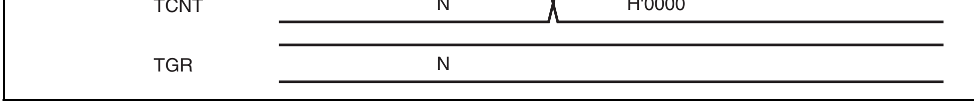


Figure 10.34   Counter Clear Timing (Compare Match)

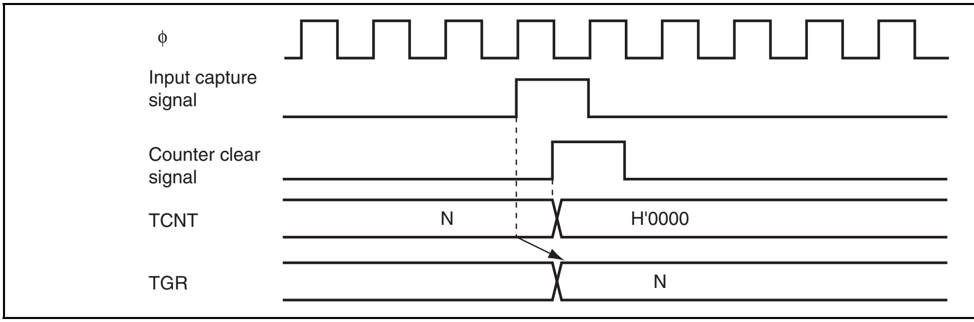


Figure 10.35   Counter Clear Timing (Input Capture)

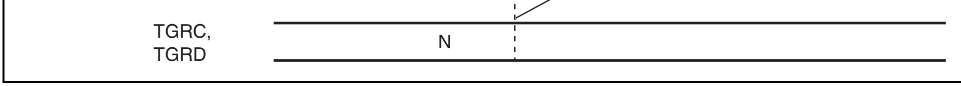


Figure 10.36   Buffer Operation Timing (Compare Match)

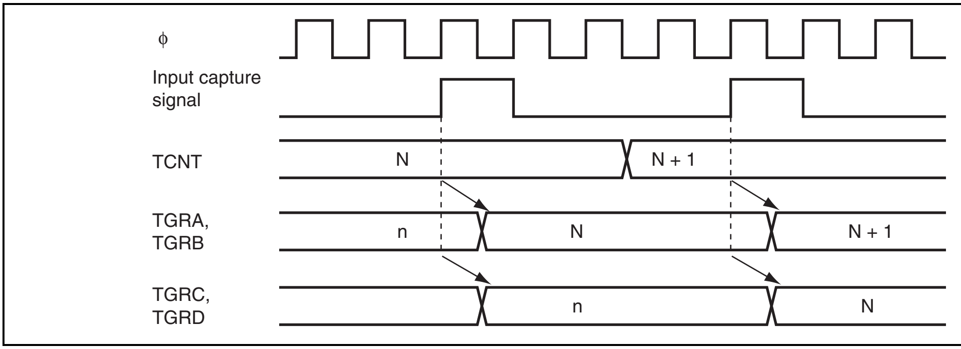


Figure 10.37   Buffer Operation Timing (Input Capture)

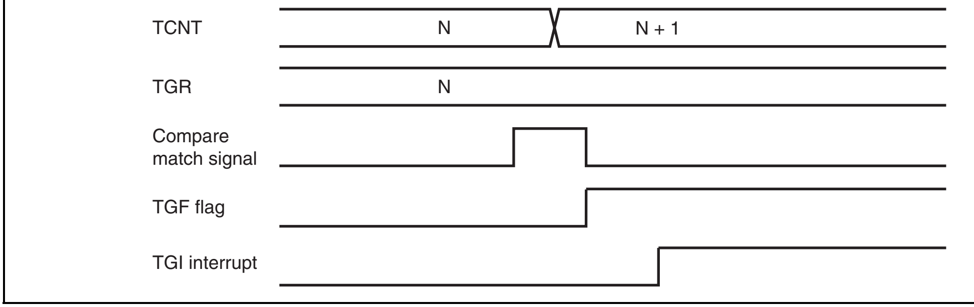


Figure 10.38 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 10.39 shows the timing for setting of the TGF flag in TSR by input capture occurr TGI interrupt request signal timing.

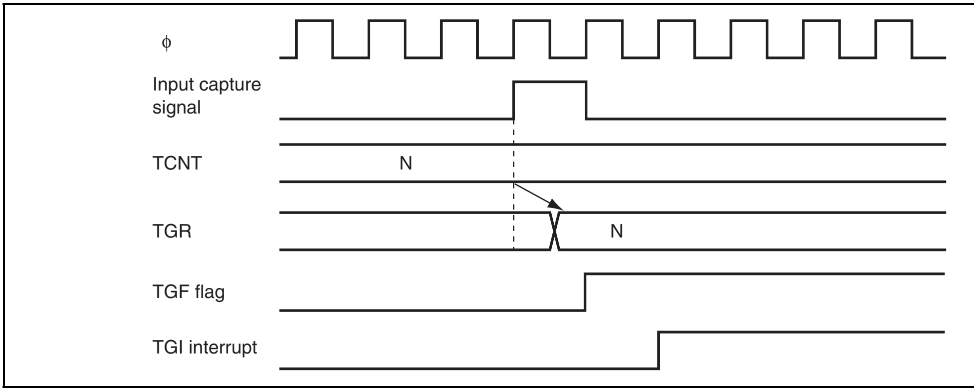


Figure 10.39 TGI Interrupt Timing (Input Capture)

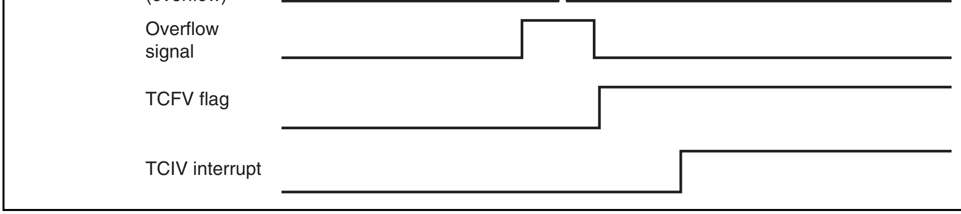


Figure 10.40 TCIV Interrupt Setting Timing

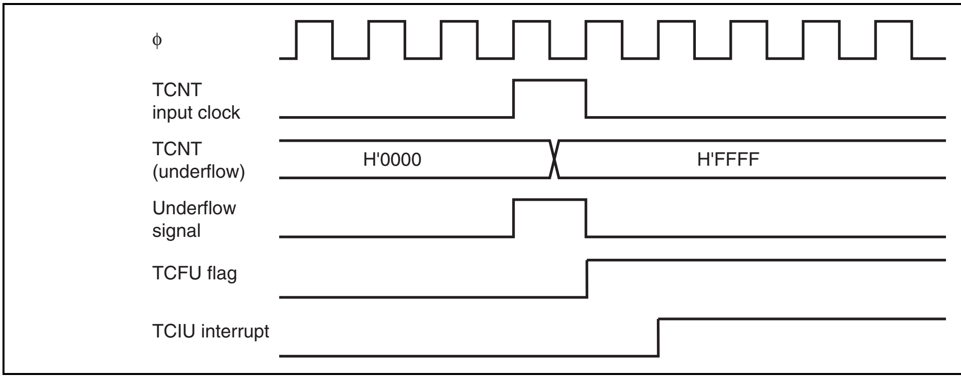


Figure 10.41 TCIU Interrupt Setting Timing

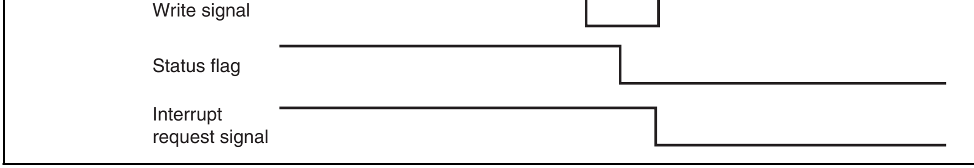
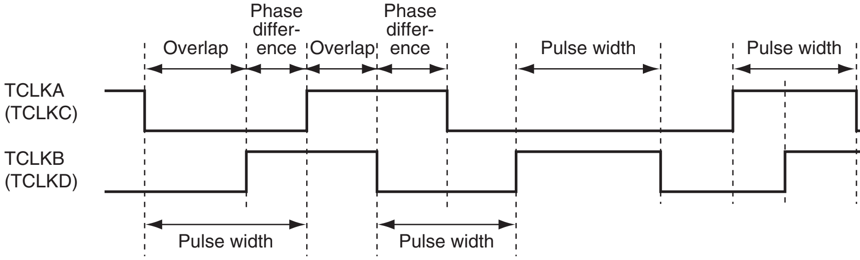


Figure 10.42 Timing for Status Flag Clearing by CPU



Notes: Phase difference and overlap : 1.5 states or more
Pulse width : 2.5 states or more

Figure 10.43 Phase Difference, Overlap, and Pulse Width in Phase Counting

10.8.2 Caution on Period Setting

When counter clearing by compare match is set, TCNT is cleared in the final state in wh matches the TGR value (the point at which the count value matched by TCNT is update 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

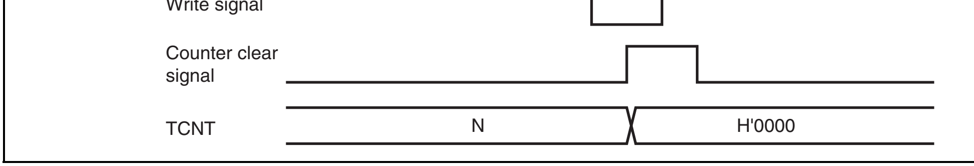


Figure 10.44 Conflict between TCNT Write and Clear Operations

10.8.4 Conflict between TCNT Write and Increment Operations

If incrementing occurs in the T_2 state of a TCNT write cycle, the TCNT write takes prece and TCNT is not incremented. Figure 10.45 shows the timing in this case.

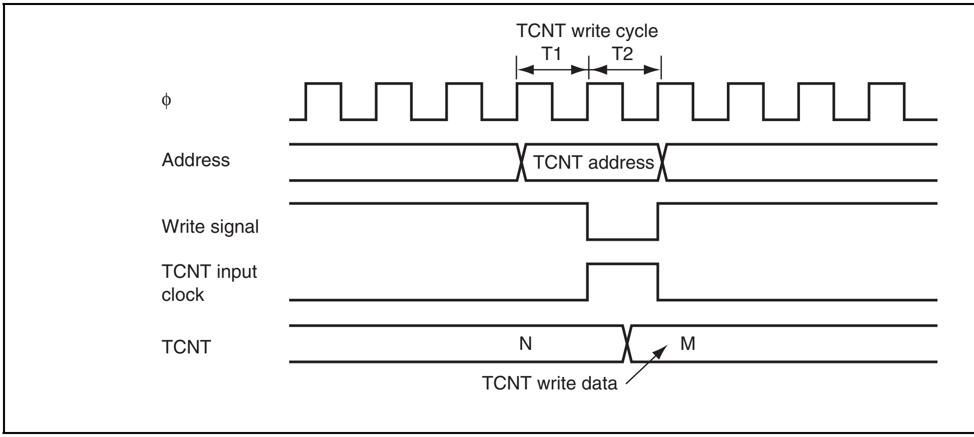


Figure 10.45 Conflict between TCNT Write and Increment Operations

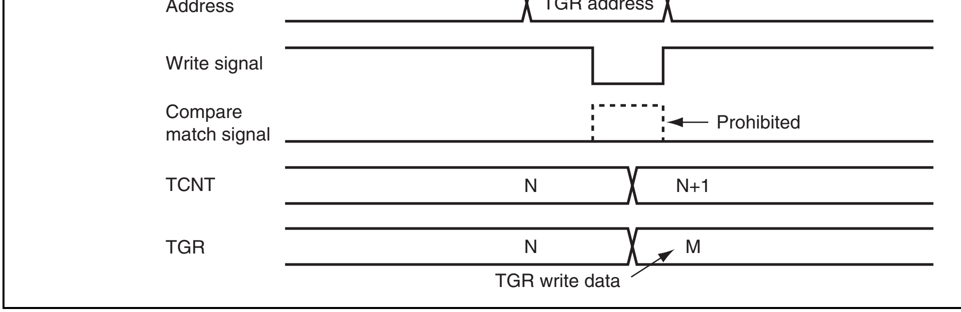


Figure 10.46   Conflict between TGR Write and Compare Match

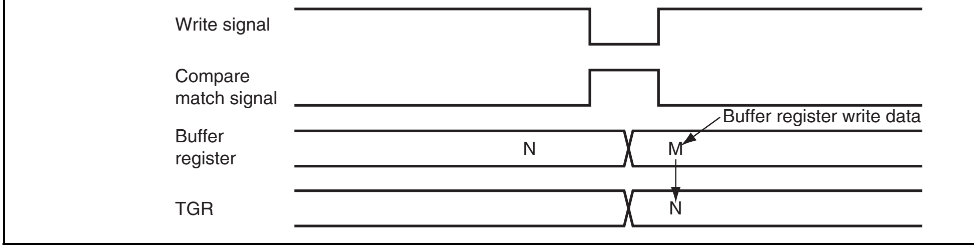


Figure 10.47 Conflict between Buffer Register Write and Compare Match

10.8.7 Conflict 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 will be the data after input capture transfer. Figure 10.48 shows the timing in this case.

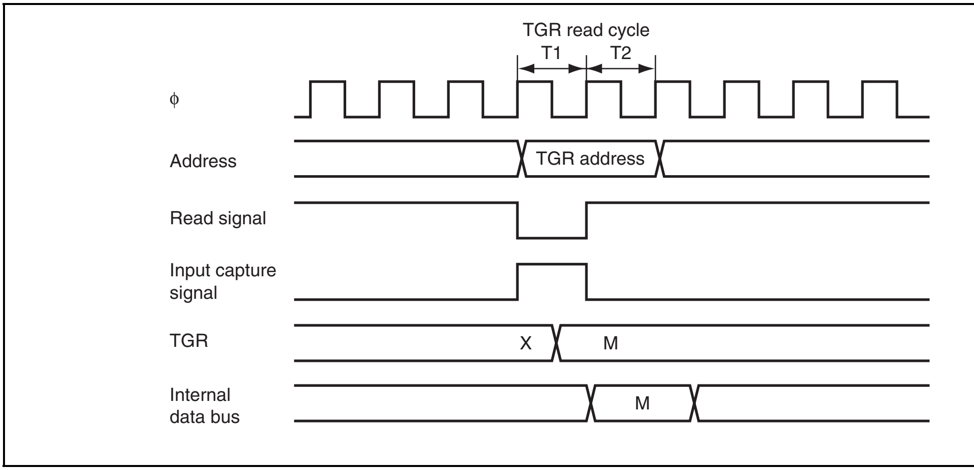


Figure 10.48 Conflict between TGR Read and Input Capture

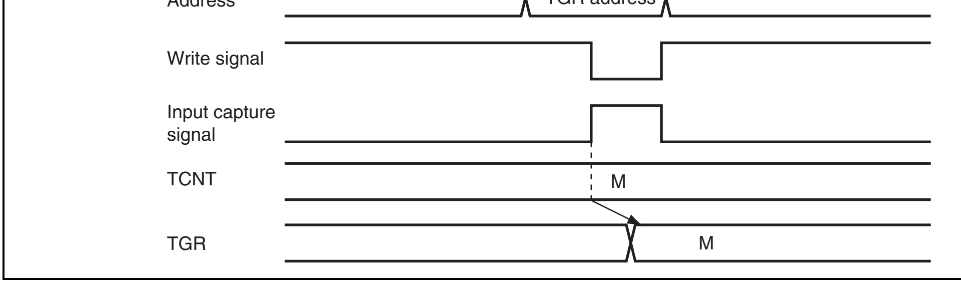


Figure 10.49   Conflict between TGR Write and Input Capture

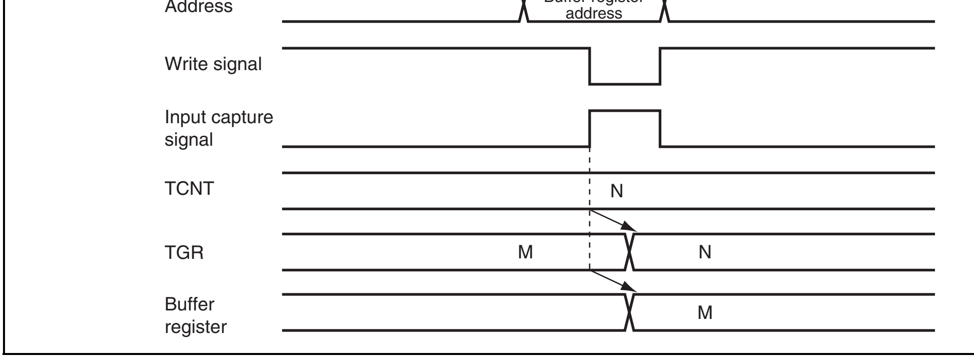


Figure 10.50   Conflict between Buffer Register Write and Input Capture

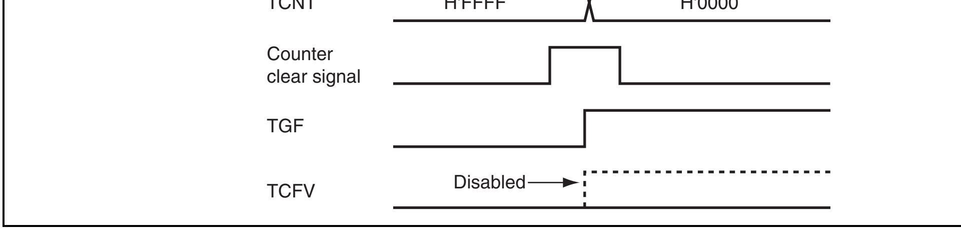


Figure 10.51 Conflict between Overflow and Counter Clearing

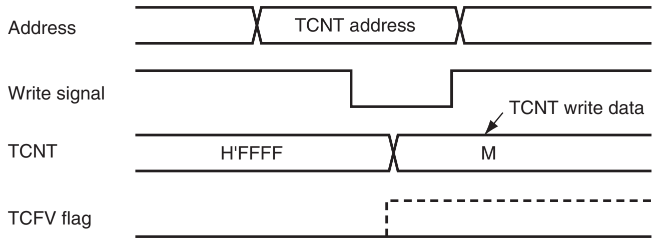


Figure 10.52 Conflict between TCNT Write and Overflow

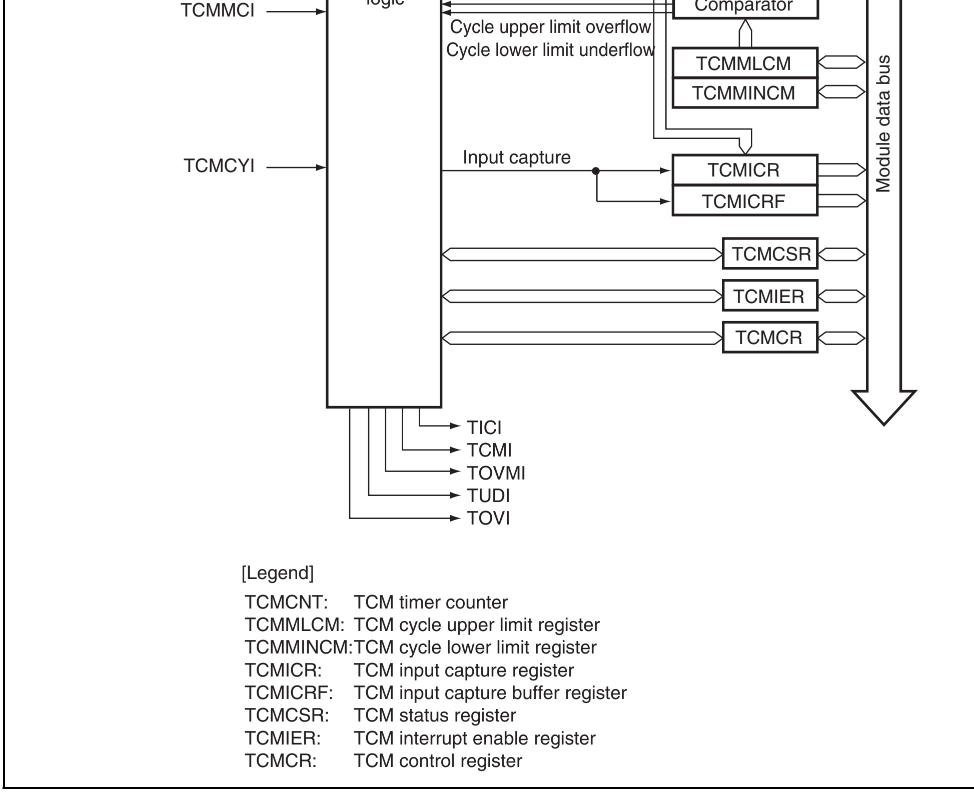
10.8.12 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB i with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLK pin with the TIOCB2 I/O pin. When an external clock is input, compare match output sho be performed from a multiplexed pin.

10.8.13 Module Stop Mode Setting

TPU operation can be enabled or disabled by the module stop control register. In the initi TPU operation is disabled. Access to TPU registers is enabled when module stop mode is cancelled. For details, see section 26, Power-Down Modes.

- 16-bit resolution
- Selectable counter clock
  - Any of seven internal clocks or an external clock
- Five interrupt sources
  - Counter overflow
  - Cycle upper limit overflow
  - Cycle lower limit underflow
  - Compare match
  - Triggering of input capture



[Legend]

TCMCNT:	TCM timer counter
TCMMLCM:	TCM cycle upper limit register
TCMMINCM:	TCM cycle lower limit register
TCMICR:	TCM input capture register
TCMICRF:	TCM input capture buffer register
TCMCSR:	TCM status register
TCMIER:	TCM interrupt enable register
TCMCR:	TCM control register

Figure 11.1 Block Diagram of the TCM

1	TCMCKI1	Input	External counter clock input
	(TCMMCI1)		Cycle measurement control input
	TCMCYI1	Input	External event input
2	TCMCKI2	Input	External counter clock input
	(TCMMCI2)		Cycle measurement control input
	TCMCYI2	Input	External event input
3	TCMCKI3	Input	External counter clock input
	(TCMMCI3)		Cycle measurement control input
	TCMCYI3	Input	External event input

11.3 Register Descriptions

The TCMs have the following registers.

Table 11.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address
Channel 0	TCM timer counter_0	TCMCNT_0	R/W	H'0000	H'FBC0
	TCM cycle upper limit register_0	TCMMLCM_0	R/W	H'FFFF	H'FBC2
	TCM cycle lower limit register_0	TCMMINCM_0	R/W	H'0000	H'FBCC
	TCM input capture register_0	TCMICR_0	R	H'0000	H'FBC4
	TCM input capture buffer register_0	TCMICRF_0	R	H'0000	H'FBC6
	TCM status register_0	TCMCSR_0	R/W	H'00	H'FBC8
	TCM control register_0	TCMCR_0	R/W	H'00	H'FBC9
	TCM interrupt enable register_0	TCMIER_0	R/W	H'00	H'FBCA

	TCM interrupt enable register_1	TCMIER_1	R/W	H'00	H'FBDA
Channel 2	TCM timer counter_2	TCMCNT_2	R/W	H'0000	H'FBE0
	TCM cycle upper limit register_2	TCMMLCM_2	R/W	H'FFFF	H'FBE2
	TCM cycle lower limit register_2	TCMMINCM_2	R/W	H'0000	H'FBEC
	TCM input capture register_2	TCMICR_2	R	H'0000	H'FBE4
	TCM input capture buffer register_2	TCMICRF_2	R	H'0000	H'FBE6
	TCM status register_2	TCMCSR_2	R/W	H'00	H'FBE8
	TCM control register_2	TCMCR_2	R/W	H'00	H'FBE9
	TCM interrupt enable register_2	TCMIER_2	R/W	H'00	H'FBEA
Channel 3	TCM timer counter_3	TCMCNT_3	R/W	H'0000	H'FBF0
	TCM cycle upper limit register_3	TCMMLCM_3	R/W	H'FFFF	H'FBF2
	TCM cycle lower limit register_3	TCMMINCM_3	R/W	H'0000	H'FBFC
	TCM input capture register_3	TCMICR_3	R	H'0000	H'FBF4
	TCM input capture buffer register_3	TCMICRF_3	R	H'0000	H'FBF6
	TCM status register_3	TCMCSR_3	R/W	H'00	H'FBF8
	TCM control register_3	TCMCR_3	R/W	H'00	H'FBF9
	TCM interrupt enable register_3	TCMIER_3	R/W	H'00	H'FBFA

In timer mode, TCMCNT is always writable. TCMCNT cannot be modified in cycle me mode. TCMCNT should always be accessed in 16-bit units and cannot be accessed in 8- TCMCNT is initialized to H'0000.

11.3.2 TCM Cycle Upper Limit Register (TCMMLCM)

TCMMLCM is a 16-bit readable/writable register. TCMMLCM is available as a compar register when the TCMMDS bit in TCMCR is cleared (operation is in timer mode). TCM is available as a cycle upper limit register when the TCMMDS bit in TCMCR is set to 1 is in cycle measurement mode).

In timer mode, the value in TCMMLCM is constantly compared with that in TCMCNT, values match, CMF in TCMCSR is set to 1. However, comparison is disabled in the sec a cycle of writing to TCMMLCM.

In cycle measurement mode, a value that sets an upper limit on the measurement period in TCMMLCM. When the second edge (first edge of the following cycle) of the measur period is detected, the value in TCMCNT is transferred to TCMICR. At this time, the va TCMICR and TCMMLCM are compared. The MAXOVF flag in TCMCSR is set to 1 if in TCMICR is greater than that in TCMMLCM. TCMMLCM should always be accesse units and cannot be accessed in 8-bit units. TCMMLCM is initialized to H'FFFF.

accessed in 16-bit units and cannot be accessed in 8-bit units. TCMMINCM is initialized
H'0000.

11.3.4 TCM Input Capture Register (TCMICR)

TCMICR is a 16-bit read-only register. In timer mode, the value in TCMCNT is transferr TCMICR on the edge selected by the IEDG bit in TCMCR. At the same time, the ICPF f TCMCSR is set to 1. In cycle measurement mode, the value in TCMCNT is transferred to TCMICR on detection of the second edge of the measurement period. At this time, the IC in TCMCSR is set to 1. TCMICR should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMICR is initialized to H'0000.

11.3.5 TCM Input Capture Buffer Register (TCMICRF)

TCMICRF is a 16-bit read only register. TCMICRF can be used as TCMICR buffer regis When input capture is generated, the value in TCMICR is transferred to TCMICRF.

TCMICR and TCMICRF should always be accessed in 16-bit units and cannot be accesse bit units. TCMICRF is initialized to H'0000.

				H'0000) [Clearing condition] Reading OVF when OVF = 1 and then writing 0 to
6	MAXOVF	0	R/(W)*	Measurement Period Upper Limit Overflow This flag indicates that the measured number of cy the waveform for measurement in cycle measuren has reached the upper limit set in TCMMLCM, cau overflow. [Setting condition] A greater value for TCMICR than TCMMLCM [Clearing condition] Reading MAXOVF when MAXOVF = 1 and then w MAXOVF
5	CMF	0	R/(W)*	Compare Match Flag (only valid in timer mode) [Setting condition] When the values in TCMCNT and TCMMLCM mat [Clearing condition] Reading CMF when CMF = 1 and then writing 0 to Note: CMF is not set in cycle measurement mode, when the values in TCMCNT and TCMMLC
4	CKSEG	0	R/W	External Clock Edge Select When bits CKS2 to CKS0 in TCMCR are set to B'1 bit selects the edge for counting of external count edge. 0: Count falling edges of the external clock. 1: Count rising edges of the external clock.

				IEDG bit in TCMICR) during the measurement perio [Setting condition] Generation of the input capture signal [Clearing condition] Reading ICPF when ICPF = 1 and then writing 0 to
2	MINUDF	0	R/(W)*	Measurement Period Lower Limit Underflow This flag indicates that the measured number of cyc the waveform for measurement in cycle measureme has reached the lower limit set in TCMMINCM, cau underflow. [Setting condition] A smaller value for TCMICR than TCMMINCM [Clearing condition] Reading MINUDF when MINUDF = 1 and then writi MINUDF
1	MCICTL	0	R/W	TCMMCI Input Polarity Inversion 0: TCMMCI input is inverted for use. 1: TCMMCI input is directly used. Note: Change this bit when CST = 0 and TCMMDS
0	—	0	R/W	Reserved The initial value should not be changed.

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

				operation stops. Clear this bit and thus return TCMCNT to H'0000 i initialization for cycle measurement mode.
6	POCTL	0	R/W	TCMCYI Input Polarity Reversal 0: Use the TCMCYI input directly 1: Use the inverted TCMCYI input Note: Modify this bit while CST = 0 and TCMMDS
5	CPSPE	0	R/W	Input Capture Stop Enable Controls whether or not counting up by TCMCNT a capture operation stop or continue when either of or MINUDF is set to 1 in cycle measurement mode does not affect operation in timer mode. 0: Counting up and input-capture operation contin the flag is set to 1. 1: Counting up and input-capture operation are dis when the flag is set to 1.

				1: Selects the falling edge of the TCMCYI input POCTL = 1 0: Selects the falling edge of the TCMCYI input 1: Selects the rising edge of the TCMCYI input
3	TCMMDS	0	R/W	TCM Mode Select Selects the TCM operating mode. 0: Timer mode The TCM provides compare match and input cap facilities. 1: Cycle measurement mode Setting this bit to 1 starts counting by TCMCNT. should be initialized to H'0000. Clear the CST in to 0 before setting to cycle measurement mode.
2	CKS2	0	R/W	Clock Select 2, 1, 0
1	CKS1	0	R/W	Selects the clock signal for input to TCMCNT.
0	CKS0	0	R/W	Note: Modify this bit when CST = 0 and TCMMDS = 000: Count ϕ/2 internal clock 001: Count ϕ/8 internal clock 010: Count ϕ/16 internal clock 011: Count ϕ/32 internal clock 100: Count ϕ/64 internal clock 101: Count ϕ/128 internal clock 110: Count ϕ/256 internal clock 111: Count external clock (select the external clock with CKSEG in TCMCSR.)

6	MAXOVIE	0	R/W	Cycle Upper Limit Overflow Interrupt Enable Enables or disables the issuing of interrupt reques setting of the MAXOVF flag in TCMCSR to 1. 0: Disable interrupt requests by MAXOVF 1: Enable interrupt requests by MAXOVF
5	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables the issuing of interrupt reques the CMF bit in TCMCSR is set to 1. 0: Disable interrupt requests by CMF 1: Enable interrupt requests by CMF
4	TCMIPE	0	R/W	Input Capture Input Enable Enables or disables input to the pin. When using i capture mode and cycle measurement mode, set 1. 0: Disable input 1: Enable input Note: Modify this bit when CST = 0 and TCMMDS
3	ICPIE	0	R/W	Input Capture Interrupt Enable Enables or disables interrupt requests when the I TCMCSR is set to 1. 0: Disable interrupt requests by ICPF 1: Enable interrupt requests by ICPF

				0: The TCMMCI signal is not used (cycle measuren always performed). 1: The TCMMCI signal is used. When MCICTL in TCMCSR is 0, cycle measuren performed only while TCMMCI is low. When MCI cycle measurement is performed only while TCM high. Note: Change this bit when CST = 0 and TCMMDS
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified

the CST bit in TCMCR is set to 1. When TCMCNT overflows (the value changes from H'0000), the OVF bit in TCMCSR is set to 1 and an interrupt request is generated if the in TCMIER is 1. Figure 11.2 shows an example of free running counter operation. In ad figure 11.3 shows TCMCNT count timing of external clock operation. The external cloc have a pulse width of no less than 1.5 cycles. The counter will not operate correctly if th are narrower than this.

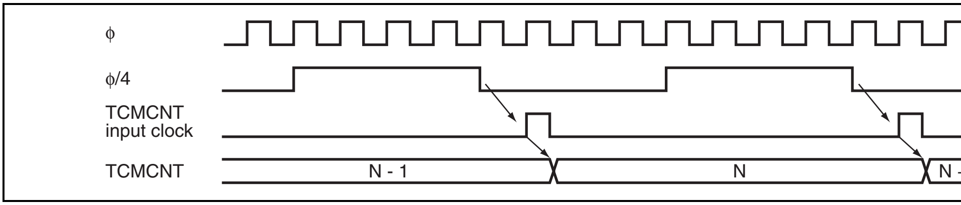


Figure 11.2 Example of Free Running Counter Operation

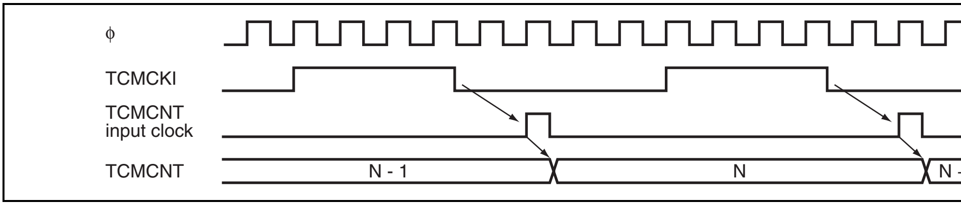


Figure 11.3 Count Timing of External Clock Operation (Falling Edges)

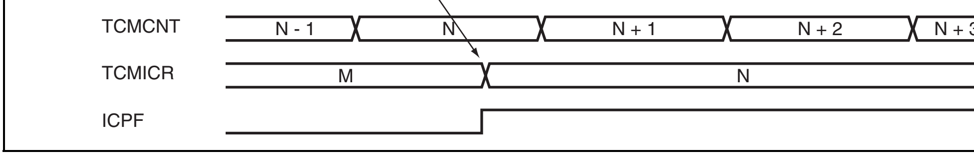


Figure 11.4 Input Capture Operation Timing (Sensing of Rising Edges)

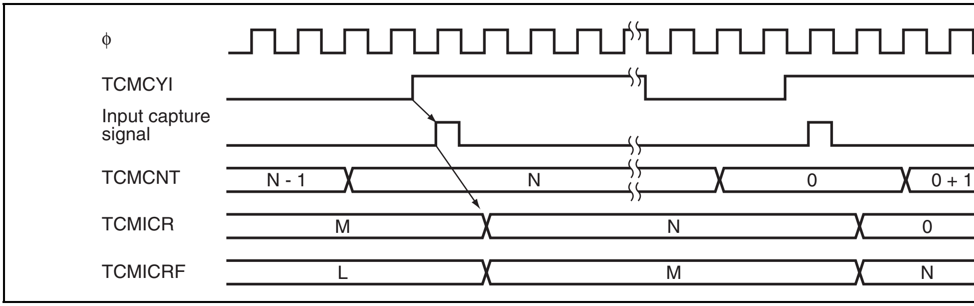


Figure 11.5 Buffer Operation of Input Capture

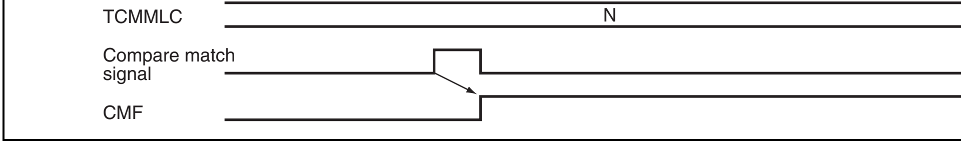


Figure 11.6 Timing of CMF Flag Setting on a Compare Match

11.4.2 Cycle Measurement Mode

When the TCMMDS bit in TCMCR is set to 1, the TCM operates in cycle measurement

(1) Counter Operation

Setting the TCMMDS bit in TCMCR to 1 selects cycle measurement mode, in which co
proceeds regardless of the setting of the CST bit in TCMCR. TCMCNT is cleared to H'0
detection of the first edge in the measurement period and counts up from there. Figure 1
an example of counter operation in cycle measurement mode.

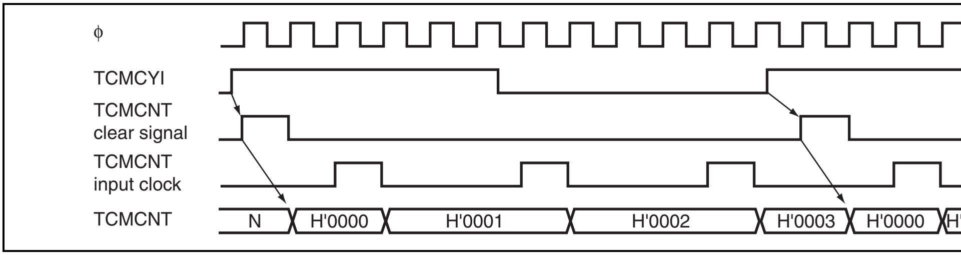


Figure 11.7 Example of Counter Operation in Cycle Measurement Mod

TCMCSR is set to 1. If TCMICR is smaller than TCMMINCM, the MINUDF bit in TCM set to 1. If generation of the corresponding interrupt request is enabled by the setting in T the request is generated. In addition, on detection of the third edge, TCMCNT is cleared t H'0000, and the next round of measurement starts.

When the CPSPE bit in TCMCR has been cleared to 0, the next round of cycle measurem start, even if the MAXOVF/MINUDF flag is set to 1.

If the MAXOVF/MINUDF flag is set to 1 while the CPSPE bit in TCMCR is set to 1, cou by TCMCNT stops and so does cycle measurement. Subsequently clearing MAXOVF/M to 0 automatically clears TCMCNT to H'0000, and counting up for cycle measurement is restarted.

Figure 11.8 shows an example of timing in speed measurement.

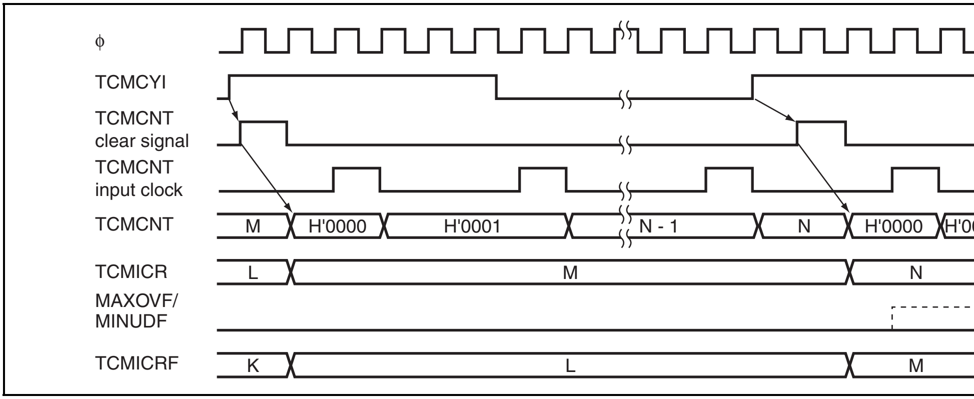


Figure 11.8 Example of Timing in Cycle Measurement

Figure 11.9 Example of Timing in Cycle Measurement when the CMMS Bit is

(3) Determination of External Event (TCMCYI) Stoppage

The timer overflow flag can be used to determine the external event (TCMCYI) stopped
Either of two sets of conditions represents the external event stopped state.

The external event can be considered to have stopped when a timer overflow is generate
the period from the start of cycle measurement mode to detection of the first edge (risin
as selected with the IEDG bit in TCMCR).

Figure 11.10 shows an example of the timing of the external event stopped state (1).

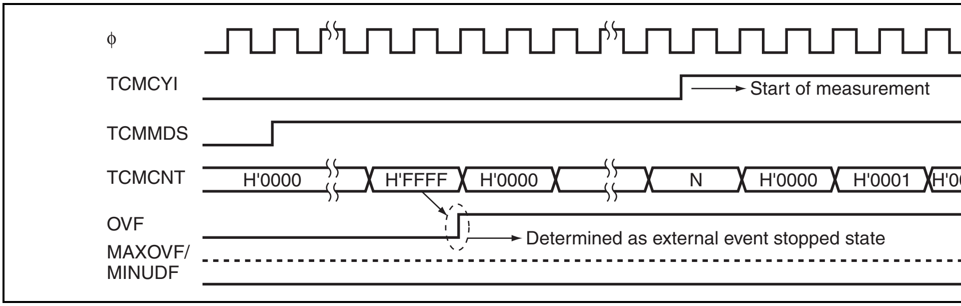


Figure 11.10 Example of Timing in External Event Stopped State (1)

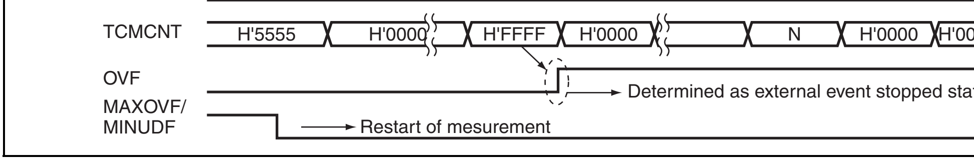
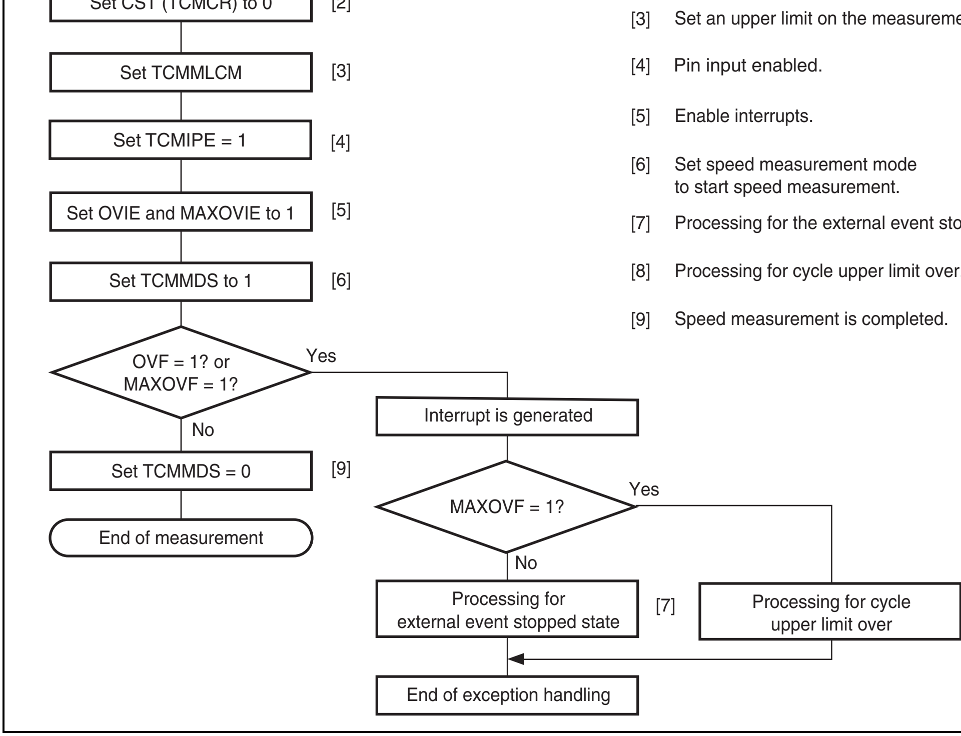


Figure 11.11 Example of Timing in External Event Stopped State (2)



[3] Set an upper limit on the measureme

[4] Pin input enabled.

[5] Enable interrupts.

[6] Set speed measurement mode to start speed measurement.

[7] Processing for the external event sto

[8] Processing for cycle upper limit over

[9] Speed measurement is completed.

Figure 11.12 Example of Cycle Measurement Mode Settings

Channel	Name	Interrupt Source	Interrupt Flag	Pr
TCM_0	TICI0	TCMICR_0 input capture	ICPF_0	Hi
	TCMI0	TCMMLCM_0 compare match	CMF_0	↑
	TOVMI0	TCMMLCM_0 overflow	MAXOVF_0	
	TUDI0	TCMMINCM_0 underflow	MINUDF_0	
	TOVI0	TCMCNT_0 overflow	OVF_0	
TCM_1	TICI1	TCMICR_1 input capture	ICPF_1	
	TCMI1	TCMMLCM_1 compare match	CMF_1	
	TOVMI1	TCMMLCM_1 overflow	MAXOVF_1	
	TUDI1	TCMMINCM_1 underflow	MINUDF_1	
	TOVI1	TCMCNT_1 overflow	OVF_1	
TCM_2	TICI2	TCMICR_2 input capture	ICPF_2	
	TCMI2	TCMMLCM_2 compare match	CMF_2	
	TOVMI2	TCMMLCM_2 overflow	MAXOVF_2	
	TUDI2	TCMMINCM_2 underflow	MINUDF_2	
	TOVI2	TCMCNT_2 overflow	OVF_2	
TCM_3	TICI3	TCMICR_3 input capture	ICPF_3	
	TCMI3	TCMMLCM_3 compare match	CMF_3	
	TOVMI3	TCMMLCM_3 overflow	MAXOVF_3	
	TUDI3	TCMMINCM_3 underflow	MINUDF_3	
	TOVI3	TCMCNT_3 overflow	OVF_3	Lo

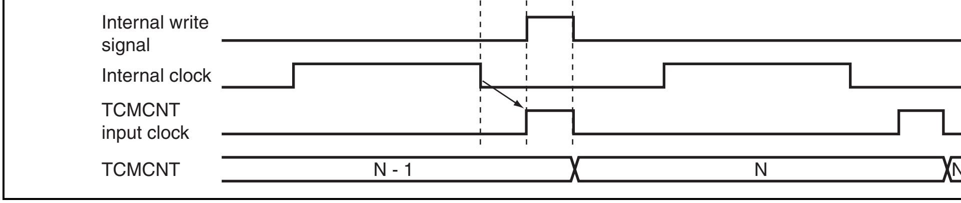


Figure 11.13 Conflict between TCMCNT Write and Count-Up Operatio

11.6.2 Conflict between TCMMLCM Write and Compare Match

When a conflict between TCMMLCM write and a compare match should occur in the se
of a cycle of writing to TCMMLCM, writing to TCMMLCM takes priority and the com
match signal is inhibited. Figure 11.14 shows the timing of this conflict.

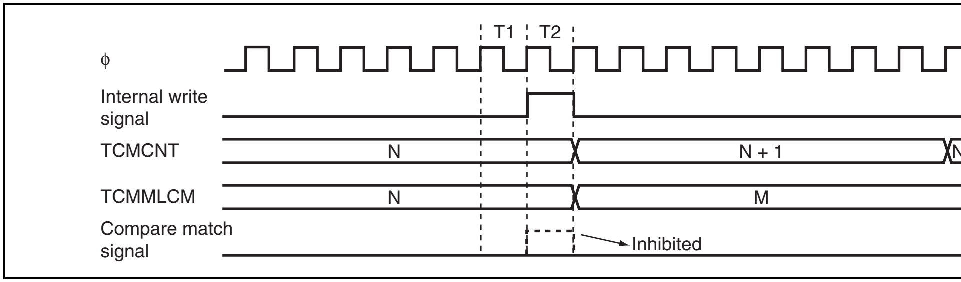


Figure 11.14 Conflict between TCMMLCM Write and Compare Match

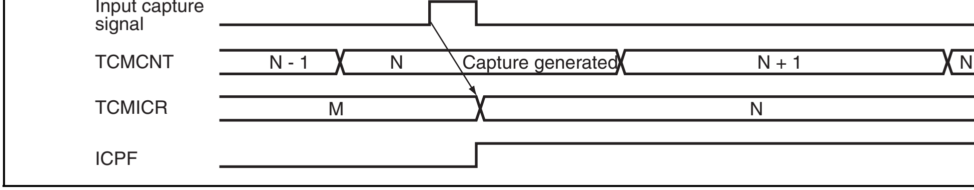


Figure 11.15 Conflict between TCMICR Read and Input Capture

11.6.4 Conflict between Edge Detection in Cycle Measurement Mode and Writing TCMMLCM or TCMMINCM

If the selected edge of TCMCYI is detected in the second half of a cycle of writing to the (TCMMLCM or TCMMINCM) in cycle measurement mode, the detected edge signal is by one cycle of the system clock (ϕ).

Figure 11.16 shows the timing of this conflict.

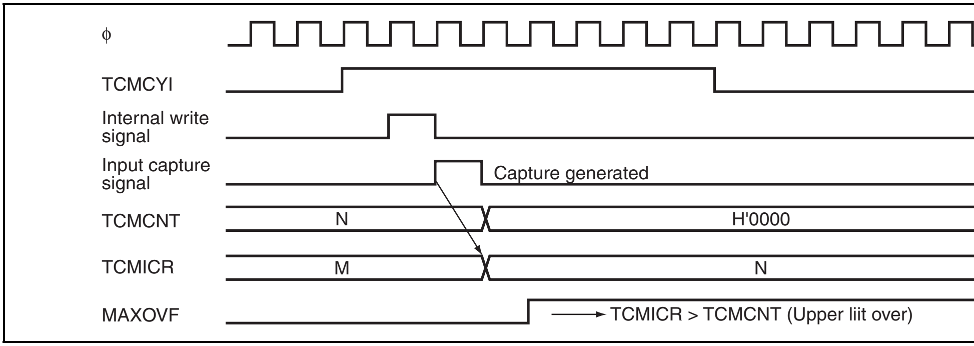


Figure 11.16 Conflict between Edge Detection and Register Write (Cycle Measurement Mode)

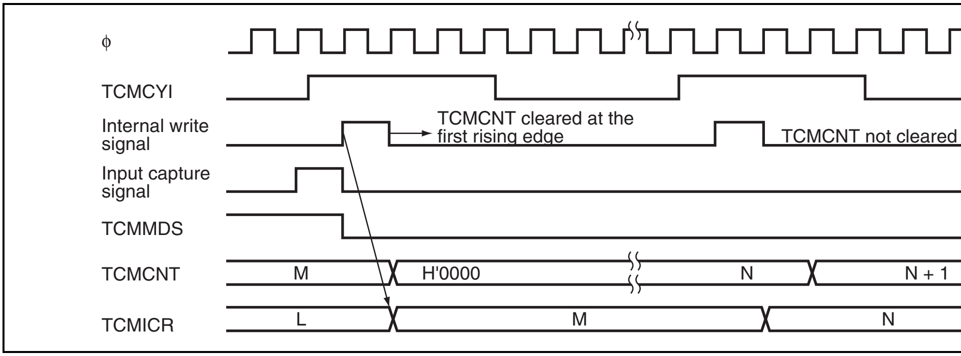


Figure 11.17 Conflict between Edge Detection and Clearing of TCMMD
(to Switch from Cycle Measurement Mode to Timer Mode)

11.6.6 Settings of TCMCKI and TCMMCI

TCMCKI and TCMMCI are multiplexed on the same pin of this LSI. Therefore, the sele external clock and the TCMMCI signal cannot be used at the same time. Do not make th CKS2 to CKS0 = B'111 and CMMS = B'1.

11.6.7 Setting for Module Stop Mode

The module-stop control register can be used to select either continuation or stoppage of operation in module-stopped mode. The default setting is for TCM operation to stop. TC registers become accessible on release from module stop mode. For details, see section 2 Down Modes.

- 16-bit resolution
- Selectable counter clock

  Any of seven internal clocks or an external clock can be selected.
- Seven interrupt sources

  Counter overflow

  Cycle upper limit overflow

  Cycle lower limit underflow

  Pulse width upper limit overflow

  Pulse width lower limit underflow

  Compare match

  Generation of input capture

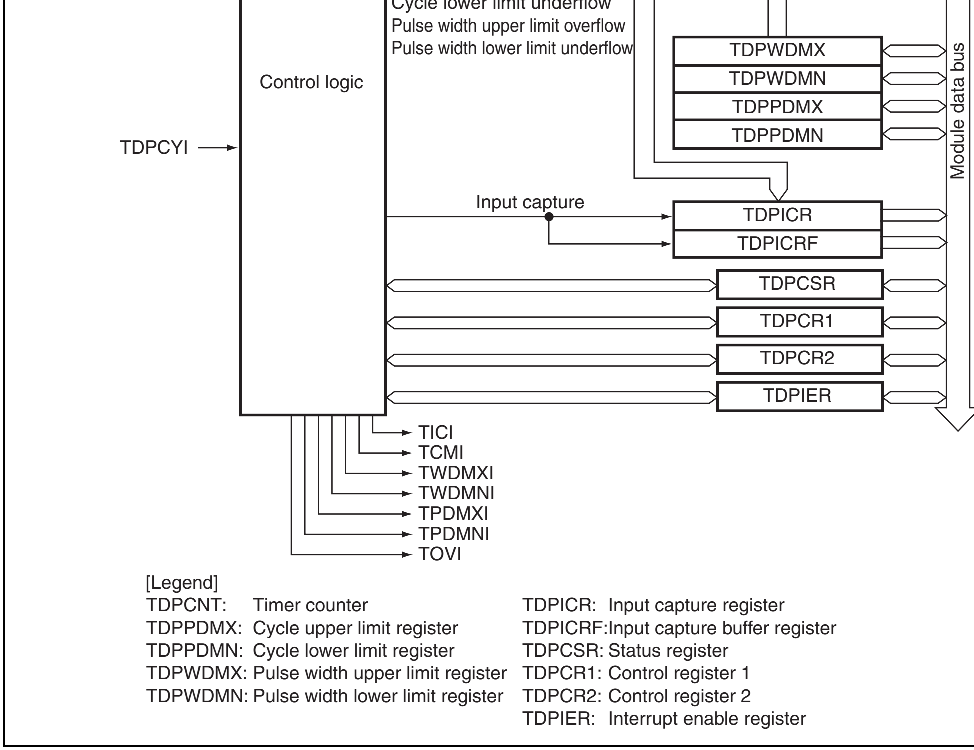

Cycle lower limit underflow
Pulse width upper limit overflow
Pulse width lower limit underflow
Control logic
TDPCYI
TDPWDMX
TDPWDMN
TDPPDMX
TDPPDMN
Module data bus
Input capture
TDPICR
TDPICRF
TDPCSR
TDPCR1
TDPCR2
TDPIER
TICI
TCMI
TWDMXI
TWDMNI
TPDMXI
TPDMNI
TOVI
[Legend]
TDPCNT: Timer counter
TDPPDMX: Cycle upper limit register
TDPPDMN: Cycle lower limit register
TDPWDMX: Pulse width upper limit register
TDPWDMN: Pulse width lower limit register
TDPICR: Input capture register
TDPICRF:Input capture buffer register
TDPCSR: Status register
TDPCR1: Control register 1
TDPCR2: Control register 2
TDPIER: Interrupt enable register

Figure 12.1 Block Diagram of TDP

	(TDPMCI1)		Cycle measurement control input
	TDPCYI1	Input	External event input
2	TDPCKI2 (TDPMCI2)	Input	External counter clock input Cycle measurement control input
	TDPCYI2	Input	External event input

12.3 Register Descriptions

The TDP has the following registers.

Table 12.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address
Channel 0	TDP timer counter_0	TDPCNT_0	R/W	H'0000	H'FB40
	TDP pulse width upper limit register_0	TDPWDMX_0	R/W	H'FFFF	H'FB42
	TDP pulse width lower limit register_0	TDPWDMN_0	R/W	H'0000	H'FB44
	TDP cycle upper limit register_0	TDPPDMX_0	R/W	H'FFFF	H'FB46
	TDP cycle lower limit register_0	TDPPDMN_0	R/W	H'0000	H'FB50
	TDP input capture register_0	TDPICR_0	R	H'0000	H'FB48
	TDP input capture buffer register_0	TDPICRF_0	R	H'0000	H'FB4A
	TDP status register_0	TDPCSR_0	R/W	H'00	H'FB4C
	TDP control register1_0	TDPCR1_0	R/W	H'00	H'FB4D
	TDP control register2_0	TDPCR2_0	R/W	H'00	H'FB4F
	TDP interrupt enable register_0	TDPIER_0	R/W	H'00	H'FB4E

	TDP input capture buffer register_1	TDPICRF_1	R	H'0000	H'FB6A
	TDP status register_1	TDPCSR_1	R/W	H'00	H'FB6C
	TDP control register1_1	TDPCR1_1	R/W	H'00	H'FB6D
	TDP control register2_1	TDPCR2_1	R/W	H'00	H'FB6F
	TDP interrupt enable register_1	TDPIER_1	R/W	H'00	H'FB6E
Channel 2	TDP timer counter_2	TDPCNT_2	R/W	H'0000	H'FB80
	TDP pulse width upper limit register_2	TDPWDMX_2	R/W	H'FFFF	H'FB82
	TDP pulse width lower limit register_2	TDPWDMN_2	R/W	H'0000	H'FB84
	TDP cycle upper limit register_2	TDPPDMX_2	R/W	H'FFFF	H'FB86
	TDP cycle lower limit register_2	TDPPDMN_2	R/W	H'0000	H'FB90
	TDP input capture register_2	TDPICR_2	R	H'0000	H'FB88
	TDP input capture buffer register_2	TDPICRF_2	R	H'0000	H'FB8A
	TDP status register_2	TDPCSR_2	R/W	H'00	H'FB8C
	TDP control register 1_2	TDPCR1_2	R/W	H'00	H'FB8D
	TDP control register 2_2	TDPCR2_2	R/W	H'00	H'FB8F
	TDP interrupt enable register_2	TDPIER_2	R/W	H'00	H'FB8E

In timer mode, TDPCNT is always writable. In cycle measurement mode, TDPCNT can
modified. TDPCNT must always be accessed in 16-bit units and cannot be accessed in 8
TDPCNT is initialized to H'0000.

12.3.2 TDP Pulse Width Upper Limit Register (TDPWDMX)

TDPWDMX is a 16-bit readable/writable register. When the TDPMDS bit in TDPCR1 i (timer mode), TDPWDMX is available as a compare match register. When the TDPMD TDPCR1 is set to 1 (cycle measurement mode), TDPWDMX is available as a pulse wid limit register.

In timer mode, the TDPWDMX value is continually compared with the TDPCNT value values match, the CMF flag in TDPCSR is set to 1. Note, however, that comparison is d the second half of a write cycle to TDPWDMX.

In cycle measurement mode, TDPWDMX can be used to set the upper limit value of the measurement pulse width. When the second edge (the second edge of this period) of the measurement period is detected, the TDPCNT value is transferred to TDPICR and the v TDPICR and TDPWDMX are compared. If the TDPICR value is greater than the TDPV value, the TWDMXOVF flag in TDPCSR is set to 1. TDPWDMX must always be acces bit units and cannot be accessed in 8-bit units. TDPWDMX is initialized to H'FFFF.

12.3.4 TDP Cycle Upper Limit Register (TDPPDMX)

TDPPDMX is a 16-bit readable/writable register. When the TDPMDS bit in TDPCR1 is s (cycle measurement mode), TDPPDMX is available as a cycle upper limit register.

In cycle measurement mode, TDPPDMX can be used to set the upper limit value of meas period. When the third edge (the first edge of the next period) of the measurement period detected, the TDPCNT value is transferred to TDPICR and the values of TDPICR and TD are compared. If the TDPICR value is greater than the TDPPDMX value, the TPDMXOV TDPCSR is set to 1. TDPPDMX must always be accessed in 16-bit units and cannot be a in 8-bit units. TDPPDMX is initialized to H'FFFF.

12.3.5 TDP Cycle Lower Limit Register (TDPPDMN)

TDPPDMN is a 16-bit readable/writable register. When the TDPMDS bit in TDPCR1 is s (cycle measurement mode), TDPPDMN is available as a cycle lower limit register.

In cycle measurement mode, TDPPDMN can be used to set the lower limit value of meas period. When the third edge (the first edge of the next period) of the measurement period detected, the TDPCNT value is transferred to TDPICR and the values of TDPICR and TD are compared. If the TDPICR value is less than the TDPPDMN value, the TPDMNUDF f TDPCSR is set to 1. TDPPDMN must always be accessed in 16-bit units and cannot be a in 8-bit units. TDPPDMN is initialized to H'0000.

TDPICRF is a 16-bit read-only register. TDPICRF can be used as a TDPICR buffer regi
When input capture occurs, the TDPICR value is transferred to TDPICRF.

TDPICRF must always be accessed in 16-bit units and cannot be accessed in 8-bit units.
TDPICRF is initialized to H'0000.

12.3.8 TDP Status Register (TDPCSR)

TDPCSR indicates the status flags and selects the external clock edge.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Timer Overflow This flag indicates a TDPCNT overflow. [Setting condition] • TDPCNT overflow (H'FFFF changes to H'000 [Clearing condition] • Reading OVF when OVF = 1 and then writing
6	TWDMXOVF	0	R/(W)*	Pulse Width Upper Limit Overflow This flag indicates that the waveform pulse width in cycle measurement mode has exceeded the u specified in TDPWDMX. [Setting condition] • When TDPICR is greater than TDPWDMX [Clearing condition] • Reading TWDMXOVF when TWDMXOVF = 1 writing 0 to TWDMXOVF

				writing 0 to TWDMNUDF
4	TPDMXOVF	0	R/(W)*	Cycle Upper Limit Overflow This flag indicates that the waveform period measu cycle measurement mode has exceeded the uppe specified in TDPPDMX. [Setting condition] • When TDPICR is greater than TDPPDMX [Clearing condition] • Reading TPDMXOVF when TPDMXOVF = 1 a writing 0 to TPDMXOVF
3	ICPF	0	R/(W)*	Input Capture Generation In timer mode, this flag indicates that the value in was transferred to TDPICR when an input captur was generated. This flag is set when the input ca signal selected by the IEDG bit is generated on th TDPCYI input pin. In cycle measurement mode, this flag indicates th value in TDPCNT was transferred to TDPICR wh rising or falling edge of the PWM waveform was d [Setting condition] • When an input capture signal is generated [Clearing condition] • Reading ICPF when ICPF = 1 and then writin ICPF

				TDPCNT value matches the TDPWDMX v is not set to 1.
1	CKSEG	0	R/(W)*	External Clock Edge Select When CKS2 to CKS0 in TDPCR1 are set to B'1 (external clock), this bit selects the edge for cou external count clock edges. 0: Falling edges of the external clock are counte 1: Rising edges of the external clock are counte
0	TPDMNUDF	0	R/(W)*	Cycle Lower Limit Underflow This flag indicates that the waveform period me cycle measurement mode is below the lower lim in TDPPDMN. [Setting condition] • When TDPICR is less than TDPPDMN [Clearing condition] • Reading TPDMNUDF when TPDMNUDF = writing 0 to TPDMNUDF

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

				capture operation stops. Clear this bit to initialize TDPCNT to H'0000 befor to cycle measurement mode.
6	POCTL	0	R/W	TDPCYI Input Polarity Inversion 0: TDPCYI input is used directly 1: TDPCYI input is inverted for use Note: Change this bit when CST = 0 and TDPMDS
5	CPSPE	0	R/W	Input Capture Stop Enable Controls whether counting up by TDPCNT and inp capture operation stop or continue when any of th TPDMXOVF, TPDMNUDF, TWDMXOVF, and TW flags is set to 1 in cycle measurement mode. This not affect operation in timer mode. 0: Counting up and input-capture operation contin any of the flags is set to 1. 1: Counting up and input-capture operation stop w of the flags is set to 1.

				1: The rising edge of TDPCYI input is selected When POCTL = 1 0: The rising edge of TDPCYI input is selected 1: The falling edge of TDPCYI input is selected
3	TDPMDS	0	R/W	TDP Mode Select Selects the TDP operating mode. 0: Timer mode In timer mode, the operating mode is input cap compare match. 1: Cycle measurement mode Setting this bit to 1 starts counting by TDPCNT CST bit in TDPCR1 to initialize TDPCNT to H'(before setting cycle measurement mode.
2	CKS2	0	R/W	Clock Select 2, 1, 0
1	CKS1	0	R/W	These bits select the clock signal for input to TDF not select the external clock in level control meas mode.
0	CKS0	0	R/W	000: Counts the ϕ internal clock 001: Counts the ϕ/2 internal clock 010: Counts the ϕ/4 internal clock 011: Counts the ϕ/8 internal clock 100: Counts the ϕ/16 internal clock 101: Counts the ϕ/32 internal clock 110: Counts the ϕ/64 internal clock 111: Counts the external clock (Select the external clock edge with CKSEG in T Note: Change this bit when CST = 0 and TDPMD

				1: The TDPMCI signal is used (cycle measuremen performed only while the TDPMCI signal is high Note: Change this bit when CST = 0 and TDPMDS
6	MCICTL	0	R/W	TDPMCI Input Polarity Inversion 0: TDPMCI input is used directly 1: TDPMCI input is inverted for use Note: Change this bit when CST = 0 and TDPMDS
5 to 1	—	0	R/W	Reserved The initial value should not be changed.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modifie

				0: OVF interrupt requests are disabled 1: OVF interrupt requests are enabled
6	TWDMXIE	0	R/W	Pulse Width Upper Limit Overflow Interrupt Enab Enables or disables the issuing of TWDMXOVF i requests when the TWDMXOVF flag in TDPCSR 0: TWDMXOVF interrupt requests are disabled 1: TWDMXOVF interrupt requests are enabled
5	TWDMNIE	0	R/W	Pulse Width Lower Limit Underflow Interrupt Ena Enables or disables the issuing of TWDMNUDF i requests when the TWDMNUDF flag in TDPCSR 0: TWDMNUDF interrupt requests are disabled 1: TWDMNUDF interrupt requests are enabled
4	TPDMXIE	0	R/W	Cycle Upper Limit Overflow Interrupt Enable Enables or disables the issuing of TPDMXOVF ir requests when the TPDMXOVF flag in TDPCSR 0: TPDMXOVF interrupt requests are disabled 1: TPDMXOVF interrupt requests are enabled
3	ICPIE	0	R/W	Input Capture Interrupt Enable Enables or disables the issuing of ICPF interrupt when the ICPF flag in TDPCSR is set to 1. 0: ICPF interrupt requests are disabled 1: ICPF interrupt requests are enabled
2	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables the issuing of CMF interrupt when the CMF flag in TDPCSR is set to 1. 0: CMF interrupt requests are disabled 1: CMF interrupt requests are enabled

Enables or disables the issuing of TPDMNUDF int requests when the TPDMNUDF flag in TDPCSR is

0: TPDMNUDF interrupt requests are disabled

1: TPDMNUDF interrupt requests are enabled

The TDP operates as a free-running counter in timer mode. The TDP starts counting up
CST bit in TDPCR1 is set to 1. When TDPCNT overflows (H'FFFF changes to H'0000)
bit in TDPCSR is set to 1 and an interrupt request is generated if the OVIE bit in TDPIE
Figure 12.2 shows an example of free-running counter operation. In addition, figure 12.
TDPCNT count timing for external clock operation. Note that the external clock require
width of at least 1.5 cycles. The counter will not operate correctly if the pulses are narro
this.

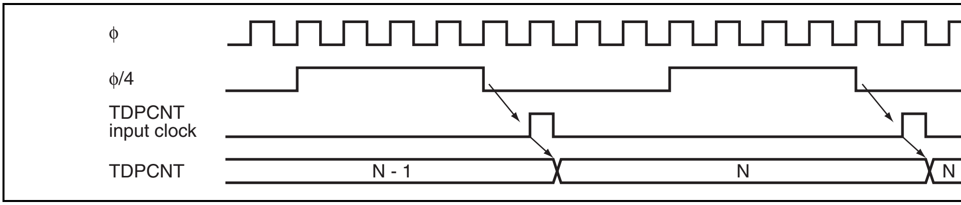


Figure 12.2 Example of Free-Running Counter Operation

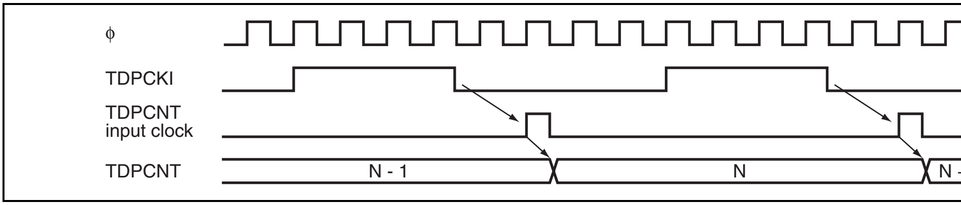


Figure 12.3 Count Timing of External Clock Operation (Falling Edges)

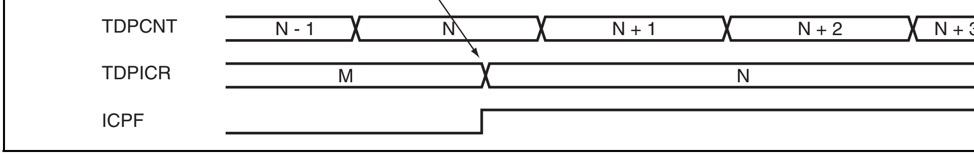


Figure 12.4 Example of Input Capture Operation Timing (Selection of Rising E

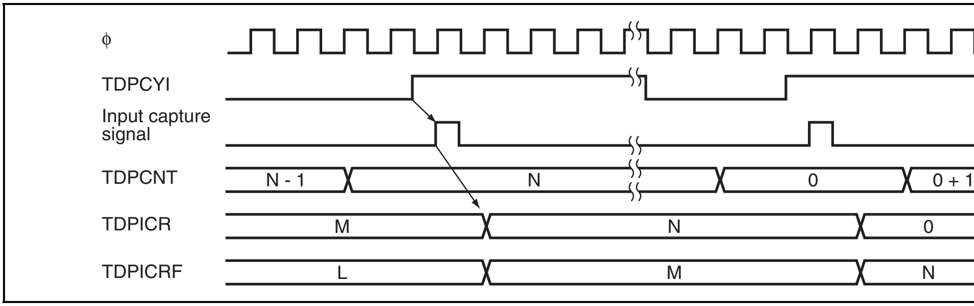


Figure 12.5 Example of Buffer Operation for Input Capture

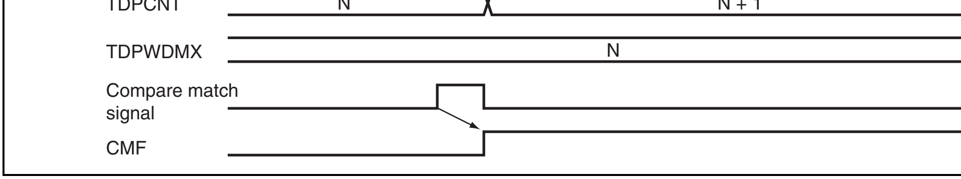


Figure 12.6 Timing of CMF Flag Setting on Compare Match

12.4.2 Cycle Measurement Mode

The TDP operates in cycle measurement mode when the TDPMDS bit in TDPCR1 is se

(1) Counter Operation

TDPCNT counts up in cycle measurement mode regardless of the setting of the CST bit TDPCR1. TDPCNT is cleared to H'0000 when the first edge in the measurement period detected, from which state it counts up. Figure 12.7 shows an example of counter operat cycle measurement mode.

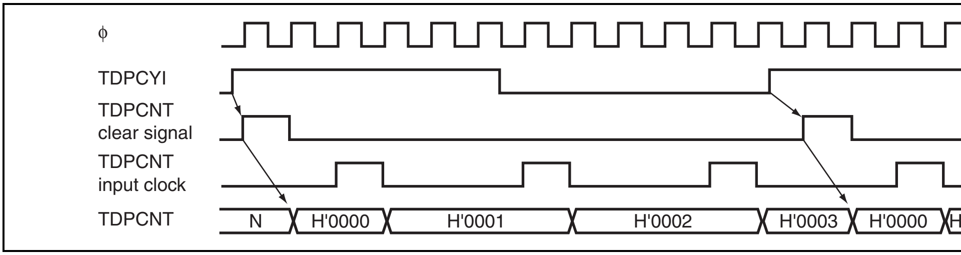


Figure 12.7 Example of Counter Operation in Cycle Measurement Mod

values in TDPWDMX and TDPWDMN. If TDPIR is greater than TDPWDMX or less tha TDPWDMN, the TWDMXOVF or TWDMNUDF flag, respectively, in TDPCSR is set to When the third edge is detected, the value in TDPCNT is transferred to TDPICR. At this value in TDPICR is compared with the values in TDPPDMX and TDPPDMN. If TDPIC greater than TDPPDMX or less than TDPPDMN, the TPDMXOVF or TPDMNUDF flag respectively, in TDPCSR is set to 1. Generation of the corresponding interrupt request is by the setting in TDPIER. Also, when the third edge is detected, TDPCNT is cleared to H and the next round of measurement starts.

When the CPSPE bit in TDPCR1 is cleared to 0, the next round of cycle measurement wi regardless of whether any of these flags is set to 1.

If any of these flags is set to 1 while the CPSPE bit in TDPCR1 is set to 1, counting up by TDPCNT stops and cycle measurement also stops. Subsequently clearing the correspondi to 0 automatically clears TDPCNT to H'0000, and counting up for cycle measurement is r

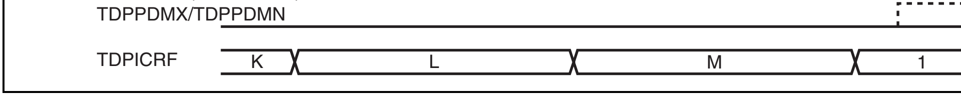


Figure 12.8 Example of Timing in Cycle Measurement

When the PMMS bit in TDPCR2 is set to 1, cycle measurement is performed only while
TDPMCI signal is high. Figure 12.9 shows an example of timing in cycle measurement
PMMS bit is set to 1.

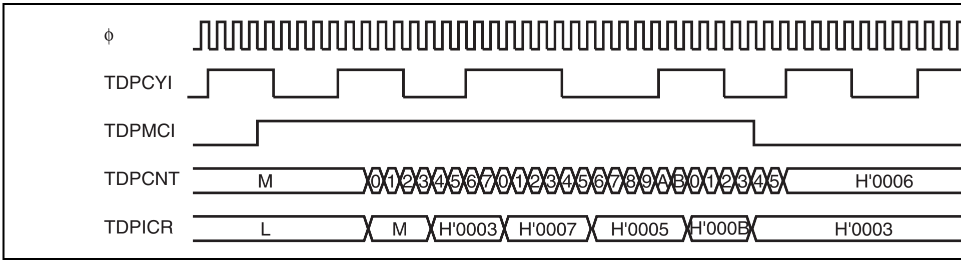


Figure 12.9 Example of Timing in Cycle Measurement (PMMS Bit = 1).

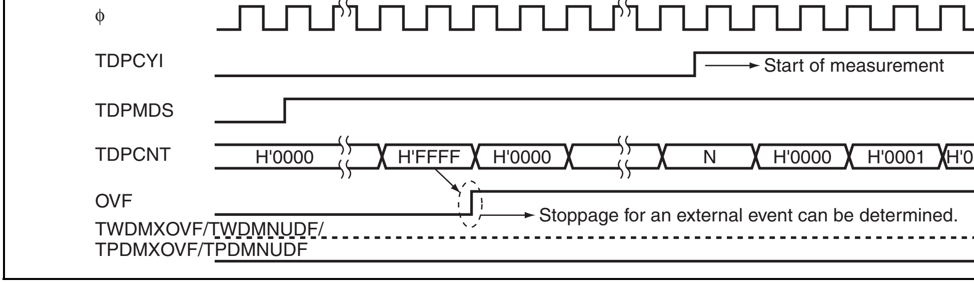


Figure 12.10 Example of Timing for Stoppage for an External Event (1)

When any of the TWDMXOVF, TWDMNUDF, TPDMXOVF, and TPDMNUDF flags is while the CPSPE bit in TDPCR1 is 1, cycle measurement stops. Thereafter, when the corresponding flag is cleared, cycle measurement restarts. When the timer overflows befo first edge is detected after the restart of cycle measurement, it is possible to determine sto an external event.

Figure 12.11 shows an example of the timing for this type of stoppage for an external eve

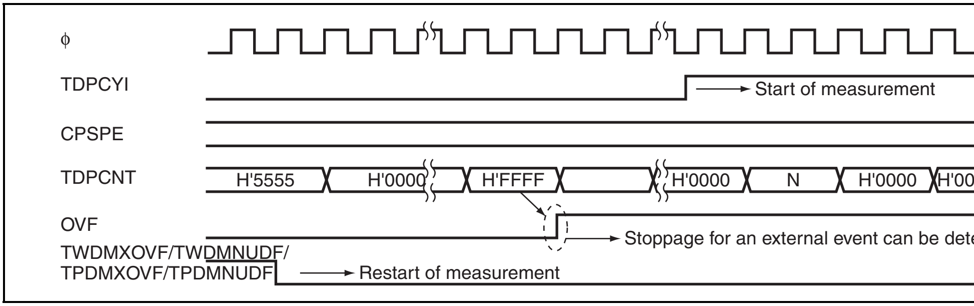
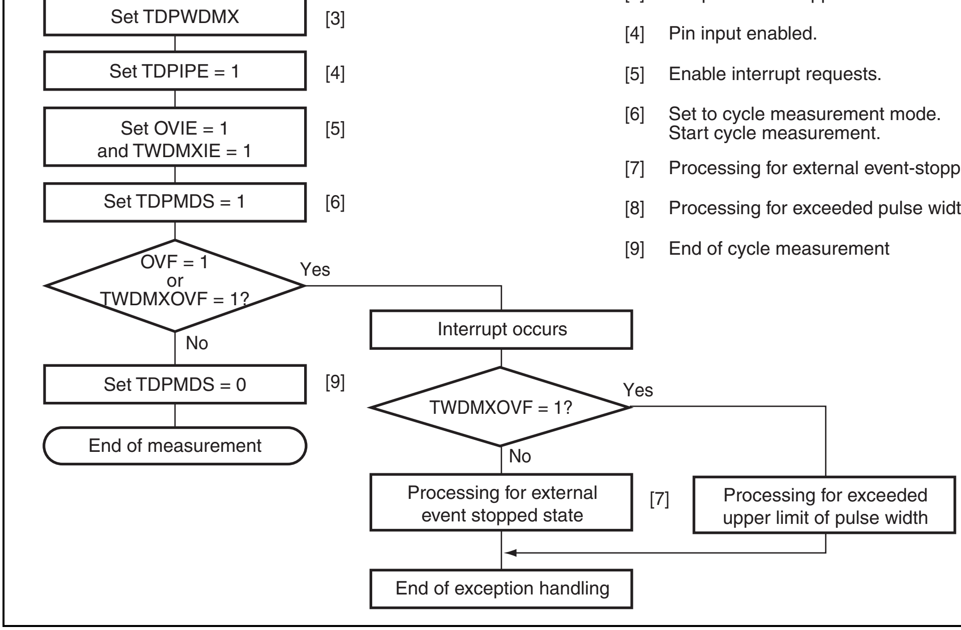


Figure 12.11 Example of Timing for Stoppage for an External Event (2)



[4] Pin input enabled.

[5] Enable interrupt requests.

[6] Set to cycle measurement mode.
Start cycle measurement.

[7] Processing for external event-stopp

[8] Processing for exceeded pulse widt

[9] End of cycle measurement

Figure 12.12 Example of Cycle Measurement Mode Settings (for Pulse Width Upper-Limit Value)

	TCMI0	TDPWDMX_0 compare match	CMF_0
	TWDMXI0	TDPWDMX_0 overflow	TWDMXOVF_0
	TWDMNI0	TDPWDMN_0 underflow	TWDMNUDF_0
	TPDMXI0	TDPPDMX_0 overflow	TPDMXOVF_0
	TPDMNI0	TDPPDMN_0 underflow	TPDMNUDF_0
	TOVI0	TDPCNT_0 overflow	OVF_0
TDP_1	TICI1	TDPICR_1 input capture	ICPF_1
	TCMI1	TDPWDMX_1 compare match	CMF_1
	TWDMXI1	TDPWDMX_1 overflow	TWDMXOVF_1
	TWDMNI1	TDPWDMN_1 underflow	TWDMNUDF_1
	TPDMXI1	TDPPDMX_1 overflow	TPDMXOVF_1
	TPDMNI1	TDPPDMN_1 underflow	TPDMNUDF_1
	TOVI1	TDPCNT_1 overflow	OVF_1
TDP_2	TICI2	TDPICR_1 input capture	ICPF_2
	TCMI2	TDPWDMX_2 compare match	CMF_2
	TWDMXI2	TDPWDMX_2 overflow	TWDMXOVF_2
	TWDMNI2	TDPWDMN_2 underflow	TWDMNUDF_2
	TPDMXI2	TDPPDMX_2 overflow	TPDMXOVF_2
	TPDMNI2	TDPPDMN_2 underflow	TPDMNUDF_2
	TOVI2	TDPCNT_2 overflow	OVF_2

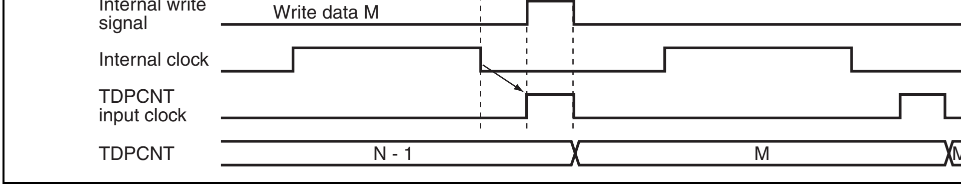


Figure 12.13 Conflict between TDPCNT Write and Counting Up

12.6.2 Conflict between TDPPDMX Write and Compare Match

If a conflict between a TDPPDMX write and a compare match occurs in the second half TDPPDMX write cycle, writing to TDPPDMX takes precedence and the compare match inhibited. Figure 12.14 shows the timing of this conflict.

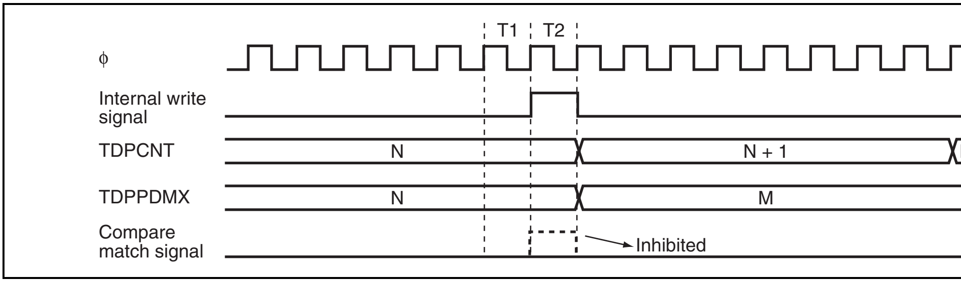


Figure 12.14 Conflict between TDPPDMX Write and Compare Match

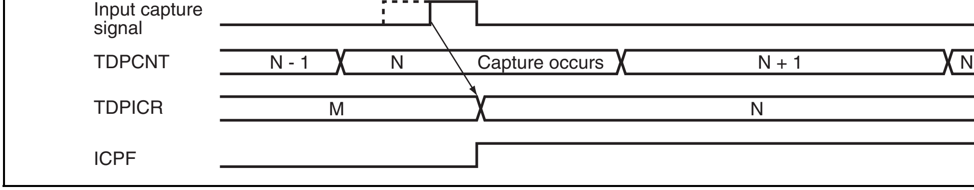


Figure 12.15 Conflict between Input Capture and TDPICR Read

12.6.4 Conflict between Edge Detection in Cycle Measurement Mode and Writing Upper Limit or Lower Limit Register

If the edge of TDPCYI is detected in the second half of a cycle of writing to any of the up limit/lower limit registers (TDPPDMX, TDPPDMN, TDPWDMX, and TDPWDMN) in c measurement mode, the detected edge signal is delayed by one cycle of the system clock

Figure 12.16 shows the timing of this conflict.

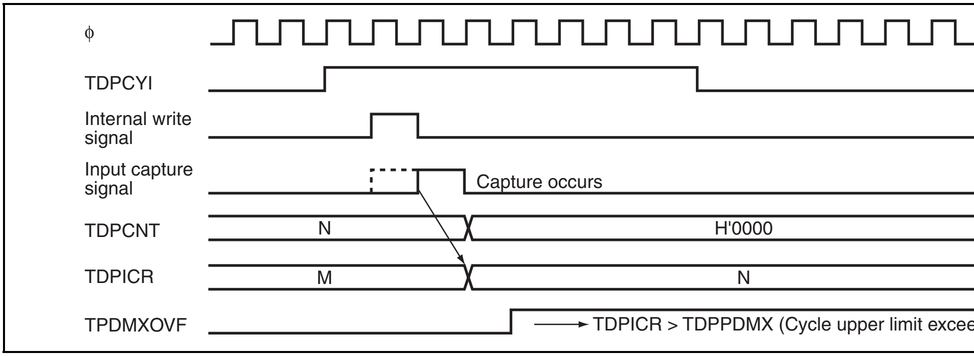


Figure 12.16 Conflict between Edge Detection and Register Write (Cycle Measurement Mode)

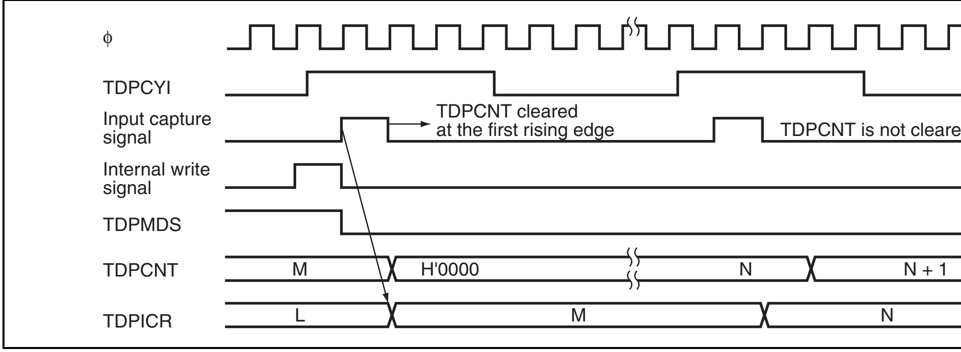


Figure 12.17 Conflict between Edge Detection and TDPMDS Bit Clearing (In Sv from Cycle Measurement Mode to Timer Mode)

12.6.6 Settings for TDPCKI and TDPMCI

TDPCKI and TDPMCI are multiplexed on the same pin of this LSI. Therefore, the selec external clock and the TDPMCI signal cannot be used at the same time. Do not make th CKS2 to CKS0 = B'111 and PMMS = B'1.

12.6.7 Setting for Module Stop Mode

The module-stop control register can be used to specify whether to continue or stop TDI operation. The default setting is for the TDP operation to stop. The TDP registers becom accessible on release from module stop mode. For details, see section 26, Power-Down I

- Selection of clock sources

  The counter input clock can be selected from six internal clocks and an external cloc

- Selection of three ways to clear the counters

  The counters can be cleared on compare-match A, compare-match B, or by an extern signal.

- Timer output controlled by two compare-match signals

  The timer output signal in each channel is controlled by two independent compare-m signals, enabling the timer to be used for various applications, such as the generation output or PWM output with an arbitrary duty cycle.

- Cascading of two channels

  — Cascading of TMR_0 and TMR_1

  Operation as a 16-bit timer can be performed using TMR_0 as the upper half and as the lower half (16-bit count mode).

  TMR_1 can be used to count TMR_0 compare-match occurrences (compare-mat mode).

  — Cascading of TMR_Y and TMR_X

  Operation as a 16-bit timer can be performed using TMR_Y as the upper half and as the lower half (16-bit count mode).

  TMR_X can be used to count TMR_Y compare-match occurrences (compare-ma mode).

- Multiple interrupt sources for each channel

  TMR_0, TMR_1, and TMR_Y: Three types of interrupts: Compare-match A, compa match B, and overflow

  TMR_X: Four types of interrupts: Compare-match A, compare match B, overflow, and input capture

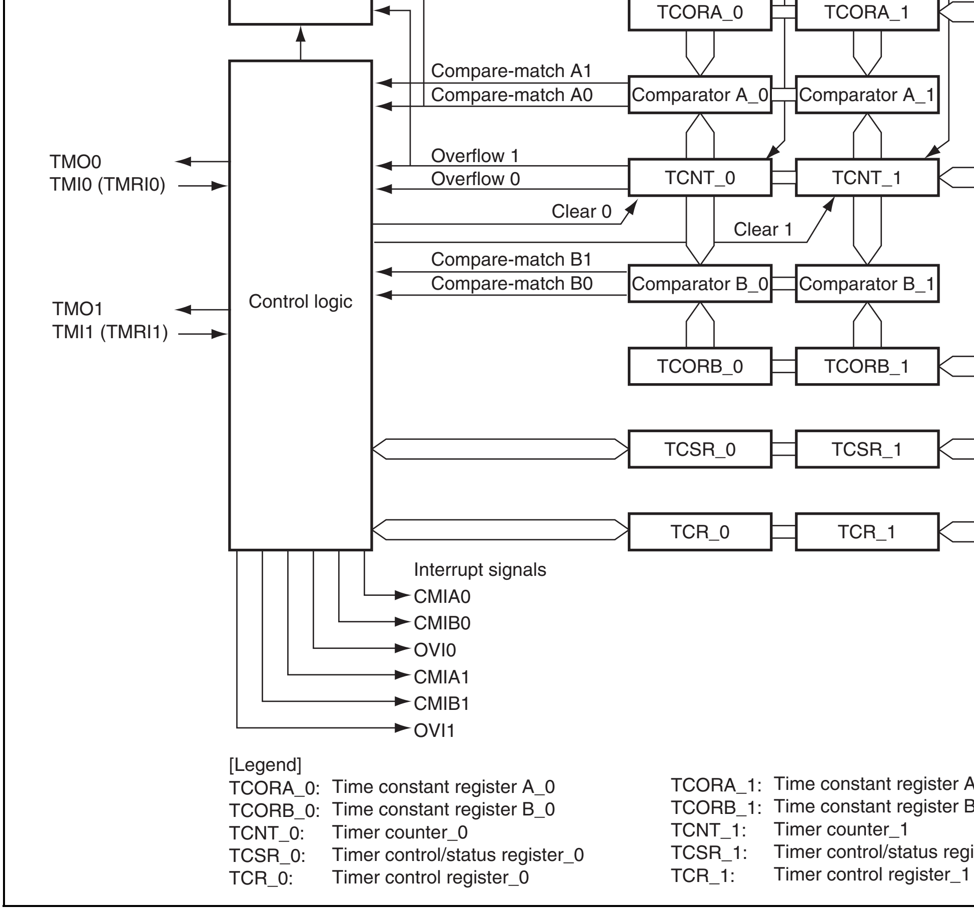
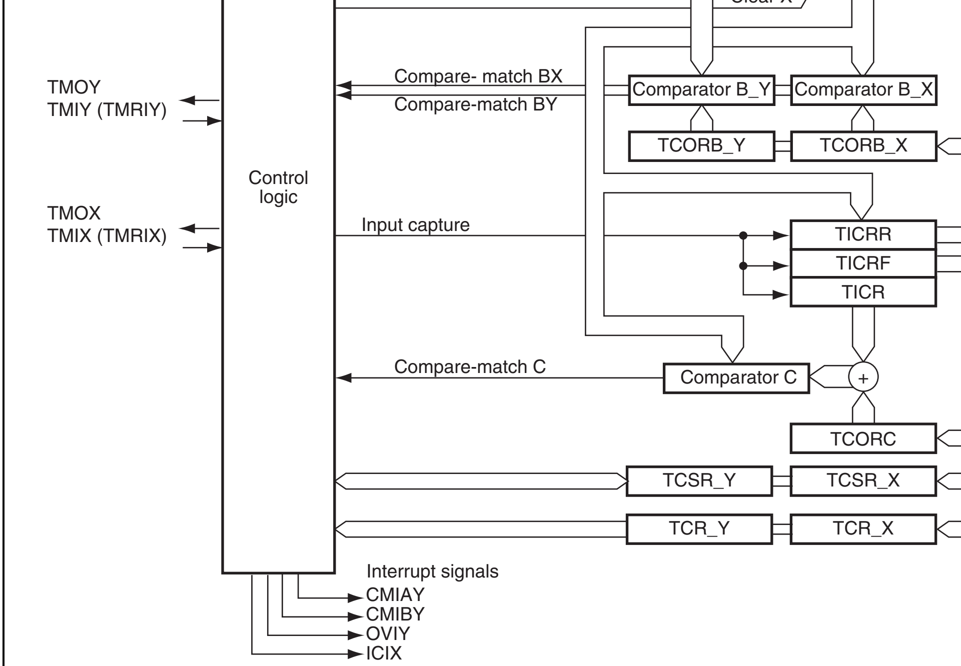


Figure 13.1 Block Diagram of 8-Bit Timer (TMR_0 and TMR_1)



[Legend]

TCORA_Y:	Time constant register A_Y	TCNT_X:	Timer counter_X
TCORB_Y:	Time constant register B_Y	TCSR_X:	Timer control/status register_X
TCNT_Y:	Timer counter_Y	TCR_X:	Timer control register_X
TCSR_Y:	Timer control/status register_Y	TICR:	Input capture register
TCR_Y:	Timer control register_Y	TCORC:	Time constant register C
TCORA_X:	Time constant register A_X	TICRR:	Input capture register R
TCORB_X:	Time constant register B_X	TICRF:	Input capture register F

Figure 13.2 Block Diagram of 8-Bit Timer (TMR_Y and TMR_X)

	TMI1 (TMCI1/TMRI1)	Input	External clock input/external reset inpu counter
TMR_Y	TMIY (TMCIY/TMRIY)	Input	External clock input/external reset inpu counter
	TMOY	Output	Output controlled by compare-match
TMR_X	TMOX	Output	Output controlled by compare-match
	TMIX (TMCIX/TMRIX)	Input	External clock input/external reset inpu counter

	Time constant register A_0	TCORA_0	R/W	H'FF	H'FFCC
	Time constant register B_0	TCORB_0	R/W	H'FF	H'FFCE
	Timer control register_0	TCR_0	R/W	H'00	H'FFC8
	Timer control/status register_0	TCSR_0	R/W	H'00	H'FFCA
Channel 1	Timer counter_1	TCNT_1	R/W	H'00	H'FFD1
	Time constant register A_1	TCORA_1	R/W	H'FF	H'FFCD
	Time constant register B_1	TCORB_1	R/W	H'FF	H'FFCF
	Timer control register_1	TCR_1	R/W	H'00	H'FFC9
	Timer control/status register_1	TCSR_1	R/W	H'10	H'FFCB
Channel Y	Timer counter_Y	TCNT_Y	R/W	H'00	H'FFF4 H'FECC*
	Time constant register A_Y	TCORA_Y	R/W	H'FF	H'FFF2 H'FECA*
	Time constant register B_Y	TCORB_Y	R/W	H'FF	H'FFF3 H'FECB*
	Timer control register_Y	TCR_Y	R/W	H'00	H'FFF0 H'FEC8*
	Timer control/status register_Y	TCSR_Y	R/W	H'00	H'FFF1 H'FEC9*
	Timer connection register S	TCONRS	R/W	H'00	H'FFFE

Note: * Upper address: when RELOCATE = 0
Lower address: when RELOCATE = 1

	Input capture register F	TICRF	R	H'00	H'FFF3	8
	Timer connection register I	TCONRI	R/W	H'00	H'FFFC	8
Common	Timer XY control register	TCRXY	R/W	H'00	H'FEC6	8

Note: Some of the registers of TMR_X and TMR_Y use the same address. The registers switched by the TMRX/Y bit in TCONRS.
TCNT_Y, TCORA_Y, TCORB_Y, and TCR_Y can be accessed when the RELOC SYSCR3 and the KINWUE bit in SYSCR are cleared to 0 and the TMRX/Y bit in T is set to 1, or when the RELOCATE bit in SYSCR3 is set to 1. TCNT_X, TCORA_ TCORB_X, and TCR_X can be accessed when the RELOCATE bit in SYSCR3, th KINWUE bit in SYSCR, and the TMRX/Y bit in TCONRS are cleared to 0, or when RELOCATE bit in SYSCR3 is set to 1.

13.3.1 Timer Counter (TCNT)

Each TCNT is an 8-bit readable/writable up-counter. TCNT_0 and TCNT_1 (or TCNT_ TCNT_Y) comprise a single 16-bit register, so they can be accessed together by word ac clock source is selected by the CKS2 to CKS0 bits in TCR. TCNT can be cleared by an e reset input signal, compare-match A signal or compare-match B signal. The method of cl can be selected by the CCLR1 and CCLR0 bits in TCR. When TCNT overflows (change H'FF to H'00), the OVF bit in TCSR is set to 1. TCNT is initialized to H'00.

13.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_1 (or TCORB_X TCORB_Y) comprise a single 16-bit register, so they can be accessed together by word TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag B (CMFB) in TCSR is set to 1. Note however that co is disabled during the T_2 state of a TCORB write cycle. The timer output from the TMO freely controlled by these compare-match B signals and the settings of output select bits OS2 in TCSR. TCORB is initialized to H'FF.

				0: CMFB interrupt request (CMIB) is disabled 1: CMFB interrupt request (CMIB) is enabled
6	CMIEA	0	R/W	Compare-Match Interrupt Enable A Selects whether the CMFA interrupt request (C enabled or disabled when the CMFA flag in TC set to 1. 0: CMFA interrupt request (CMIA) is disabled 1: CMFA interrupt request (CMIA) is enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether the OVF interrupt request (OV enabled or disabled when the OVF flag in TCS to 1. 0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	These bits select the method by which the time counter is cleared. 00: Clearing is disabled 01: Cleared on compare-match A 10: Cleared on compare-match B 11: Cleared on rising edge of external reset inp
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock input to TCNT and c condition, together with the ICKS1 and ICKS0 STCR. For details, see table 13.3.
0	CKS0	0	R/W	

	0	1	0	—	1	Increments at falling edge of int clock ϕ/32
	0	1	1	—	0	Increments at falling edge of int clock ϕ/1024
	0	1	1	—	1	Increments at falling edge of int clock ϕ/256
	1	0	0	—	—	Increments at overflow signal fr TCNT_1*
TMR_1	0	0	0	—	—	Disables clock input
	0	0	1	0	—	Increments at falling edge of int clock ϕ/8
	0	0	1	1	—	Increments at falling edge of int clock ϕ/2
	0	1	0	0	—	Increments at falling edge of int clock ϕ/64
	0	1	0	1	—	Increments at falling edge of int clock ϕ/128
	0	1	1	0	—	Increments at falling edge of int clock ϕ/1024
	0	1	1	1	—	Increments at falling edge of int clock ϕ/2048
	1	0	0	—	—	Increments at compare-match A TCNT_0*

0	0	1	—	1	Increments at ϕ/4096
0	1	0	—	1	Increments at ϕ/8192
0	1	1	—	1	Increments at ϕ/16384
1	0	0	—	1	Increments at overflow signal fr TCNT_X*
1	0	1	—	x	Increments at rising edge of ext clock
1	1	0	—	x	Increments at falling edge of ex clock
1	1	1	—	x	Increments at both rising and fa edges of external clock

0	1	0	1	—	Increments at ϕ/4096
0	1	1	1	—	Increments at ϕ/8192
1	0	0	1	—	Increments at compare-match A TCNT_Y*
1	0	1	x	—	Increments at rising edge of exte clock
1	1	0	x	—	Increments at falling edge of exte clock
1	1	1	x	—	Increments at both rising and fall edges of external clock

Note: * If the TMR_Y clock input is set as the TCNT_X overflow signal and the TMR_X input is set as the TCNT_Y compare-match signal simultaneously, a count-up cannot be generated. These settings should not be made.

[Legend]

x: Don't care

—: Invalid

				[Clearing condition] Read CMFB when CMFB = 1, then write 0 in
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_0 and TCORA_0 r [Clearing condition] Read CMFA when CMFA = 1, then write 0 in
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_0 overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OV
4	ADTE	0	R/W	A/D Trigger Enable Enables or disables A/D converter start reque compare-match A. 0: A/D converter start requests by compare-m are disabled 1: A/D converter start requests by compare-m are enabled
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMO0 pin output l be changed by compare-match B of TCORB_ TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_1 and TCORB_1 m [Clearing condition] Read CMFB when CMFB = 1, then write 0 in C
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_1 and TCORA_1 m [Clearing condition] Read CMFA when CMFA = 1, then write 0 in C
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_1 overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	—	1	R	Reserved This bit is always read as 1 and cannot be mod

1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMO1 pin output l be changed by compare-match A of TCORA_ TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

- TCSR_X

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_X and TCORB_X r [Clearing condition] Read CMFB when CMFB = 1, then write 0 in
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_X and TCORA_X r [Clearing condition] Read CMFA when CMFA = 1, then write 0 in

				external reset signal in that order. [Clearing condition] Read ICF when ICF = 1, then write 0 in ICF
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMOX pin output le be changed by compare-match B of TCORB_X TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMOX pin output le be changed by compare-match A of TCORA_X TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

				[Setting condition] When the values of TCNT_Y and TCORA_Y r [Clearing condition] Read CMFA when CMFA = 1, then write 0 in
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_Y overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OV
4	ICIE	0	R/W	Input Capture Interrupt Enable Enables or disables the ICF interrupt request when the ICF bit in TCSR_X is set to 1. 0: ICF interrupt request (ICIX) is disabled 1: ICF interrupt request (ICIX) is enabled
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMOY pin output be changed by compare-match B of TCORB_ TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

13.3.6 Time Constant Register C (TCORC)

TCORC is an 8-bit readable/writable register. The sum of contents of TCORC and TICR
compared with TCNT. When a match is detected, a compare-match C signal is generated
However, comparison at the T_2 state in the write cycle to TCORC and at the input capture
TICR is disabled. TCORC is initialized to H'FF.

13.3.7 Input Capture Registers R and F (TICRR and TICRF)

TICRR and TICRF are 8-bit read-only registers. While the ICST bit in TCONRI is set to
contents of TCNT are transferred at the rising edge and falling edge of the external reset i
(TMRIX) in that order. The ICST bit is cleared to 0 when one capture operation ends. TI
TICRF are initialized to H'00.

				pulse by means of a single capture operation control of the ICST bit. When a rising edge fo a falling edge is detected on TMRIX after the is set to 1, the contents of TCNT at those poi captured into TICRR and TICRF, respectively ICST bit is cleared to 0. [Clearing condition] When a rising edge followed by a falling edge detected on TMRIX [Setting condition] When 1 is written in ICST after reading ICST
3 to 0	—	All 0	R/W	Reserved The initial values should not be modified.

13.3.9 Timer Connection Register S (TCONRS)

TCONRS selects whether to access TMR_X or TMR_Y registers.

Bit	Bit Name	Initial Value	R/W	Description
7	TMRX/Y	0	R/W	TMR_X/TMR_Y Access Select For details, see table 13.4. 0: The TMR_X registers are accessed at addr H'(FF)FFF0 to H'(FF)FFF5 1: The TMR_Y registers are accessed at addr H'(FF)FFF0 to H'(FF)FFF5
6 to 0	—	All 0	R/W	Reserved The initial values should not be modified.

TCRXY selects the TMR_X and TMR_Y output pins and internal clock.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	CKSX	0	R/W	TMR_X Clock Select For details about selection, see table 13.3.
4	CKSY	0	R/W	TMR_Y Clock Select For details about selection, see table 13.3.
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

According to the above settings, the waveforms with the TCORA cycle and TCORB pul
can be output without the intervention of software.

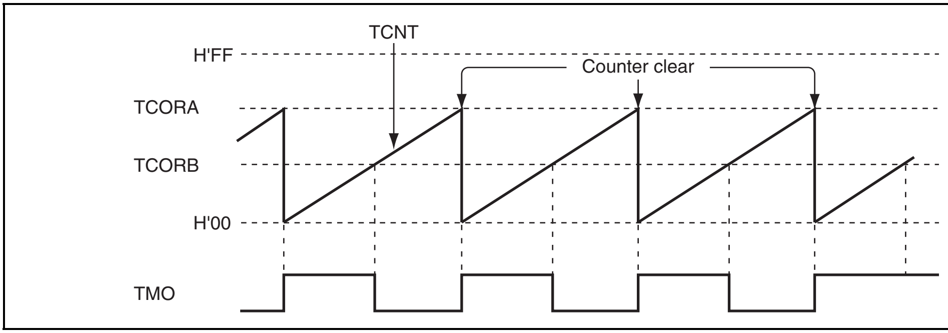


Figure 13.3 Pulse Output Example

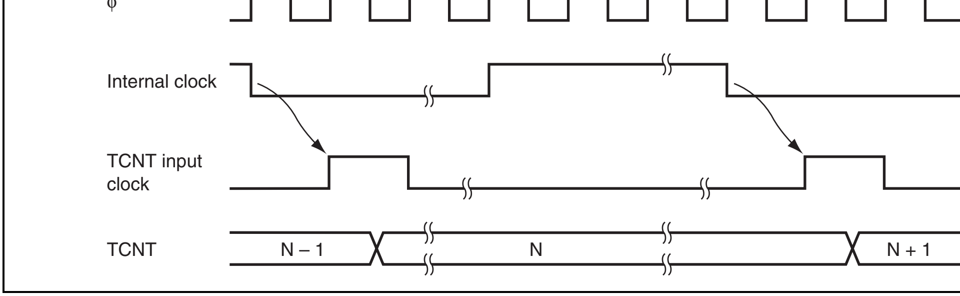


Figure 13.4 Count Timing for Internal Clock Input

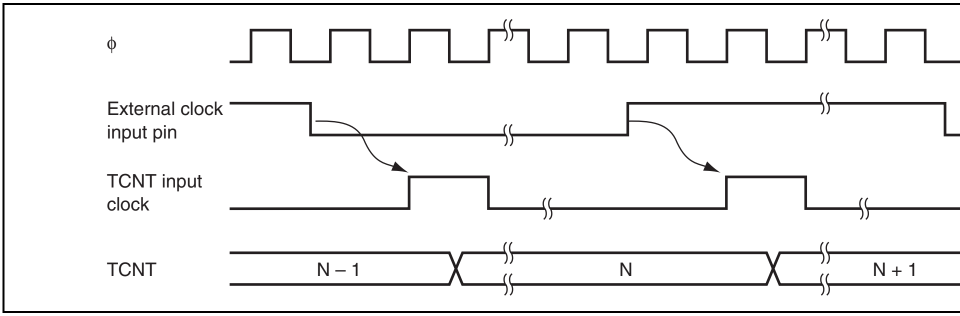


Figure 13.5 Count Timing for External Clock Input (Both Edges)

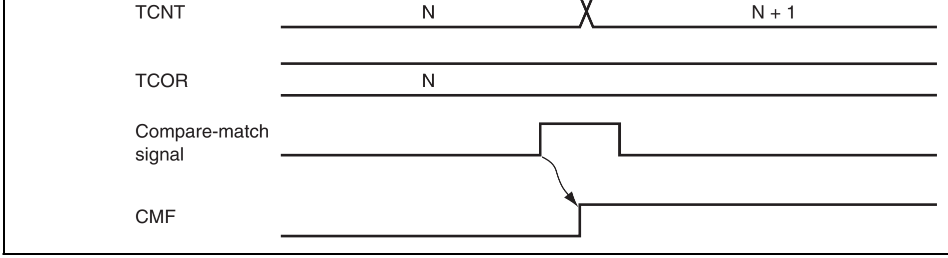


Figure 13.6   Timing of CMF Setting at Compare-Match

13.5.3 Timing of Timer Output at Compare-Match

When a compare-match signal occurs, the timer output changes as specified by the OS3 bits in TCSR. Figure 13.7 shows the timing of timer output when the output is set to tog compare-match A signal.

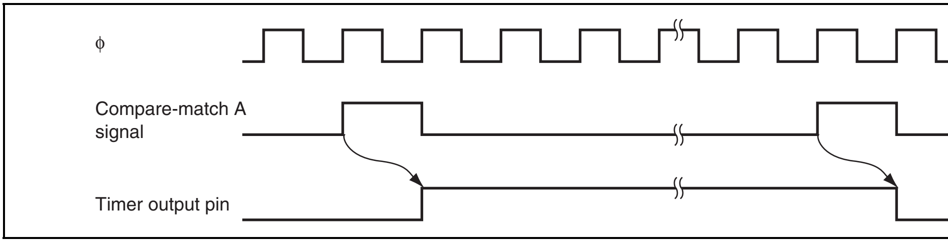


Figure 13.7   Timing of Toggled Timer Output by Compare-Match A Sign

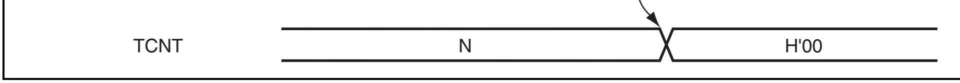


Figure 13.8   Timing of Counter Clear by Compare-Match

13.5.5 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings of CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 stat 13.9 shows the timing of clearing the counter by an external reset input.

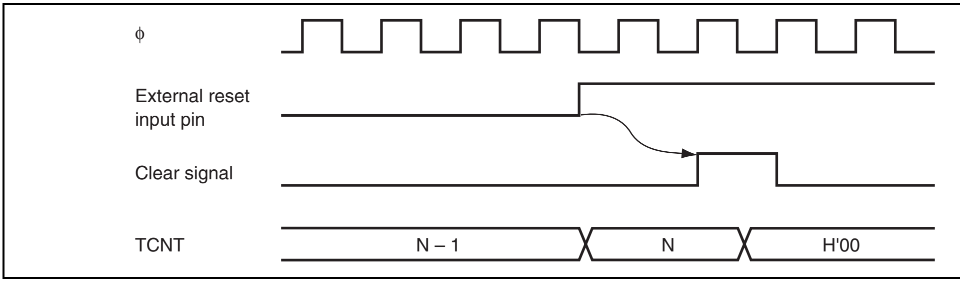


Figure 13.9   Timing of Counter Clear by External Reset Input

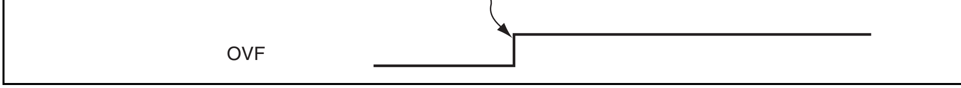


Figure 13.10 Timing of OVF Flag Setting

- Setting of compare-match flags
  - The CMF flag in TCSR_0 is set to 1 when a 16-bit compare-match occurs.
  - The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compa the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit comp match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is also cleared counter clear by the TMI0 pin has been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bi be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accord the 16-bit compare-match conditions.
  - Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accord the lower 8-bit compare-match conditions.

13.6.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B'100, TCNT_1 counts the occurrence of compa A for TMR_0. TMR_0 and TMR_1 are controlled independently. Conditions such as sett CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each or TMR_0 and TMR_1.

occupying the lower 8 bits.

- Setting of compare-match flags
  - — The CMF flag in TCSR_Y is set to 1 when an upper 8-bit compare-match occurs
  - — The CMF flag in TCSR_X is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - — If the CCLR1 and CCLR0 bits in TCR_Y have been set for counter clear at comp match, only the upper eight bits of TCNT_Y are cleared. The upper eight bits of are also cleared when counter clear by the TMRIY pin has been set.
  - — The settings of the CCLR1 and CCLR0 bits in TCR_X are enabled, and the lowe TCNT_X can be cleared by the counter.
- Pin output
  - — Control of output from the TMOY pin by bits OS3 to OS0 in TCSR_Y is in acco with the upper 8-bit compare-match conditions.
  - — Control of output from the TMOX pin by bits OS3 to OS0 in TCSR_X is in acco with the lower 8-bit compare-match conditions.

13.7.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_X are set to B'100 and the CKSX bit in TCRXY is se TCNT_X counts the occurrence of compare-match A for TMR_Y. TMR_X and TMR_Y controlled independently. Conditions such as setting of the CMF flag, generation of inte output from the TMO pin, and counter clearing are in accordance with the settings for ea channel.

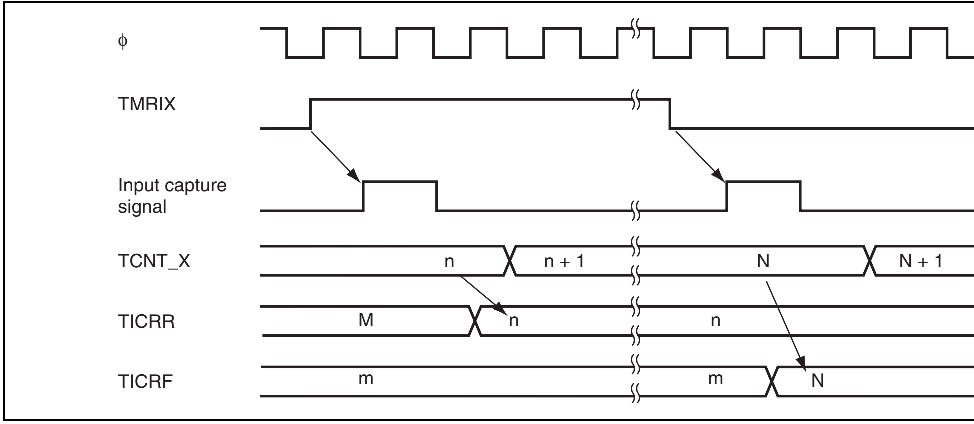


Figure 13.11 Timing of Input Capture Operation

If the input capture signal is input while TICRR and TICRF are being read, the input capt signal is delayed by one system clock (φ) cycle. Figure 13.12 shows the timing of this op

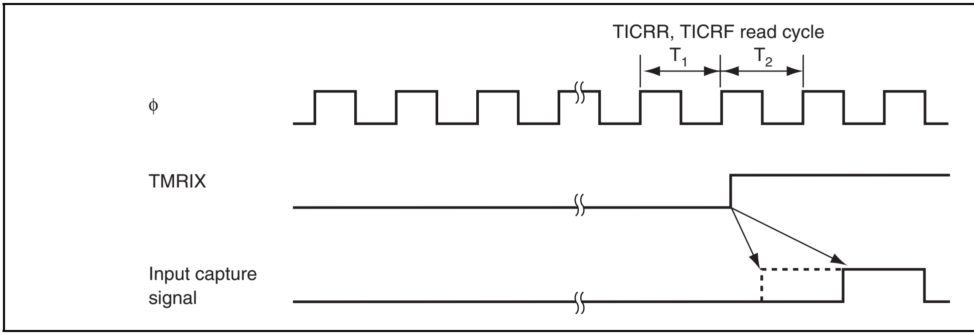


Figure 13.12 Timing of Input Capture Signal
(Input capture signal is input during TICRR and TICRF read)

1	TMIX pin input selection

Channel	Name	Interrupt Source	Interrupt Flag	Prio
TMR_0	CMIA0	TCORA_0 compare-match	CMFA	High
	CMIB0	TCORB_0 compare-match	CMFB	
	OVI0	TCNT_0 overflow	OVF	
TMR_1	CMIA1	TCORA_1 compare-match	CMFA	
	CMIB1	TCORB_1 compare-match	CMFB	
	OVI1	TCNT_1 overflow	OVF	
TMR_Y	CMIAY	TCORA_Y compare-match	CMFA	
	CMIBY	TCORB_Y compare-match	CMFB	
	OVIY	TCNT_Y overflow	OVF	
TMR_X	ICIX	Input capture	ICF	
	CMIAX	TCORA_X compare-match	CMFA	
	CMIBX	TCORB_X compare-match	CMFB	
	OVIX	TCNT_X overflow	OVF	Low

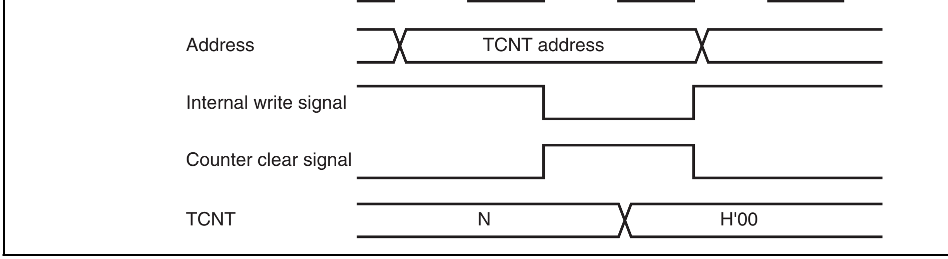


Figure 13.13   Conflict between TCNT Write and Clear

13.9.2 Conflict between TCNT Write and Count-Up

If a count-up occurs during the T_2 state of a TCNT write cycle as shown in figure 13.14, counter write takes priority and the counter is not incremented.

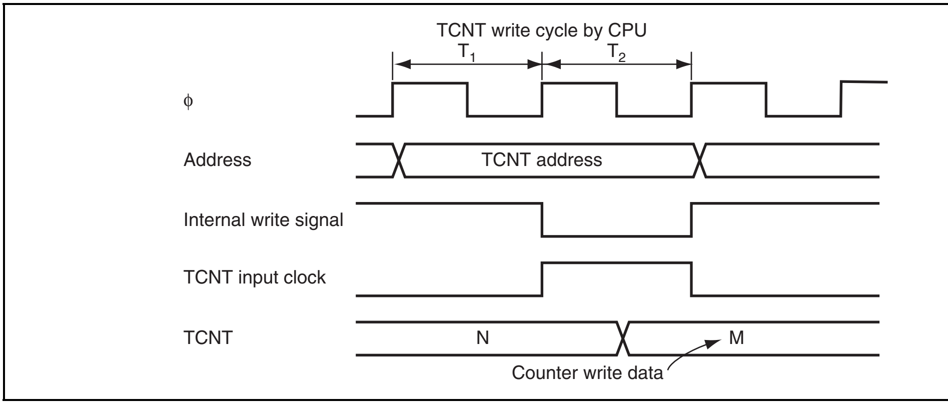


Figure 13.14   Conflict between TCNT Write and Count-Up

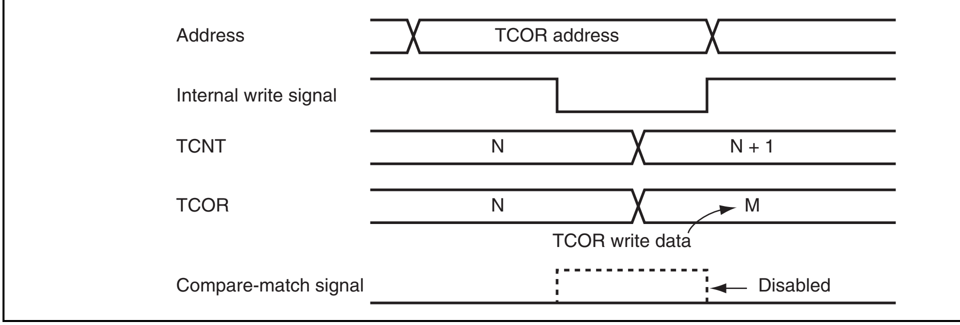


Figure 13.15 Conflict between TCOR Write and Compare-Match

13.9.4 Conflict between Compare-Matches A and B

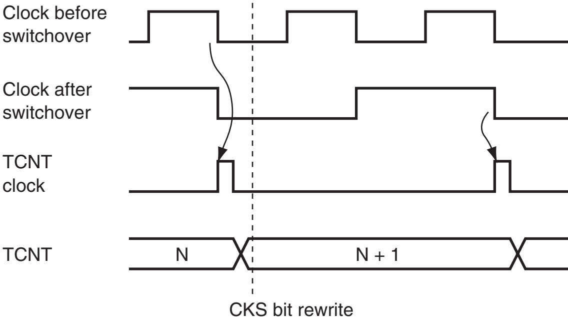
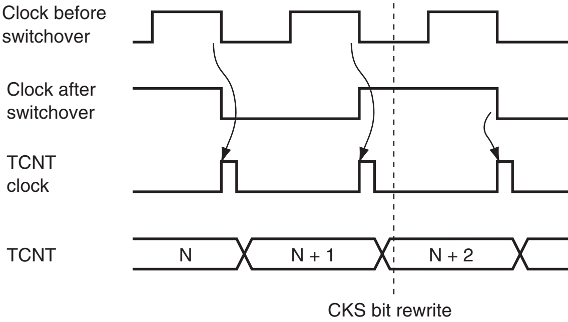
If compare-matches A and B occur at the same time, the operation follows the output stat defined for compare-match A or B, according to the priority of the timer output shown in 13.7.

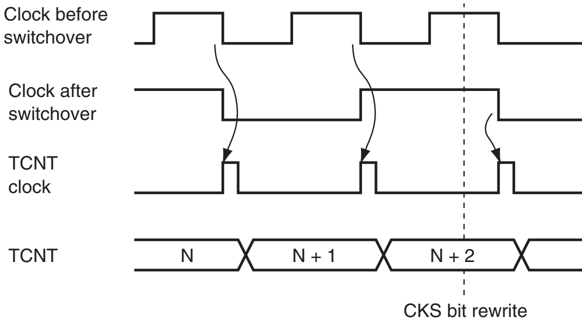
Table 13.7 Timer Output Priorities

Output Setting	Pric
Toggle output	Higl
1 output	↑
0 output	
No change	Low

Erroneous incrementation can also happen when switching between internal and externa

Table 13.8 Switching of Internal Clocks and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Clock switching from low to low level*[1]	 Clock before switchover Clock after switchover TCNT clock TCNT N N + 1 CKS bit rewrite
2	Clock switching from low to high level*[2]	 Clock before switchover Clock after switchover TCNT clock TCNT N N + 1 N + 2 CKS bit rewrite

		CKS bit rewrite
4	Clock switching from high to high level	 Clock before switchover Clock after switchover TCNT clock TCNT N N + 1 N + 2 CKS bit rewrite

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.

setting is for TMR operation to be halted. Register access is enabled by canceling the mo

mode. For details, see section 26, Power-Down Modes.

14.1 Features

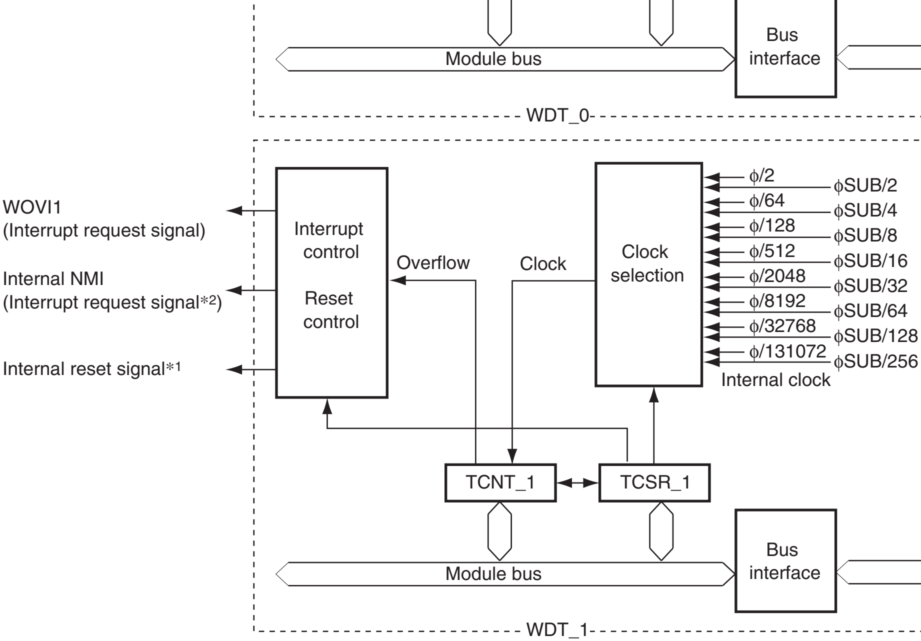
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

Watchdog Timer Mode:

- If the counter overflows, whether an internal reset or an internal NMI interrupt is ge can be selected.

Interval Timer Mode:

- If the counter overflows, an interval timer interrupt (WOVI) is generated.



[Legend]

TCSR_0: Timer control/status register_0
TCNT_0: Timer counter_0
TCSR_1: Timer control/status register_1
TCNT_1: Timer counter_1

Notes: 1. The internal reset signal first resets the WDT in which the overflow has occurred first.
2. The internal NMI interrupt signal can be independently output from either WDT_0 or WDT_1. The interrupt controller does not distinguish the NMI interrupt request from WDT_0 from that from WDT_1.

Figure 14.1 Block Diagram of WDT

14.3 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCSR and TC to be written to in a method different from normal registers. For details, see section 14.6 on Register Access. For details on the system control register, see section 3.2.2, System Register (SYSCR).

Table 14.2 Register Configuration

						D
Channel	**Register Name**	**Abbreviation**	**R/W**	**Initial Value**	**Address**	**W**
Channel 0	Timer counter_0	TCNT_0	R/W	H'00	H'FFA8	1
					H'FFA9*	8
	Timer control/status register_0	TCSR_0	R/W	H'00	H'FFA8	1
					H'FFA8*	8
Channel 1	Timer counter_1	TCNT_1	R/W	H'00	H'FFEA	1
					H'FFEB*	8
	Timer control/status register_1	TCSR_1	R/W	H'00	H'FFEA	1
					H'FFEA*	8

Note: * Address in the upper cell: when writing.
Address in the lower cell: when reading

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed (changes H'FF to H'00). [Setting condition] When TCNT overflows (changes from H'FF to When internal reset request generation is selec watchdog timer mode, OVF is cleared automat the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1, then 0 to OVF • When 0 is written to TME
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdo or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting initialized to H′00.
4	—	0	R/W	Reserved The initial value should not be changed.

parentheses:

000: ϕ/2 (frequency: 25.6 μs)

001: ϕ/64 (frequency: 819.2 μs)

010: ϕ/128 (frequency: 1.6 ms)

011: ϕ/512 (frequency: 6.6 ms)

100: ϕ/2048 (frequency: 26.2 ms)

101: ϕ/8192 (frequency: 104.9 ms)

110: ϕ/32768 (frequency: 419.4 ms)

111: ϕ/131072 (frequency: 1.68 s)

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

				the internal reset. [Clearing conditions] When TCSR is read when OVF = 1*[2], then 0 is to OVF When 0 is written to TME
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchd or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting initialized to H'00.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided cycle of φ–based prescal 1: Counts the divided cycle of φSUB–based pre (PSS)
3	RST/$\overline{\text{NMI}}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI i when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

011: φ/512 (frequency: 6.6 ms)

100: φ/2048 (frequency: 26.2 ms)

101: φ/8192 (frequency: 104.9 ms)

110: φ/32768 (frequency: 419.4 ms)

111: φ/131072 (frequency: 1.68 s)

When PSS = 1:

000: φSUB/2 (cycle: 15.6 ms)

001: φSUB/4 (cycle: 31.3 ms)

010: φSUB/8 (cycle: 62.5 ms)

011: φSUB/16 (cycle: 125 ms)

100: φSUB/32 (cycle: 250 ms)

101: φSUB/64 (cycle: 500 ms)

110: φSUB/128 (cycle: 1 s)

111: φSUB/256 (cycle: 2 s)

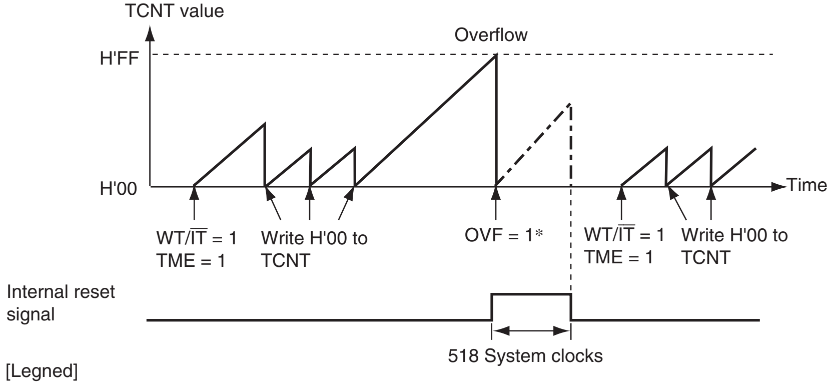
Notes: 1. Only 0 can be written, to clear the flag.

2. When OVF is polled with the interval timer interrupt disabled, OVF = 1 must b least twice.

If the RST/$\overline{\text{NMI}}$ bit of TCSR is set to 1, when the TCNT overflows, an internal reset sign LSI is issued for 518 system clocks as shown in figure 14.2. If the RST/$\overline{\text{NMI}}$ bit is cleare when the TCNT overflows, an NMI interrupt request is generated.

An internal reset request from the watchdog timer and a reset input from the $\overline{\text{RES}}$ pin are processed in the same vector. Reset source can be identified by the XRST bit status in SY If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset cause WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the XRST bit in SYSCR is set to 1.

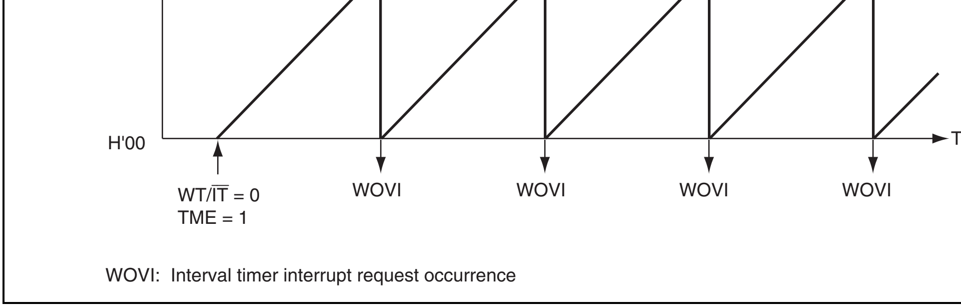
An NMI interrupt request from the watchdog timer and an interrupt request from the NM processed in the same vector. Do not handle an NMI interrupt request from the watchdog and an interrupt request from the NMI pin at the same time.



[Legned]
WT/$\overline{\text{IT}}$: Timer mode select bit
TME: Timer enable bit
OVF: Overflow flag

Note * After the OVF bit becomes 1, it is cleared to 0 by an internal reset.
The XRST bit is also cleared to 0.

Figure 14.2 Watchdog Timer Mode (RST/$\overline{\text{NMI}}$ = 1) Operation



WOVI: Interval timer interrupt request occurrence

Figure 14.3 Interval Timer Mode Operation

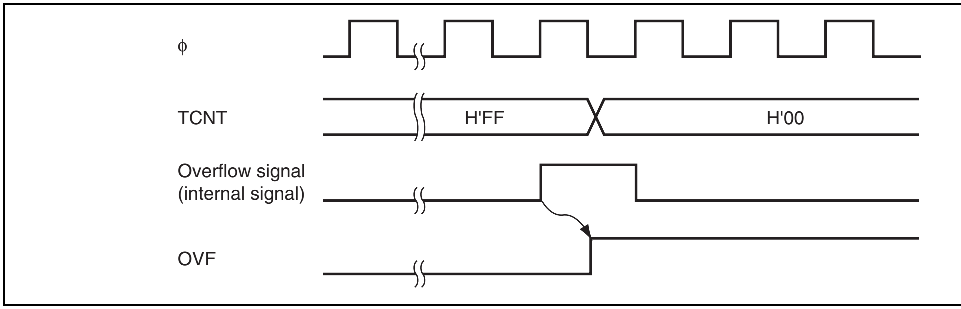


Figure 14.4 OVF Flag Set Timing

Name	Interrupt Source	Interrupt Flag
WOVI	TCNT overflow	OVF

14.6 Usage Notes

14.6.1 Notes on Register Access

The watchdog timer’s registers, TCNT and TCSR differ from other registers in being mo
difficult to write to. The procedures for writing to and reading from these registers are giv
below.

(1) Writing to TCNT and TCSR (Example of WDT_0)

These registers must be written to by a word transfer instruction. They cannot be written
byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, satisfy the relative conditi
shown in figure 14.5 to write to TCNT or TCSR. To write to TCNT, the higher bytes mus
the value H'5A and the lower bytes must contain the write data before the transfer instruc
execution. To write to TCSR, the higher bytes must contain the value H'A5 and the lower
must contain the write data.

(2) Reading from TCNT and TCSR (Example of WDT_0)

These registers are read in the same way as other registers. The read address is H'FFA8 f
and H'FFA9 for TCNT.

14.6.2 Conflict between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the
takes priority and the timer counter is not incremented. Figure 14.6 shows this operation

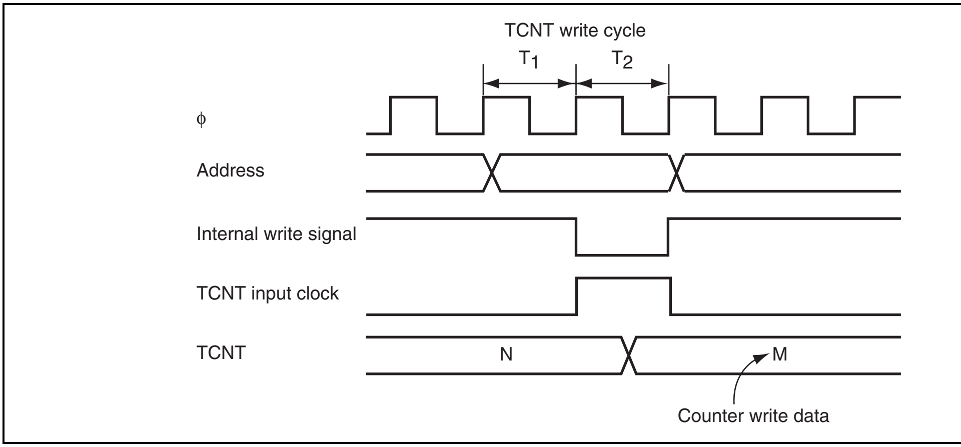


Figure 14.6 Conflict between TCNT Write and Increment

PSS bit.

14.6.5 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from/to watchdog timer to/from interval timer, while the WDT is operating, errors could occur in the operation. Software must stop the watchdog timer (by the TME bit to 0) before switching the mode.

15.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and re be executed simultaneously. Double-buffering is used in both the transmitter and the enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

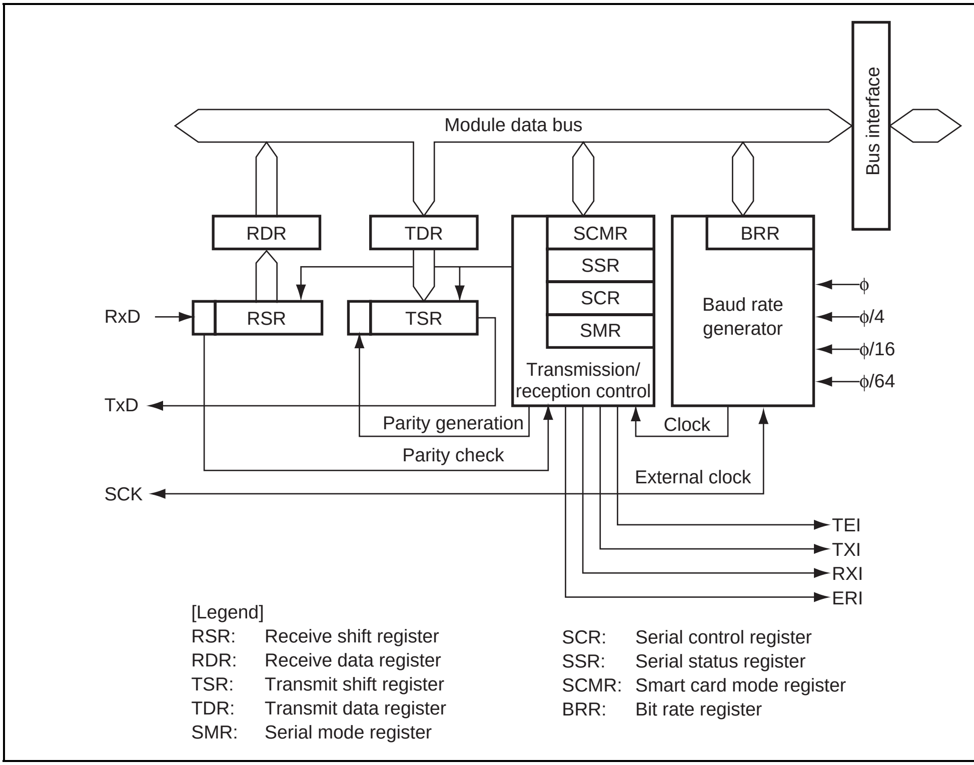
  The External clock can be selected as a transfer clock source (except for the smart ca interface).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode
- Four interrupt sources

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and r error — that can issue requests.

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 framing error
- Multiprocessor communication capability

Figure 15.1 shows a block diagram of SCI.



[Legend]

RSR:	Receive shift register	SCR:	Serial control register
RDR:	Receive data register	SSR:	Serial status register
TSR:	Transmit shift register	SCMR:	Smart card mode register
TDR:	Transmit data register	BRR:	Bit rate register
SMR:	Serial mode register		

Figure 15.1 Block Diagram of SCI

2	SCK2	Input/Output	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 channel designation.

Channel 1	Serial mode register_1	SMR_1	R/W	H'00	H'FF88	8
	Bit rate register_1	BRR_1	R/W	H'FF	H'FF89	8
	Serial control register_1	SCR_1	R/W	H'00	H'FF8A	8
	Transmit data register_1	TDR_1	R/W	H'FF	H'FF8B	8
	Serial status register_1	SSR_1	R/W	H'84	H'FF8C	8
	Receive data register_1	RDR_1	R	H'00	H'FF8D	8
	Smart card mode register_1	SCMR_1	R/W	H'F2	H'FF8E	8
Channel 2	Serial mode register_2	SMR_2	R/W	H'00	H'FFA0	8
	Bit rate register_2	BRR_2	R/W	H'FF	H'FFA1	8
	Serial control register_2	SCR_2	R/W	H'00	H'FFA2	8
	Transmit data register_2	TDR_2	R/W	H'FF	H'FFA3	8
	Serial status register_2	SSR_2	R/W	H'84	H'FFA4	8
	Receive data register_2	RDR_2	R	H'00	H'FFA5	8
	Smart card mode register_2	SCMR_2	R/W	H'F2	H'FFA6	8

receive the next data. Since RSR and RDR function as a double buffer in this way, conti receive operations be performed. After confirming that the RDRF bit in SSR is set to 1, for only once. RDR cannot be written to by the CPU. The initial value of RDR is H'00.

15.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is emp transfers the transmit data written in TDR to TSR and starts transmission. The double-bu structures of TDR and TSR enable continuous serial transmission. If the next transmit da already been written to TDR when one frame of data is transmitted, the SCI transfers the data to TSR to continue transmission. Although TDR can be read from or written to by t all times, to achieve reliable serial transmission, write transmit data to TDR for only onc confirming that the TDRE bit in SSR is set to 1. The initial value of TDR is H'FF.

15.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the transfers transmit data from TDR to TSR, and then sends the data to the TxD pin. TSR c directly accessed by the CPU.

				0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchro mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-firs and the MSB of TDR is not transmitted i transmission. In clocked synchronous mode, a fixed data 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronou When this bit is set to 1, the parity bit is add transmit data before transmission, and the is checked in reception. For a multiprocess parity bit addition and checking are not perf regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchron mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checke second stop bit is 0, it is treated as the star the next transmit frame.

				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 regist and the baud rate, see section 15.3.9, Bit Register (BRR). n is the decimal display o of n in BRR (see section 15.3.9, Bit Rate F (BRR)).

- Bit Functions in Smart Card Interface Mode (when SMIF in SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode ope GSM mode, the TEND set timing is put for 11.0 etu* from the start and the clock outp function is appended. For details, see sect 15.7.8, Clock Output Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer m operation. For details, see section 15.7.3, Transfer Mode.
5	PE	0	R/W	Parity Enable (valid only in asynchronous When this bit is set to 1, the parity bit is ad transmit data before transmission, and the is checked in reception. Set this bit to 1 in card interface mode.

				a 1-bit data transfer time in smart card inter mode. 00: 32 clock cycles (S = 32) 01: 64 clock cycles (S = 64) 10: 372 clock cycles (S = 372) 11: 256 clock cycles (S = 256) For details, see section 15.7.4, Receive Da Sampling Timing and Reception Margin. S described in section 15.3.9, Bit Rate Regist (BRR).
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the ba 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 registe and the baud rate, see section 15.3.9, Bit R Register (BRR). n is the decimal display of of n in BRR (see section 15.3.9, Bit Rate R (BRR)).

Note: * etu: Element Time Unit (time taken to transfer one bit)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt re enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI inte requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is e
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enab
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled the MP bit in SMR is 1 in asynchronous m When this bit is set to 1, receive data in w multiprocessor bit is 0 is skipped, and sett RDRF, FER, and ORER status flags in S disabled. On receiving data in which the multiprocessor bit is 1, this bit is automati cleared and normal reception is resumed. details, see section 15.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt re enabled.

- Clocked synchronous mode

0x: Internal clock (SCK pin functions as clo
output.)

1x External clock (SCK pin functions as clo
input.)

[Legend]

x: Don't care

				When this bit is set to 1, transmission is e
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is ena
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled when the MP bit in SMR is 1 in asynchro mode) Write 0 to this bit in smart card interface
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in smart card interface
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Controls the clock output from the SCK p GSM mode, clock output can be dynamic switched. For details, see section 15.7.8, Output Control. • When GM in SMR = 0 00: Output disabled (SCK pin functions a 01: Clock output 1x: Reserved • When GM in SMR = 1 00: Output fixed to low 01: Clock output 10: Output fixed to high 11: Clock output

[Legend]

x: Don't care

				Indicates whether TDR contains transmit [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TDR is ready for data write [Clearing condition] When 0 is written to TDRE after reading T
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that receive data is stored in RD [Setting condition] When serial reception ends normally and data is transferred from RSR to RDR [Clearing condition] When 0 is written to RDRF after reading F The RDRF flag is not affected and retains previous value when the RE bit in SCR is to 0.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] When the next serial reception is complet RDRF = 1 [Clearing condition] When 0 is written to ORER after reading 1

				When a parity error is detected during re [Clearing condition] When 0 is written to PER after reading P
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of t of a 1-byte serial transmit character [Clearing condition] When 0 is written to TDRE after reading
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the frame. When the RE bit in SCR is cleare previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be the transmit frame.

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

				When 0 is written to TDRE after reading T
6	RDRF	0	R/(W)*[1]	Receive Data Register Full Indicates that receive data is stored in RD [Setting condition] When serial reception ends normally and r data is transferred from RSR to RDR [Clearing condition] When 0 is written to RDRF after reading R The RDRF flag is not affected and retains previous value when the RE bit in SCR is 0.
5	ORER	0	R/(W)*[1]	Overrun Error [Setting condition] When the next serial reception is complete RDRF = 1 [Clearing condition] When 0 is written to ORER after reading O
4	ERS	0	R/(W)*[1]	Error Signal Status [Setting condition] When a low error signal is sampled [Clearing condition] When 0 is written to ERS after reading ER

				[Setting conditions] • When both TE and EPS in SCR are • When ERS = 0 and TDRE = 1 after a time passed after the start of 1-byte transfer. The set timing depends on t register setting as follows. • When GM = 0 and BLK = 0, 2.5 etu* transmission start • When GM = 0 and BLK = 1, 1.5 etu* transmission start • When GM = 1 and BLK = 0, 1.0 etu* transmission start • When GM = 1 and BLK = 1, 1.0 etu* transmission start [Clearing condition] When 0 is written to TDRE after reading
1	MPB	0	R	Multiprocessor Bit Not used in smart card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in smart card interface

Notes: 1. Only 0 can be written to clear the flag.

2. etu: Element Time Unit (time taken to transfer one bit)

				0: TDR contents are transmitted with LSB Receive data is stored as LSB first in I 1: TDR contents are transmitted with MS Receive data is stored as MSB first in The SDIR bit is valid only when the 8-bit c format is used for transmission/reception the 7-bit data format is used, data is alwa transmitted/received with LSB-first.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level SINV bit does not affect the logic level of parity bit. When the parity bit is inverted, O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they a Receive data is stored as it is in RDR. 1: TDR contents are inverted before bein transmitted. Receive data is stored in i form in RDR.
1	—	1	R	Reserved This bit is always read as 1 and cannot b modified.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interf mode is selected. 0: Normal asynchronous or clocked sync mode 1: Smart card interface mode

Mode	Bit Rate	Error
Asynchronous mode	$B = \dfrac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \{ \dfrac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N}$
Clocked synchronous mode	$B = \dfrac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	—
Smart card interface mode	$B = \dfrac{\phi \times 10^6}{S \times 2^{2n+1} \times (N + 1)}$	$\text{Error (\%)} = \{ \dfrac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N}$

[Legend] B: Bit rate (bit/s)

N: BRR setting for baud rate generator (0 ≤ N ≤ 255)

ϕ: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following table

SMR Setting		
CKS1	**CKS0**	**n**
0	0	0
0	1	1
1	0	2
1	1	3

SMR Setting		
BCP1	**BCP0**	**S**
0	0	3
0	1	6
1	0	3
1	1	2

Table 15.4 shows sample N settings in BRR in normal asynchronous mode. Table 15.5 s
maximum bit rate settable for each frequency. Table 15.7 and 15.9 show sample N settin
BRR in clocked synchronous mode and smart card interface mode, respectively. In smar
interface mode, the number of basic clock cycles S in a 1-bit data transfer time can be se
For details, see section 15.7.4, Receive Data Sampling Timing and Reception Margin. T
and 15.8 show the maximum bit rates with external clock input.

1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11
38400	—	—	—	0	7	0.00	0	7	1.73	0	9

	Operating Frequency ϕ (MHz)										
	12.288			**14**			**14.7456**			**1**	
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	217	0.08	2	248	–0.17	3	64	0.70	3	70
150	2	159	0.00	2	181	0.16	2	191	0.00	2	207
300	2	79	0.00	2	90	0.16	2	95	0.00	2	103
600	1	159	0.00	1	181	0.16	1	191	0.00	1	207
1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15
38400	0	9	0.00	—	—	—	0	11	0.00	0	12

[Legend]

—: Can be set, but there will be a degree of error.

Note: * Make the settings so that the error does not exceed 1%.

1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32
31250	0	16	1.20	0	17	0.00	0	19	–1.70	0	19
38400	0	16	0.00	0	14	–2.34	0	15	0.00	0	15

[Legend]

—: Can be set, but there will be a degree of error.

Note: * Make the settings so that the error does not exceed 1%.

Table 15.5 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
8	250000	0	0	14.7456	460800	0
9.8304	307200	0	0	16	500000	0
10	312500	0	0	17.2032	537600	0
12	375000	0	0	18	562500	0
12.288	384000	0	0	19.6608	614400	0
14	437500	0	0	20	625000	0

Table 15.7 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

	Operating Frequency φ (MHz)						
	8		10		16		2
Bit Rate (bit/s)	n	N	n	N	n	N	n
110							
250	3	124	—	—	3	249	
500	2	249	—	—	3	124	—
1k	2	124	—	—	2	249	—
2.5k	1	199	1	249	2	99	2
5k	1	99	1	124	1	199	1
10k	0	199	0	249	1	99	1
25k	0	79	0	99	0	159	0
50k	0	39	0	49	0	79	0
100k	0	19	0	24	0	39	0
250k	0	7	0	9	0	15	0
500k	0	3	0	4	0	7	0
1M	0	1			0	3	0
2.5M			0	0*			0
5M							0

[Legend]

Blank: Setting prohibited.

— : Can be set, but there will be a degree of error.

*: Continuous transfer or reception is not possible.

	Operating Frequency ϕ (MHz)										
	10.00			13.00			14.2848			16.	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
9600	0	1	30	0	1	-8.99	0	1	0.00	0	1

	Operating Frequency ϕ (MHz)					
	18.00			20.00		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)
9600	0	2	-15.99	0	2	-6.65

Table 15.10 Maximum Bit Rate for Each Frequency (Smart Card Interface Mode,

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n
10.00	13441	0	0	16.00	21505	0
13.00	17473	0	0	18.00	24194	0
14.2848	19200	0	0	20.00	26882	0

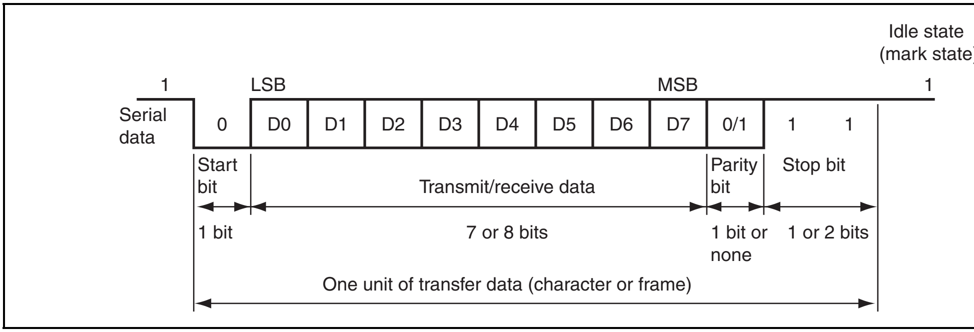


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	S	8-bit data	STOP		
0	0	0	1	S	8-bit data	STOP	ST	
0	1	0	0	S	8-bit data	P	ST	
0	1	0	1	S	8-bit data	P	ST	
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	ST
0	—	1	0	S	8-bit data	MPB	ST	
0	—	1	1	S	8-bit data	MPB	ST	
1	—	1	0	S	7-bit data	MPB	STOP	
1	—	1	1	S	7-bit data	MPB	STOP	ST

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined 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 allowe system design.

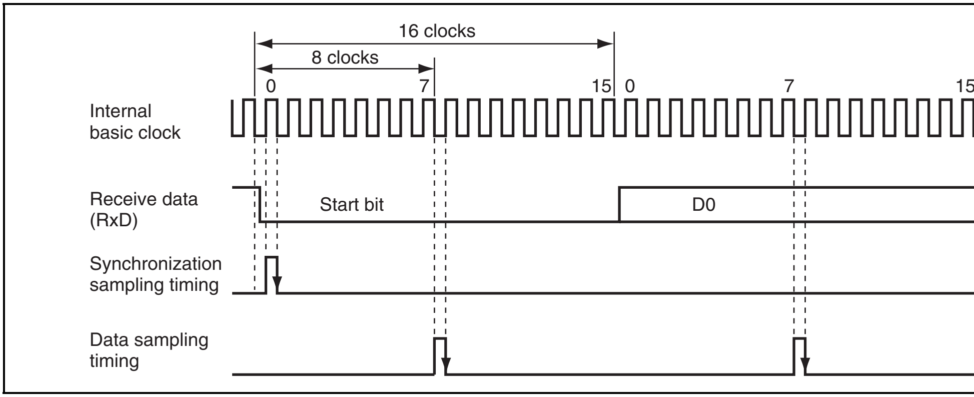


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

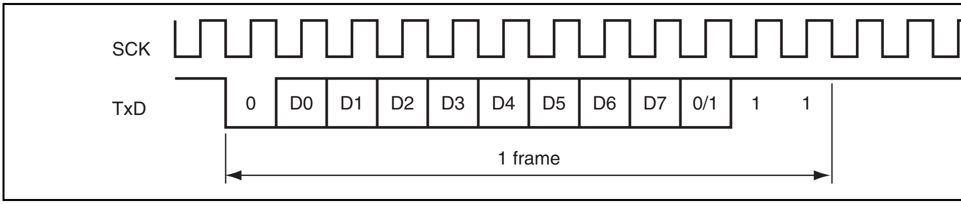
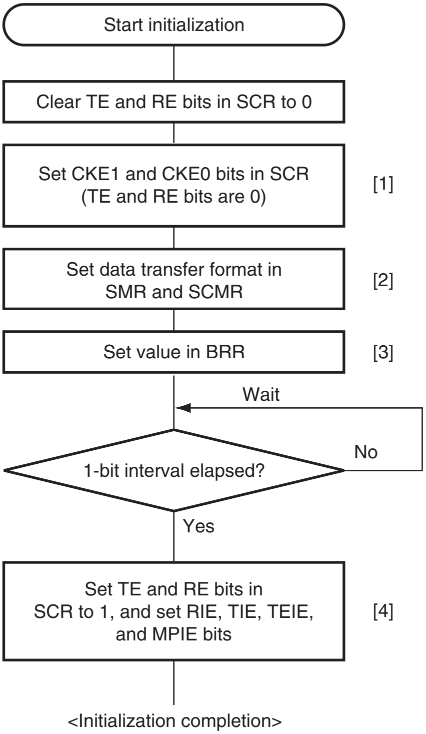


Figure 15.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)


Start initialization
Clear TE and RE bits in SCR to 0
Set CKE1 and CKE0 bits in SCR
(TE and RE bits are 0)
[1]
Set data transfer format in
SMR and SCMR
[2]
Set value in BRR
[3]
Wait
1-bit interval elapsed?
No
Yes
Set TE and RE bits in
SCR to 1, and set RIE, TIE, TEIE,
and MPIE bits
[4]
<Initialization completion>

[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 15.5 Sample SCI Initialization Flowchart

TDR before transmission of the current transmit data has finished, continuous transm
be enabled.

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity b
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, a
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
state" is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a
interrupt request is generated.

Figure 15.7 shows a sample flowchart for transmission in asynchronous mode.

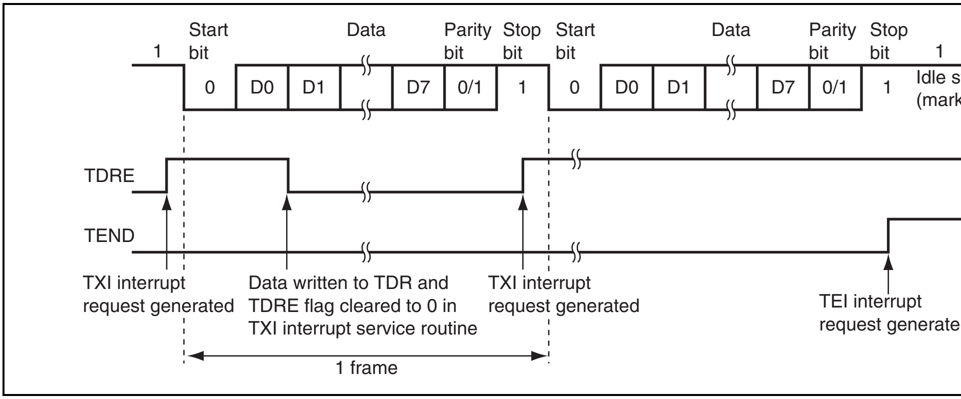
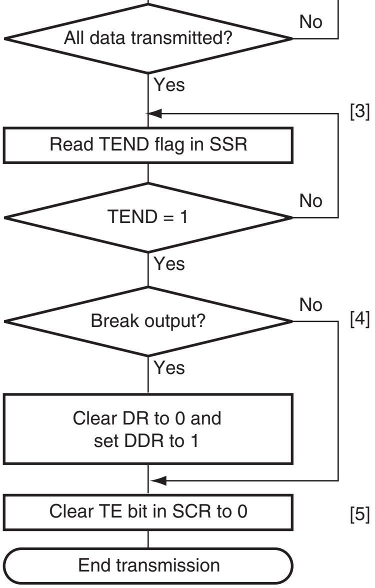

1
Start bit
0
D0
D1
Data
D7
Parity bit
0/1
Stop bit
1
Start bit
0
D0
D1
Data
D7
Parity bit
0/1
Stop bit
1
1
Idle s
(mark
TDRE
TEND
TXI interrupt request generated
Data written to TDR and TDRE flag cleared to 0 in TXI interrupt service routine
TXI interrupt request generated
TEI interrupt request generate
1 frame

Figure 15.6 Example of Operation in Transmission in Asynchronous Mo
(Example with 8-Bit Data, Parity, One Stop Bit)



To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0.

[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.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [5].

Figure 15.7 Sample Serial Transmission Flowchart

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 transfer
RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is genera

4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 a
data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI inter
request is generated.

5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data i
transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt re
generated. Because the RXI interrupt routine reads the receive data transferred to RD
reception of the next receive data has finished, continuous reception can be enabled.

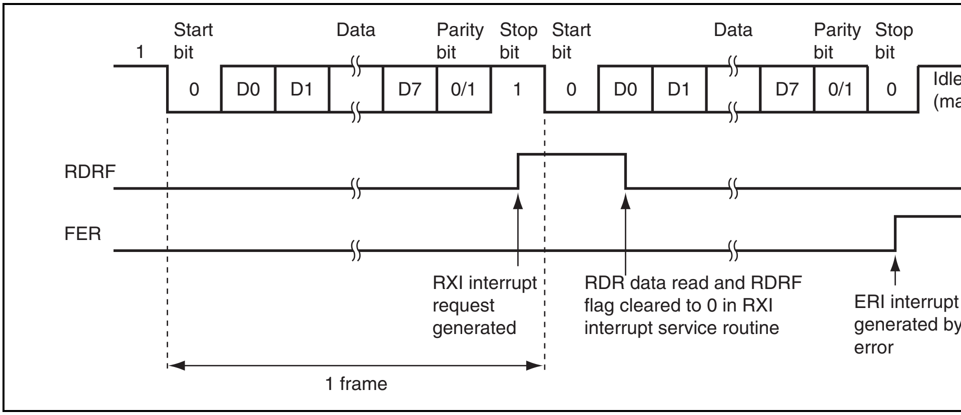
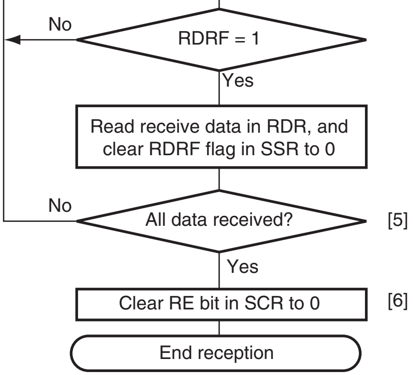


Figure 15.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

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 + framin
1	1	0	1	Lost	Overrun error + parity
0	0	1	1	Transferred to RDR	Framing error + parity
1	1	1	1	Lost	Overrun error + framin parity error

Note: * The RDRF flag retains the state it had before data reception.



[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.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [6].

[Legend]
∨: Logical add (OR)

Figure 15.9 Sample Serial Reception Flowchart (1)

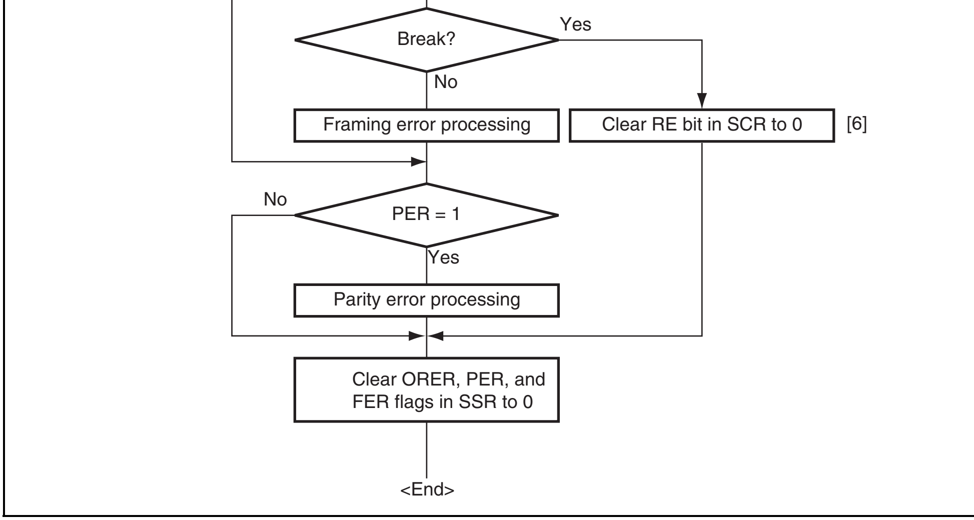


Figure 15.9 Sample Serial Reception Flowchart (2)

transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cyc
15.10 shows an example of inter-processor communication using the multiprocessor for
transmitting station first sends the ID code of the receiving station with which it wants t
serial communication as data with a 1 multiprocessor bit added. It then sends transmit d
with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the
station compares that data with its own ID. The station whose ID matches then receives
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
transfer of receive data from RSR to RDR, error flag detection, and setting the RDRF, F
ORER status flags in SSR to 1 are prohibited until data with a 1 multiprocessor bit is re
reception of a receive character with a 1 multiprocessor bit, the MPB bit in SSR is set to
MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in S
1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit
are the same as those in normal asynchronous mode. The clock used for multiprocessor
communication is the same as that in normal asynchronous mode.

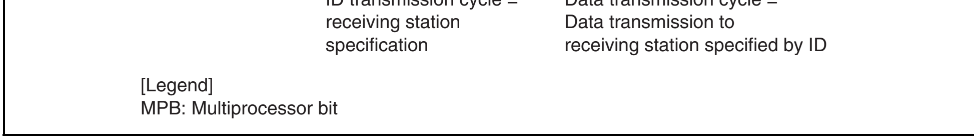
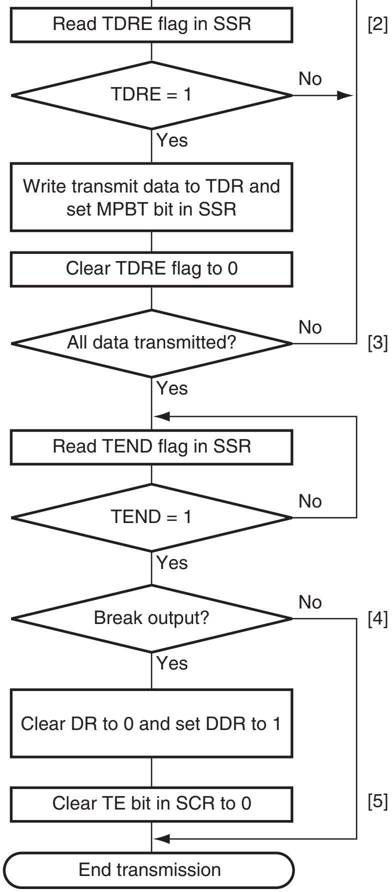


Figure 15.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



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.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set port DDR to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [5].

Figure 15.11 Sample Multiprocessor Serial Transmission Flowchart

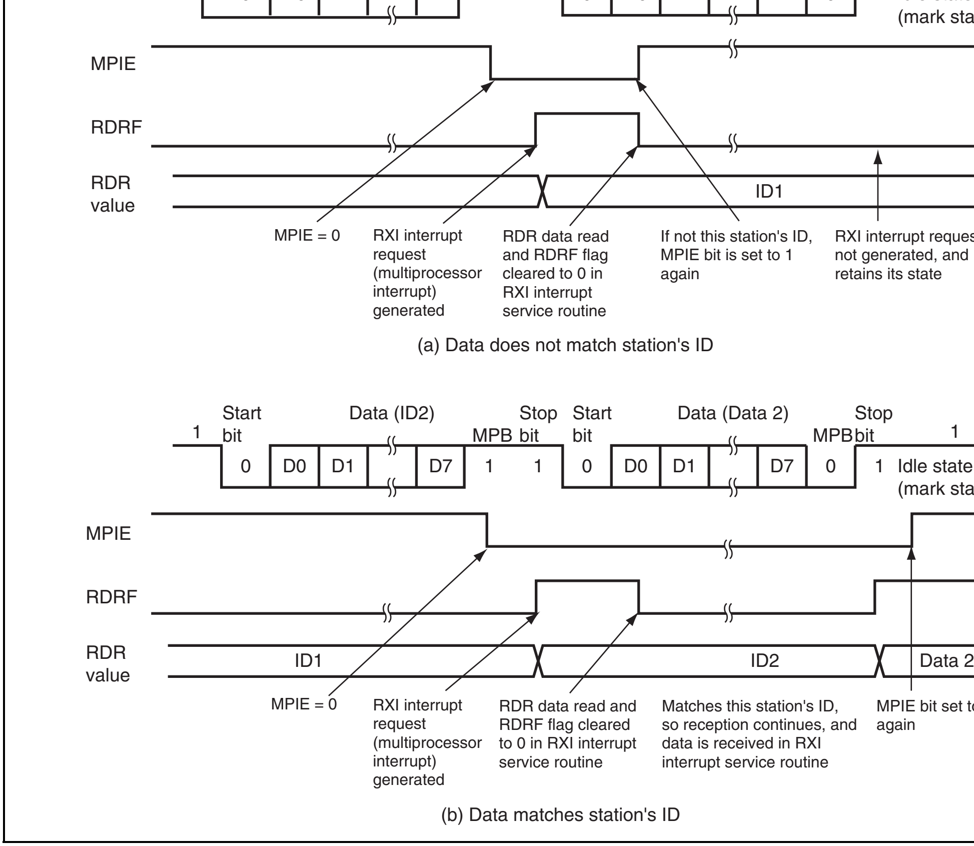
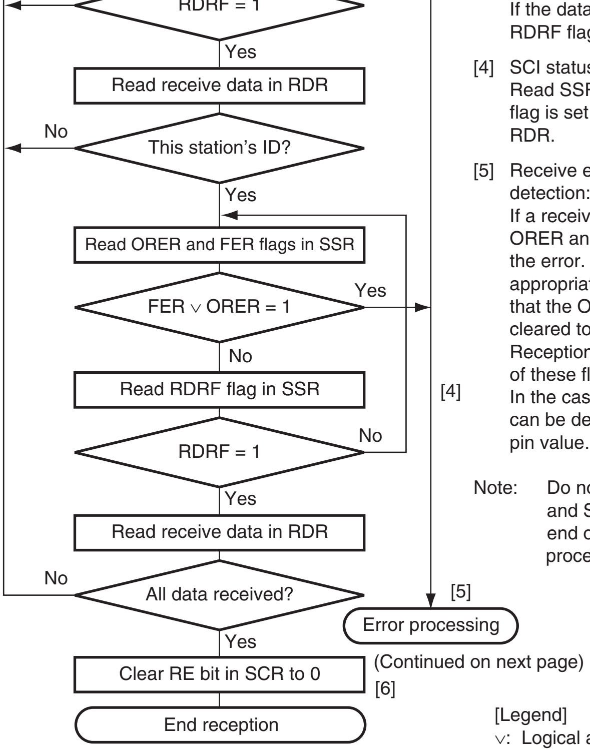

(mark sta
MPIE
RDRF
RDR value
ID1
MPIE = 0
RXI interrupt request (multiprocessor interrupt) generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt service routine
If not this station's ID, MPIE bit is set to 1 again
RXI interrupt reques not generated, and retains its state
(a) Data does not match station's ID
1
Start bit
0
D0
D1
D7
Data (ID2)
MPB
1
Stop bit
1
Start bit
0
D0
D1
D7
Data (Data 2)
MPB
0
Stop bit
1
1
Idle state (mark sta
MPIE
RDRF
RDR value
ID1
ID2
Data 2
MPIE = 0
RXI interrupt request (multiprocessor interrupt) generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt service routine
Matches this station's ID, so reception continues, and data is received in RXI interrupt service routine
MPIE bit set t again
(b) Data matches station's ID

Figure 15.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



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 processing 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 processing, ensure that the ORER and FER flags are all 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.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of reception except the process of [6].

Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1)

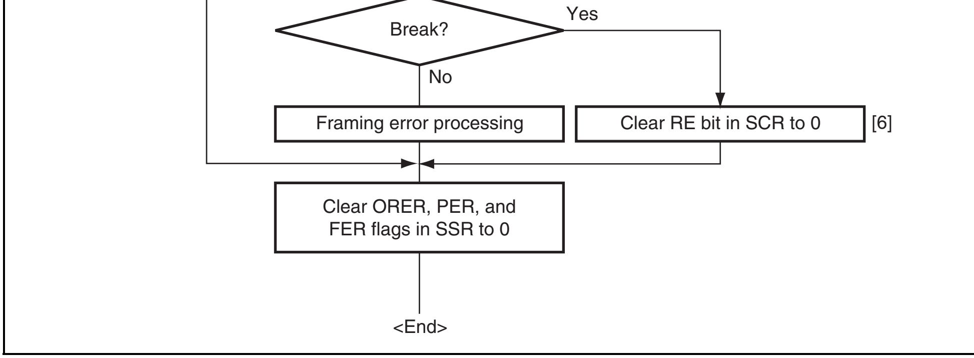


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (2)

a double-buffered structure, so that the next transmit data can be written during transmis previous receive data can be read during reception, enabling continuous data transfer.

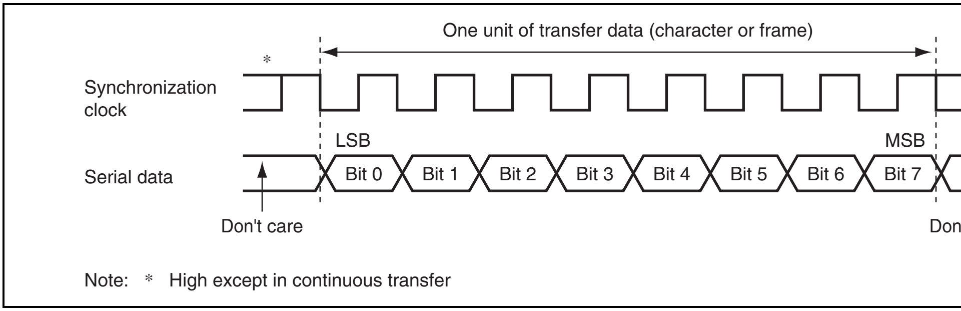
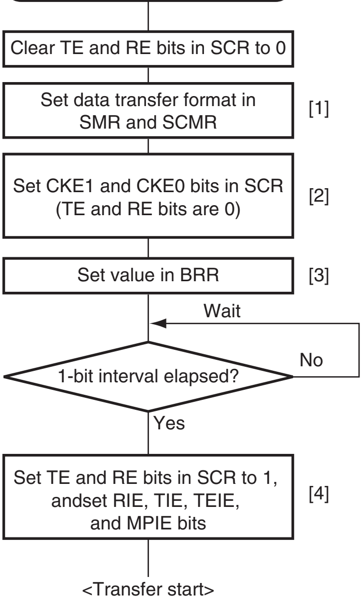


Figure 15.14 Data Format in Synchronous Communication (LSB-First)

15.6.1 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 tl and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronizati is output from the SCK pin. Eight synchronization clock pulses are output in the transfer character, and when no transfer is performed the clock is fixed high.


Clear TE and RE bits in SCR to 0
Set data transfer format in SMR and SCMR
[1]
Set CKE1 and CKE0 bits in SCR (TE and RE bits are 0)
[2]
Set value in BRR
[3]
Wait
1-bit interval elapsed?
No
Yes
Set TE and RE bits in SCR to 1, andset RIE, TIE, TEIE, and MPIE bits
[4]
<Transfer start>

and SCMR.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, MPIE, TE, and RE to 0.

[3] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1.
Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 15.15 Sample SCI Initialization Flowchart

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 c mode has been specified and synchronized with the input clock when use of an exter has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial trans 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 main output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interru is generated. The SCK pin is fixed high.

Figure 15.17 shows a sample flowchart for serial data transmission. Even if the TDRE fl cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) Make sure to clear the receive error flags to 0 before starting transmission. Note that cle RE bit to 0 does not clear the receive error flags.

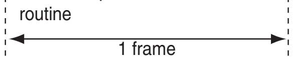
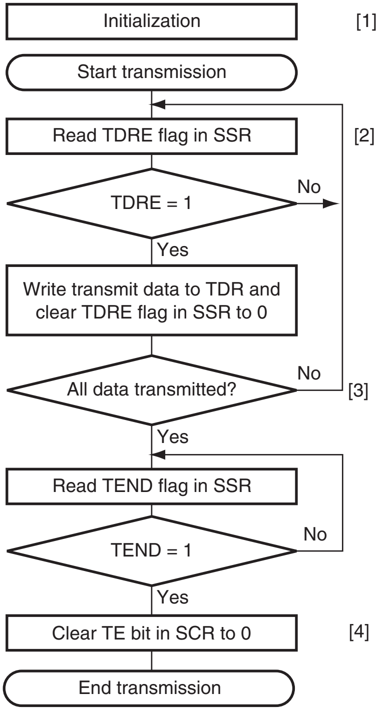


Figure 15.16 Sample SCI Transmission Operation in Clocked Synchronous M



[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.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [4].

Figure 15.17 Sample Serial Transmission Flowchart

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 rec generated. Because the RXI interrupt routine reads the receive data transferred to RD reception of the next receive data has finished, continuous reception can be enabled.

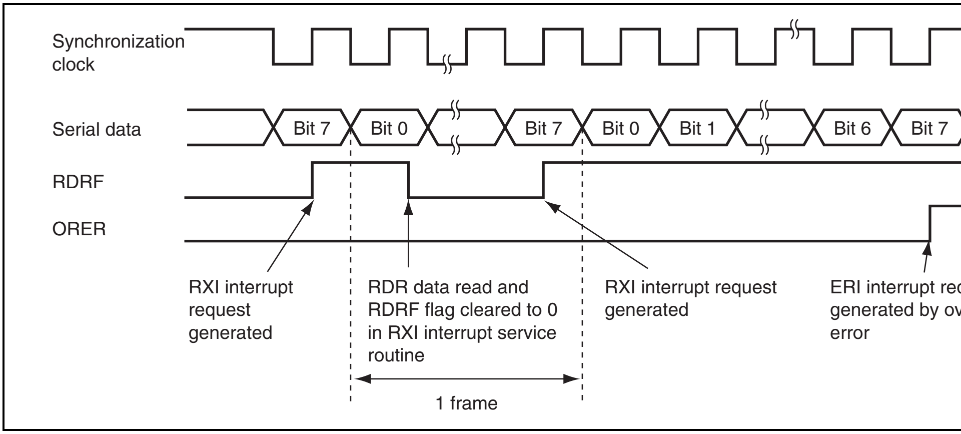
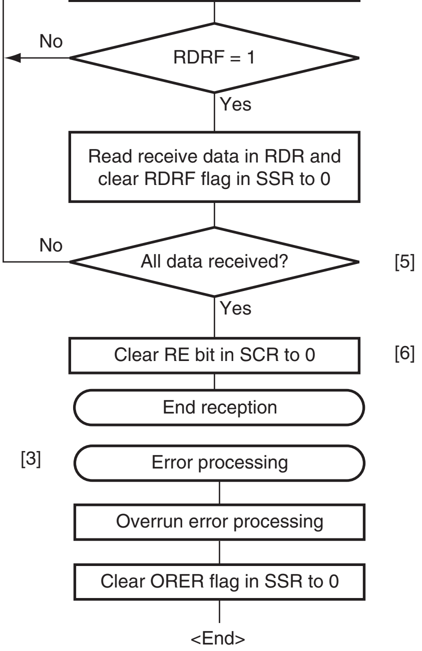

Synchronization clock
Serial data
Bit 7
Bit 0
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
RDRF
ORER
RXI interrupt request generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt service routine
RXI interrupt request generated
ERI interrupt rec generated by ov error
1 frame

Figure 15.18 Example of SCI Receive Operation in Clocked Synchronous M

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.19 shows a sample for serial data reception.



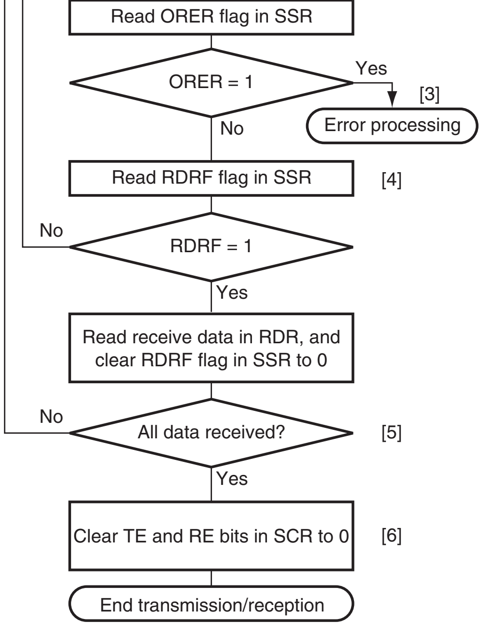
data in RDR and clear the RDRF flag to 0.
Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of reception except the process of [6].

Figure 15.19 Sample Serial Reception Flowchart

that the RDRF bit in SSR and receive error flags (ORER, FER, and PER) are cleared to
simultaneously set the TE and RE bits to 1 with a single instruction.



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 fla 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, befor 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.

Notes: 1. When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.
2. Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission/receptio except the process of [6].

Figure 15.20 Sample Flowchart of Simultaneous Serial Transmission and Rece

TxD and RxD pins and pull up the data transmission line to VCC using a resistor. Settin
and TE bits in SCR to 1 with the IC card not connected enables closed transmission/rece
allowing self diagnosis. To supply the IC card with the clock pulses generated by the SC
the SCK pin output to the CLK pin of the IC card. A reset signal can be supplied via the
port of this LSI.

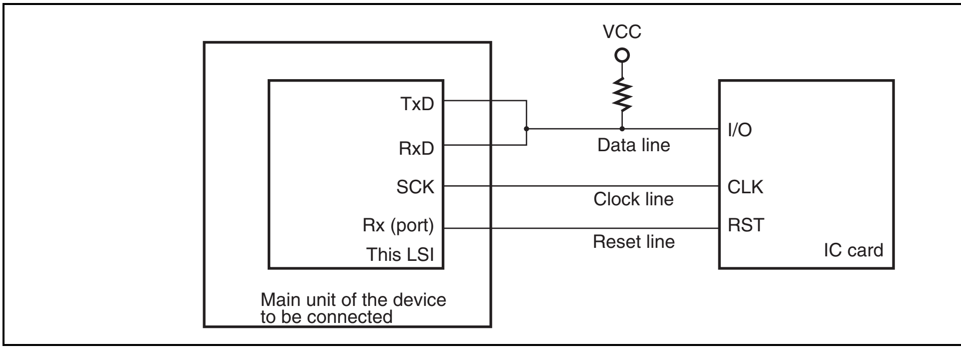


Figure 15.21 Pin Connection for Smart Card Interface

after two or more etu.

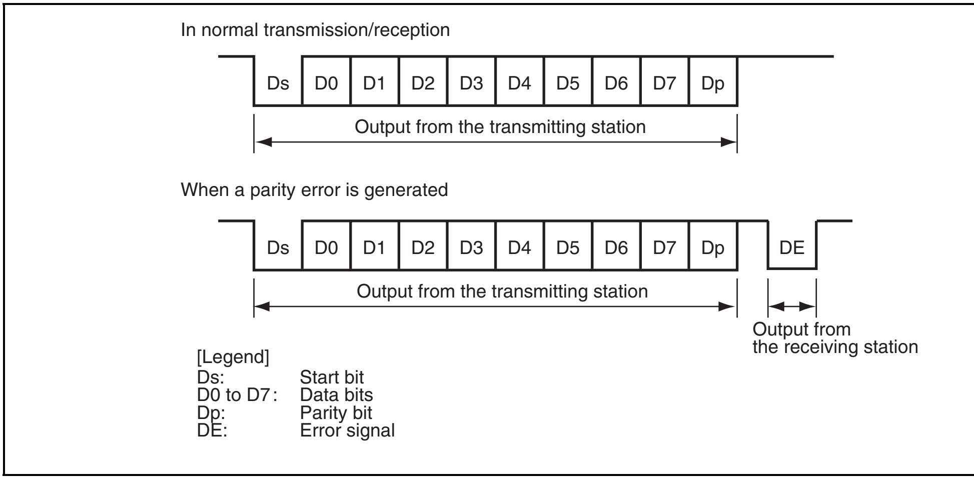


Figure 15.22 Data Formats in Normal Smart Card Interface Mode

For communication with the IC cards of the direct convention and inverse convention typ follow the procedure below.

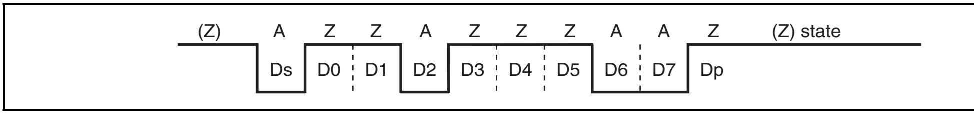


Figure 15.23 Direct Convention (SDIR = SINV = O/$\overline{\text{E}}$ = 0)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respe
and data is transferred with MSB-first as the start character, as shown in figure 15.24. Th
data in the start character in the figure is H'3F. When using the inverse convention type,
both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even p
which is prescribed by the smart card standard, and corresponds to state Z. Since the SIN
this LSI only inverts data bits D7 to D0, write 1 to the $O/\overline{E}$ bit in SMR to invert the parit
both transmission and reception.

15.7.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following

- If a parity error is detected during reception, no error signal is output. Since the PER
SSR is set by error detection, clear the bit before receiving the parity bit of the next f
- During transmission, at least 1 etu is secured as a guard time after the end of the pari
before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR
11.5 etu after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in no
smart card interface mode, the flag is always read as 0 because no error signal is tran

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|\,D - 0.5\,|}{N}(1 + F) \right| \times 100\ [\%] \quad \cdots \quad \text{Formula}$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock rate deviation

Assuming values of F = 0, D = 0.5, and N = 372 in formula (1), the reception margin is determined by the formula below.

$$M = (0.5 - 1 / 2 \times 372) \times 100\ [\%] = 49.866\%$$

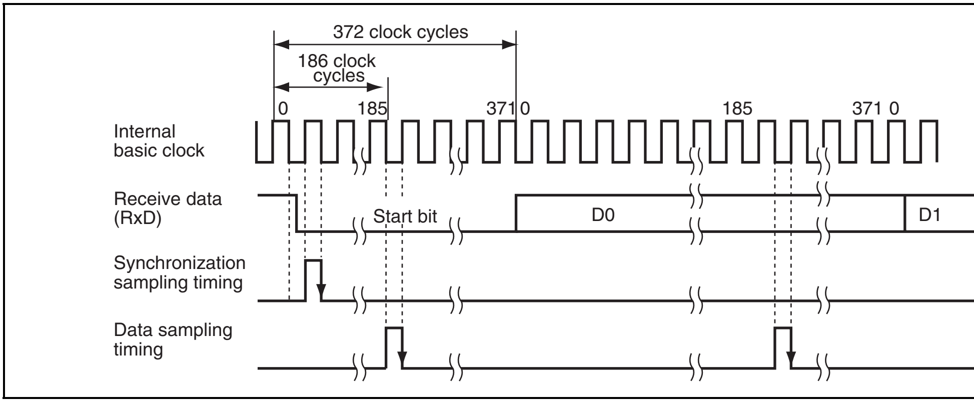


Figure 15.25 Receive Data Sampling Timing in Smart Card Interface Mod
(When Clock Frequency is 372 Times the Bit Rate)

TxD and RxD pins are changed from port pins to SCI pins, placing the pins into high impedance state.

5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE1 and CKE0 bits in SCR appropriately. Clear the TIE, RIE, TE, RE, MP TEIE bits to 0 simultaneously. When the CKE0 bit is set to 1, the SCK pin is allowe clock pulses.
7. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least 1 b Setting prohibited the TE and RE bits to 1 simultaneously except for self diagnosis.

To switch from reception to transmission, first verify that reception has completed, and the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. R completion can be verified by reading the RDRF flag or PER and ORER flags. To switc transmission to reception, first verify that transmission has completed, and initialize the the end of initialization, TE and RE should be set to 0 and 1, respectively. Transmission completion can be verified by reading the TEND flag.

15.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Data transmission in smart card interface mode (except in block transfer mode) is differe that in normal serial communication interface mode in that an error signal is sampled an re-transmitted. Figure 15.26 shows the data re-transfer operation during transmission.

1. If an error signal from the receiving end is sampled after one frame of data has been transmitted, the ERS bit in SSR is set to 1. Here, an ERI interrupt request is generate RIE bit in SCR is set to 1. Clear the ERS bit to 0 before the next parity bit is sampled
2. For the frame in which an error signal is received, the TEND bit in SSR is not set to re-transferred from TDR to TSR allowing automatic data retransmission.

request to be generated at error occurrence.

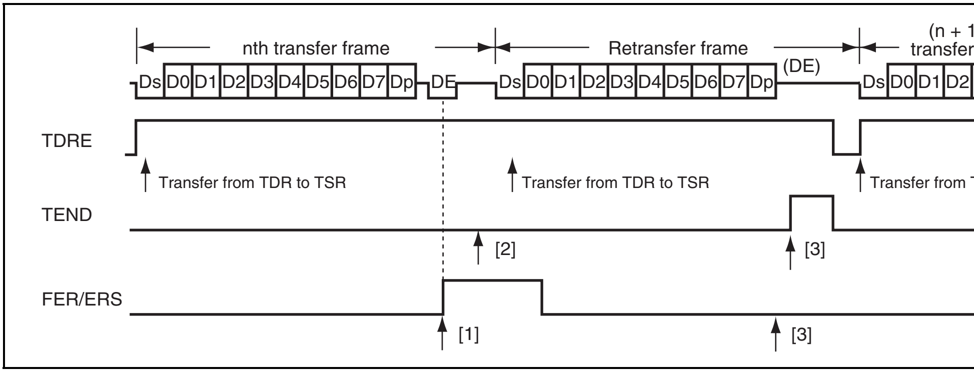


Figure 15.26 Data Re-transfer Operation in SCI Transmission Mode

Note that the TEND flag is set in different timings depending on the GM bit setting in SM which is shown in figure 15.27.

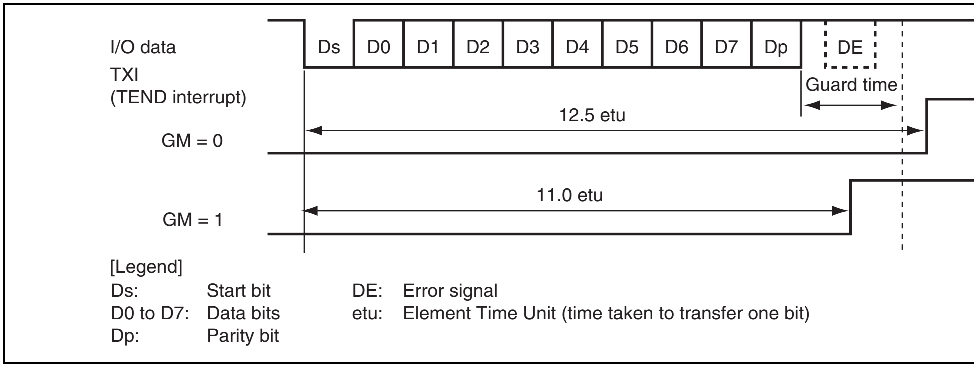


Figure 15.27 TEND Flag Set Timings during Transmission

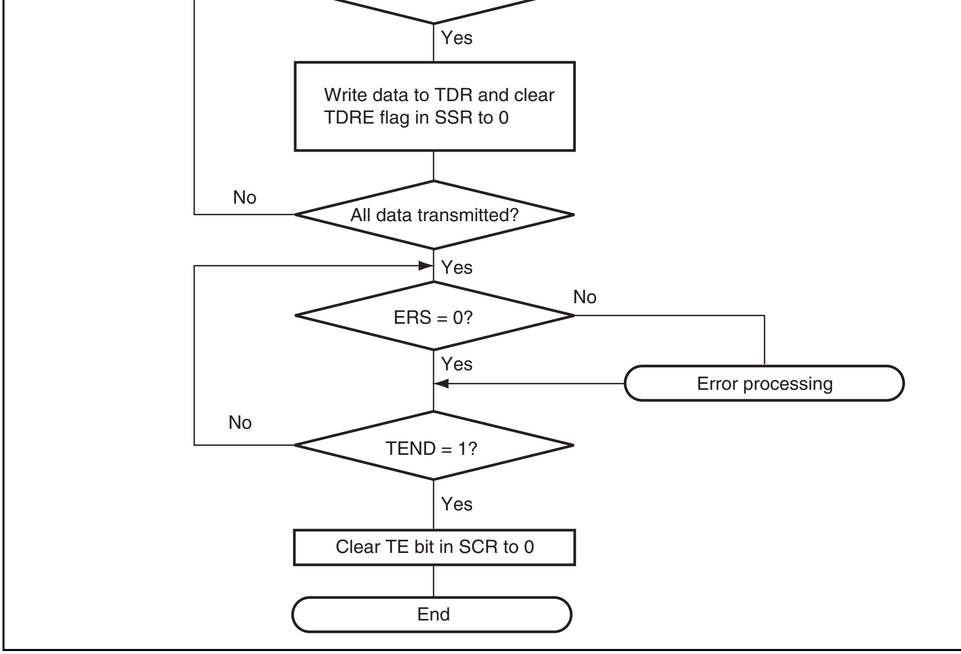


Figure 15.28 Sample Transmission Flowchart

to have been received successfully, and the RDRF bit in SSR is set to 1. Here, an RXI
request is generated if the RIE bit in SCR is set.

Figure 15.30 shows a sample flowchart for reception. In reception, setting the RIE bit to 1
an RXI interrupt request to be generated when the RDRF flag is set to 1. If an error occur
reception, i.e., either the ORER or PER flag is set to 1, a transmit/receive error interrupt (
request is generated and the error flag must be cleared. Even if a parity error occurs and P
to 1 in reception, receive data is transferred to RDR, thus allowing the data to be read.

Note: For operations in block transfer mode, see section 15.4, Operation in Asynchrono

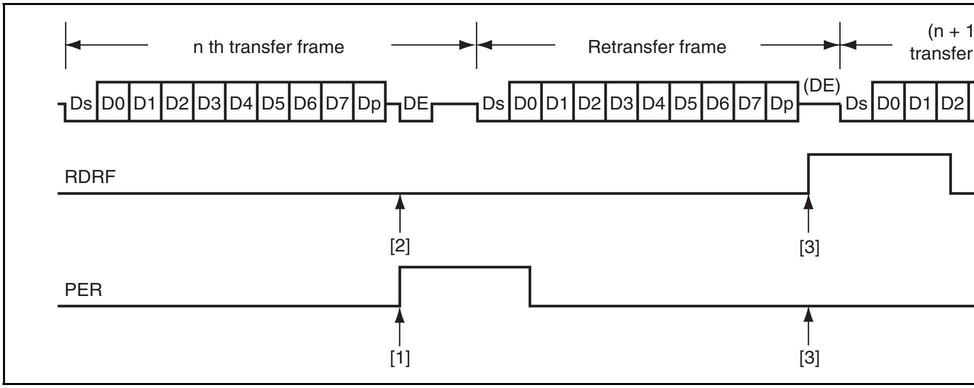


Figure 15.29 Data Re-transfer Operation in SCI Reception Mode

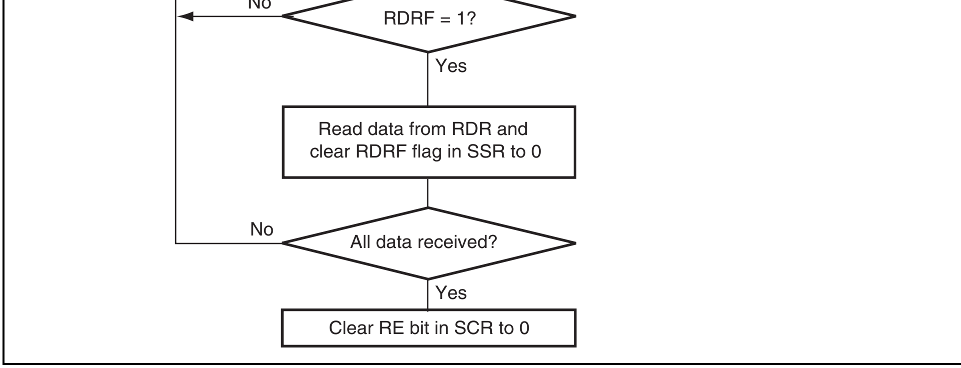


Figure 15.30 Sample Reception Flowchart

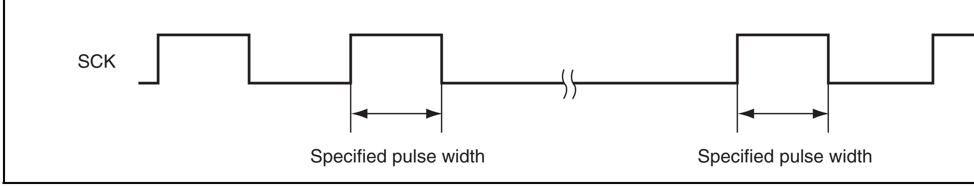


Figure 15.31 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure the appropriate clock duty ratio.

(1) At Power-On

To secure the appropriate clock duty ratio simultaneously with power-on, use the followi procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, u pull-up or pull-down resistor.
2. Fix the SCK pin to the specified output using the CKE1 bit in SCR.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the CKE0 bit in SCR to 1 to start clock output.

(3) At Transition from Software Standby Mode to Smart Card Interface Mode

1. Cancel software standby mode.
2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the approp ratio is then generated.

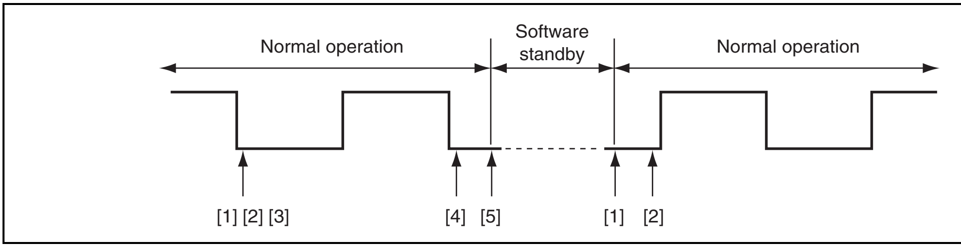


Figure 15.32 Clock Stop and Restart Procedure

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1 interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Table 15.13 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag
1	ERI1	Receive error	ORER, FER, PER
	RXI1	Receive data full	RDRF
	TXI1	Transmit data empty	TDRE
	TEI1	Transmit end	TEND
2	ERI2	Receive error	ORER, FER, PER
	RXI2	Receive data full	RDRF
	TXI2	Transmit data empty	TDRE
	TEI2	Transmit end	TEND

	TXI1	Transmit data empty	TEND
2	ERI2	Receive error, error signal detection	ORER, PER, ERS
	RXI2	Receive data full	RDRF
	TXI2	Transmit data empty	TEND

In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus gen TXI interrupt request. If an error occurs, the SCI automatically re-transmits the same da re-transmission, the TEND flag remains 0. Therefore, the SCI automatically transmits th specified number of bytes, including re-transmission in the case of error. However, the E in SSR, which is set at error occurrence, is not automatically cleared; the ERS flag must by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be ge error occurrence.

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to error occurs, the RDRF flag is not set but the error flag is set. Therefore, an ERI interru is issued to the CPU instead; the error flag must be cleared.

When framing error detection is performed, a break can be detected by reading the RxD p directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag in SS 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.

15.9.3 Mark State and Break Sending

When the TE bit in SCR is 0, the TxD pin is used as an I/O port whose direction (input or and level are determined by DR and DDR of the port. This can be used to set the TxD pin state (high level) or send a break during serial data transmission. To maintain the commu line at mark state until TE is set to 1, set both DDR and DR to 1. Since the TE bit is clear this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a during serial transmission, first set DDR to 1 and DR to 0, and then clear the TE bit to 0. TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission s TxD pin becomes an I/O port, and 0 is output from the TxD pin.

15.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mod

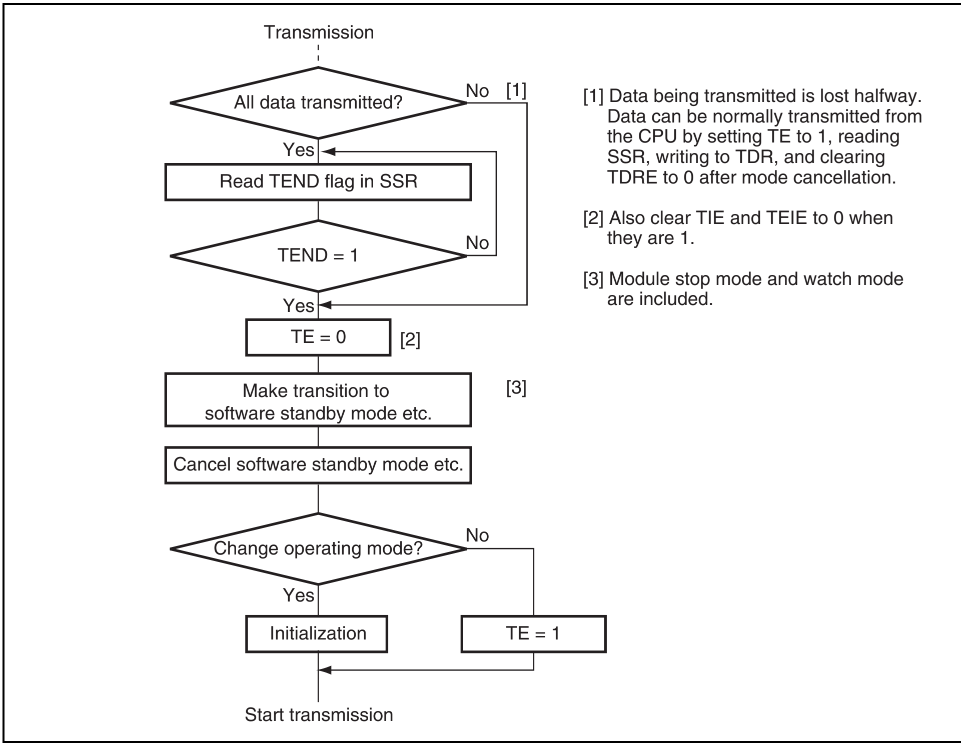
Transmission cannot be started when a receive error flag (ORER, FER, or RER) is SSR is even if the TDRE flag in SSR is cleared to 0. Be sure to clear the receive error flags to 0 b starting transmission. Note also that the receive error flags cannot be cleared to 0 even if t bit in SCR is cleared to 0.

15.9.5 Relation between Writing to TDR and TDRE Flag

Data can be written to TDR irrespective of the TDRE flag status in SSR. However, if the is written to TDR when the TDRE flag is 0, that is, when the previous data has not been transferred to TSR yet, the previous data in TDR is lost. Be sure to write transmit data to after verifying that the TDRE flag is set to 1.

write to TDR, clear TDRE in this order, and then start transmission. To transmit data in transmission mode, initialize the SCI first.

Figure 15.33 shows a sample flowchart for mode transition during transmission. Figures 15.35 show the pin states during transmission.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting TE to 1, reading SSR, writing to TDR, and clearing TDRE to 0 after mode cancellation.

[2] Also clear TIE and TEIE to 0 when they are 1.

[3] Module stop mode and watch mode are included.

Figure 15.33 Sample Flowchart for Mode Transition during Transmissio

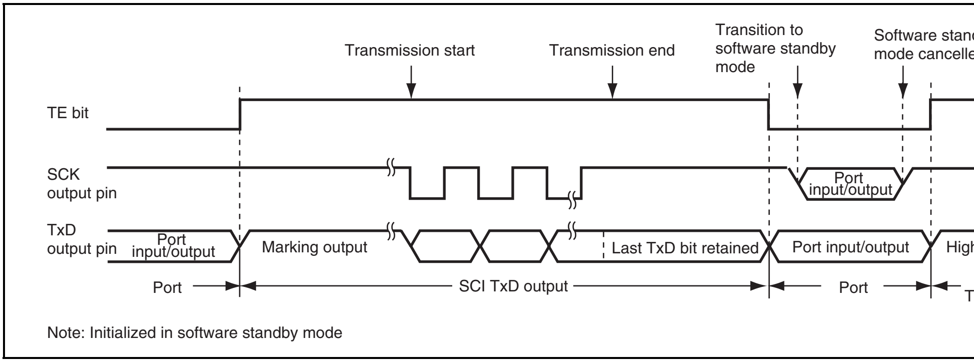
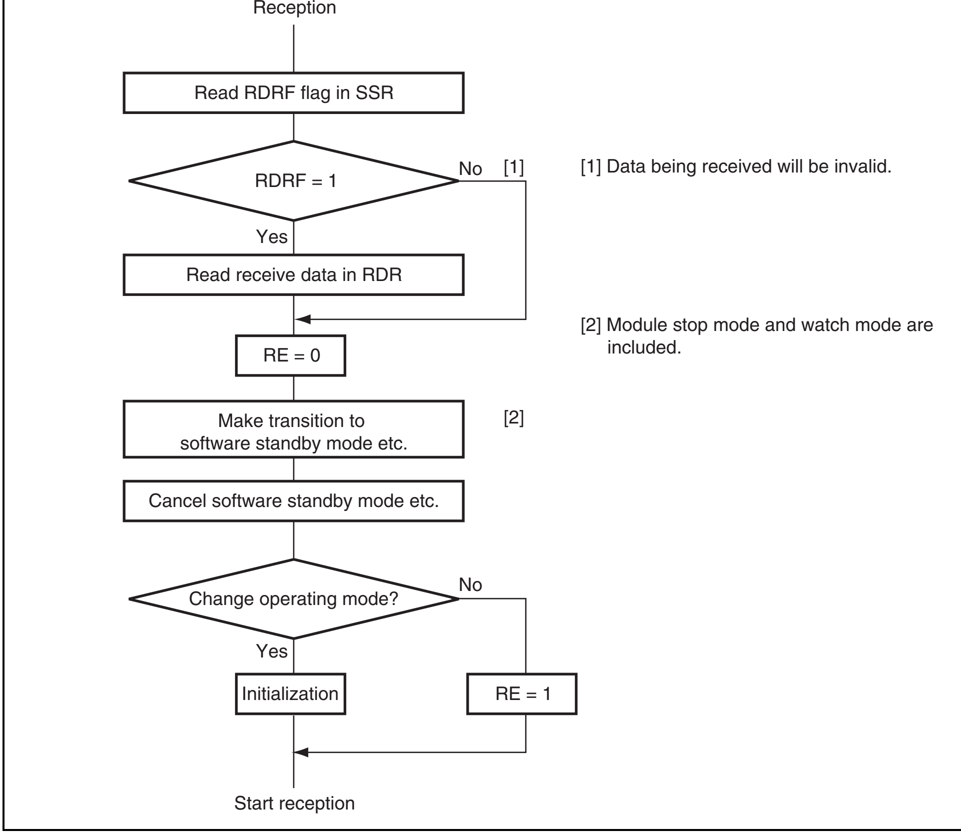


Figure 15.35 Pin States during Transmission in Clocked Synchronous Mod (Internal Clock)



[1] Data being received will be invalid.

[2] Module stop mode and watch mode are included.

Figure 15.36   Sample Flowchart for Mode Transition during Reception

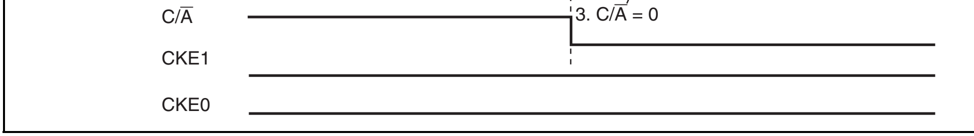


Figure 15.37 Switching from SCK Pins to Port Pins

To prevent the low pulse output that is generated when switching the SCK pins to the por specify the SCK pins for input (pull up the SCK/port pins externally), and follow the proc below with DDR = 1, DR = 1, C/$\overline{\text{A}}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1.

1. End serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/A bit = 0 (switch to port output)
5. CKE1 bit = 0

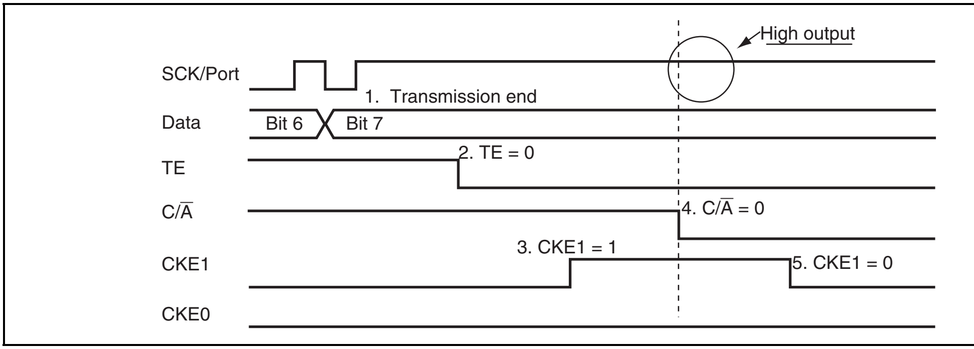


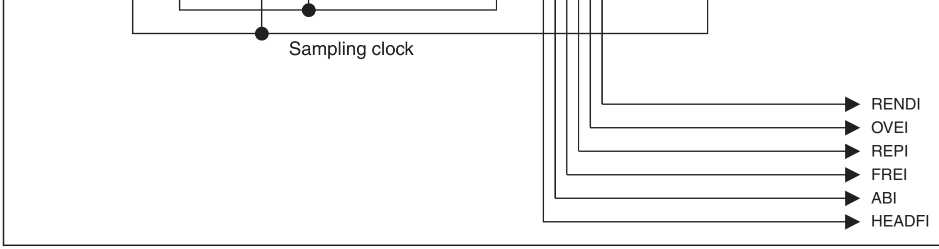
Figure 15.38 Prevention of Low Pulse Output at Switching from SCK Pins to Po

- Noise canceling function

  Input noise can be filtered out by using a maximum of four stages of filters.
- Polarity inversion of the input signal supported
- 18-byte FIFO incorporated
- Six interrupt sources: receive end, framing error, overrun error, repeat detection, abo generation, and header detection

  Interrupt sources can be specified by checking each flag.

Figure 16.1 is a block diagram of the CIR.



[Legend]
SFR: Receive shift register
CCR1: Receive control register 1
CCR2: Receive control register 2
CSTR: Receive status register
CEIR: Interrupt enable register
BRR: Bit rate register
CIRRDR0 to 17: Receive data register 0 to 17
HHMIN: Header minimum high-level period register

HHMAX: Header maximum high-level period re
HLMIN: Header minimum low-level period regis
HLMAX: Header maximum low-level period reg
DT1MIN: Data level 1 minimum period register
DT1MAX: Data level 1 maximum period registe
DT0MIN: Data level 0 minimum period register
DT0MAX: Data level 0 maximum period registe
RMIN: Repeat header minimum low-level perio
RMAX: Repeat header maximum low-level peri

Figure 16.1 CIR Block Diagram

Table 16.2 shows the CIR register configuration.

Table 16.2 List of Register Addresses

Register Name	Abbreviation	R/W	Initial Value
Receive control register 1	CCR1	R/W	H'00
Receive control register 2	CCR2	R/W	H'00
Receive status register	CSTR	R/W	H'00
Interrupt enable register	CEIR	R/W	H'00
Bit rate register	BRR	R/W	H'FF
Receive data register 0 to 17	CIRRDR0 to CIRRDR17	R	H'00
Header minimum high-level period register	HHMIN	R/W	H'0000
Header maximum high-level period register	HHMAX	R/W	H'0000
Header minimum low-level period register	HLMIN	R/W	H'00
Header maximum low-level period register	HLMAX	R/W	H'00
Data level 0 minimum period register	DT0MIN	R/W	H'00
Data level 0 maximum period register	DT0MAX	R/W	H'00
Data level 1 minimum period register	DT1MIN	R/W	H'00
Data level 1 maximum period register	DT1MAX	R/W	H'00

CCR1 enable/disable the CIR reception, controls a software reset of the CIR, select the p
the CIR input signals, and select the reference clock for CIR reception.

Bit	Bit Name	Initial Value	R/W	Description
7	CIRE	0	R/W	CIR Receive Enable 0: The CIR reception is disabled. 1: The CIR reception is enabled (Port is CIRI pin).
6	SRES	0	R/W	CIR Software Reset Controls initialization of the internal sequence CIR. 0: Normal operation 1: The internal sequencer is cleared. Writing 1 to this bit generates a clear signal f internal sequencer in the corresponding mod resulting in the initialization of the CIR's inter
5	CPHS	0	R/W	Input Signal Polarity Select 0: CIR input signal is used as is. 1: CIR input signal is inverted before use.
4	MLS	0	R/W	Receive Data Format Select 0: LSB-first data is received. 1: MSB-first data is received.

1	CLK1	0	R/W	Reference Clock
0	CLK0	0	R/W	00: Internal clock φ 01: Internal clock φ/2 10: Internal clock φ/4 11: subclock φsub

16.3.2 Receive Control Register 2 (CCR2)

CCR2 consists of the bits that select the CIR communication format.

Bit	Bit Name	Initial Value	R/W	Description
7	TFM1	0	R/W	Reception Signal Format Select
6	TFM0	0	R/W	00: NEC format (4 bytes are used) (Address, $\overline{\text{address}}$, command, and $\overline{\text{con}}$ stored in CIRRDR.) 01: NEC format (2 bytes are used) (Address and command are stored in C 10: Setting prohibited 11: Setting prohibited
5 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

				[Clearing condition] When the CIR has finished data reception.
6	CIRRDRF	0	R	Receive Data Register Full Indicates whether CIRRDR contains a recei or not. This bit cannot be modified. [Setting condition] When a receive data is stored into CIRRDR. [Clearing condition] When a receive data has been read from CI
5	REPF	0	R/W*	Repeat Detection Flag Indicates a repeat is generated. [Setting condition] When a repeat is detected. [Clearing condition] When writing 0 after reading REPF = 1.
4	OVRF	0	R/W*	Overrun Error Flag Indicates CIRRDR overflows. [Setting condition] When the next data is stored in CIRRDR wh CIRRDR is full. [Clearing condition] When writing 0 after reading OVRF = 1.

				(transfer format) is detected. [Setting condition] When data other than logic 0 or 1 is detect [Clearing condition] When writing 0 after reading ABF = 1.
1	FRF	0	R/W*	Framing Error Flag [Setting condition] • When a stop is detected during data re • When the time period of a stop is too sh [Clearing condition] When writing 0 after reading FRF = 1.
0	HEADF	0	R/W*	Header Detection Flag [Setting condition] When a header is detected. [Clearing condition] When writing 0 after reading HEADF = 1.

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

				1: REPI interrupt request is enabled.
4	OVEIE	0	R/W	Overrun Error Interrupt Enable 0: OVEI interrupt request is disabled. 1: OVEI interrupt request is enabled.
3	RENDIE	0	R/W	Receive End Interrupt Enable 0: RENDI interrupt request is disabled. 1: RENDI interrupt request is enabled.
2	ABIE	0	R/W	Abort Interrupt Enable 0: ABI interrupt request is disabled. 1: ABI interrupt request is enabled.
1	FREIE	0	R/W	Framing Error Interrupt Enable 0: FREI interrupt request is disabled. 1: FREI interrupt request is enabled.
0	HEADFIE	0	R/W	Header Detection Interrupt Enable 0: HEADFI interrupt request is disabled. 1: HEADFI interrupt request is enabled.

The following formula is used for calculating the bit rate, and the following table shows setting examples to obtain a target bit rate.

B = T / (N + 1)

B: Bit rate (bits/s)

T: Frequency of the reference clock (Hz) set by the CLK1 and CLK0 bits in CCR1 (ϕ, ϕ ϕsub)

N: Set value in BRR (0 ≤ N ≤ 255)

Table 16.3 Setting Example of BRR

Carrier Frequency	ϕ	CLK1 and CLK0 Setting	BRR Setting Value	Bit Rate (Kbit/s)	Devia Targe Frequ
38kHz	20 MHz	ϕ	H'FF	78.1	51.36
		ϕ/2	H'FF	39.1	2.72%
		ϕ/4	H'83	37.9	–0.32
	10 MHz	ϕ	H'FF	39.1	2.72%
		ϕ/2	H'83	37.9	–0.32
		ϕ/4	H'41	37.9	–0.32
	8 MHz	ϕ	H'D2	37.9	–0.23
		ϕ/2	H'69	37.7	–0.70
		ϕ/4	H'34	37.7	–0.70
	—	ϕsub	H'00	32.8	2.34%

16.3.7 Header Minimum/Maximum High-Level Period Register (HHMIN and HH

HHMIN and HHMAX control the noise canceler circuit, and specify the minimum and m high-level period for a header or repeat header, and low-level period for a stop.

- HHMIN

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	RFMBN4 to RFMBN0	All 0	R	Receive Byte Counter The RFMBN value is incremented by 1 (+1) time a byte is received. However, when RFM reaches B'10011, an overrun error occurs. A time, a receive data is not stored in CIRRDR When CIRRDR is read after the CIR has fini receiving (CIRBUSY = 0), RFMBN is decrem by 1 (–1). When CIRRDR is read while RFM B'00000, an undefined value is read. When is read during the CIR reception, an undefin is read and RFMBN is not decremented.
10	—	0	R/W	Reserved The initial value should not be changed.
9 to 0	HHMIN9 to HHMIN0	All 0	R/W	Specifies the minimum high-level period for or repeat header and the minimum low-level for a stop.

13	FLTE	0	R/W	Noise Canceler Circuit Enable 0: Disables the noise canceler circuit 1: Enables the noise canceler circuit
12	FLTCK1	0	R/W	Division Ratio Select for Noise Canceler Ci Clock Divides the frequency of the sampling clocl reception selected by BRR. 00: Not divided 01: Divided by 2 10: Divided by 4 11: Divided by 8
11	FLTCK0	0	R/W	
10	—	0	R/W	Reserved The initial value should not be changed.
9 to 0	HHMAX9 to HHMAX0	All 0	R/W	Specifies the maximum high-level period fo header or repeat header and the maximum period for a stop.

- HLMAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HLMAX7 to HLMAX0	H'00	R/W	Specifies the maximum low-level period for a

16.3.9 Data Level 1 Minimum/Maximum Period Register (DT1MIN/DT1MAX)

DT1MIN and DT1MAX specify the minimum and maximum low-level period for logic 1

- DT1MIN

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DT1MIN7 to DT1MIN0	H'00	R/W	Specifies the minimum low-level period for lo

- DT1MAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DT1MAX7 to DT1MAX0	H'00	R/W	Specifies the maximum low-level period for l

- DT0MAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DT0MAX7 to DT0MAX0	H'00	R/W	Specifies the maximum low/high-level perio 0, high-level period for logic 1, and high-lev for a stop/repeat.

16.3.11 Repeat Header Minimum/Maximum Low-Level Period Register (RMIN/I

RMIN and RMAX specify the minimum and maximum low-level period for a repeat he

- RMIN

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RMIN7 to RMIN0	H'00	R/W	Specifies the minimum low-level period for header.

- RMAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RMAX7 to RMAX0	H'00	R/W	Specifies the maximum low-level period fo header.

Header
Address
$\overline{\text{Address}}$
Command
$\overline{\text{Command}}$
S

Figure 16.2 NEC Format

(1) Header, Address, and Command

When a 9-ms high level and the following 4.5-ms low level are detected, they are recogni header. For addresses and commands, when both of a high-level period and the following level period are 0.56 ms, they are recognized as logic 0. When a high-level period is 0.56 the following low-level period is 1.78 ms, they are recognized as logic 1.

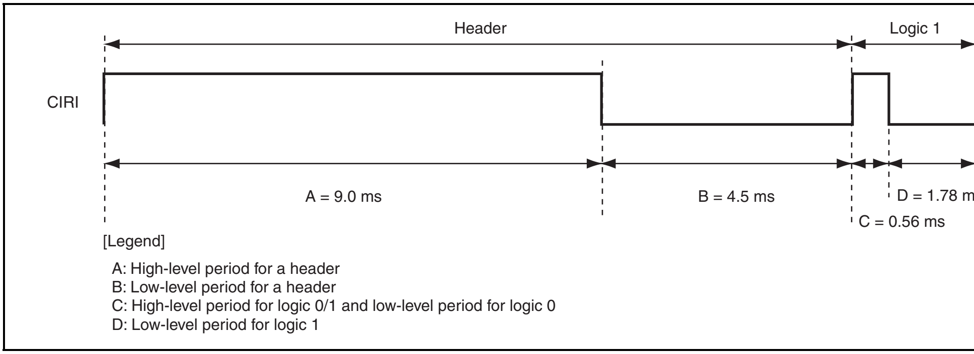

Header
Logic 1
CIRI
A = 9.0 ms
B = 4.5 ms
D = 1.78 m
C = 0.56 ms
[Legend]
A: High-level period for a header
B: Low-level period for a header
C: High-level period for logic 0/1 and low-level period for logic 0
D: Low-level period for logic 1

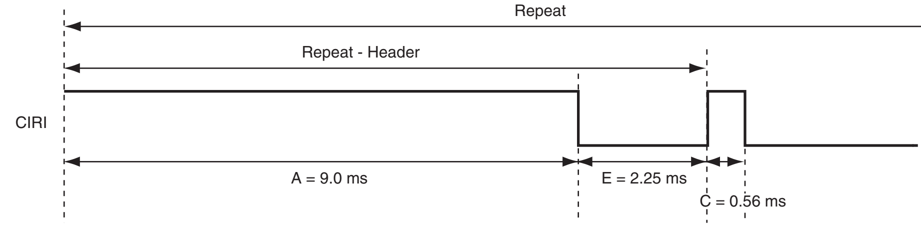
Figure 16.3 Header, Address, and Command

[Legend]
A: Low-level period for a stop
C: High-level period for logic 0/1, low-level period for logic 0, and high-level period for a stop
D: Low-level period for logic 1

Figure 16.4 Stop

(3) Repeat

When a key of the remote controller remains pressed, the command is sent only once, fo a repeat signal. When a 9-ms high level and the following 2.25-ms low level are detecte recognized as a repeat header.



[Legend]
A: High-level period for a repeat header
E: Low-level period for a repeat header
C: High-level period for a repeat

Figure 16.5 Repeat

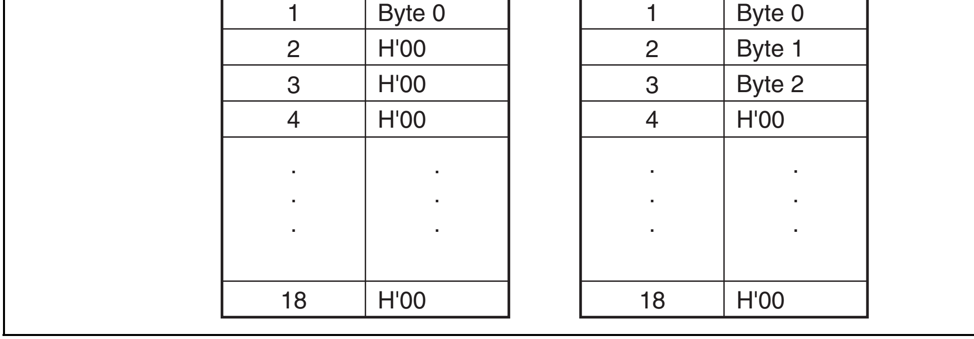
T: Frequency of the reference clock (Hz) set by the CLK1 and CLK0 bits in C
(ϕ, ϕ/2, ϕ/4, or ϕsub)

N: Setting value in BRR ($0 \le N \le 255$)

M: Value in the maximum/minimum value setting register

level period for a stop					
Minimum low-level period for a header	HLMIN	B	H'3D	3.20 ms	3.15 ms
Maximum low-level period for a header	HLMAX	B	H'6F	5.82 ms	5.85 ms
Minimum value of low/high-level period for logic 0, high-level period for logic 1, and high-level period for a burst	DT0MIN	C	H'07	0.37 ms	0.39 ms
Maximum value of low/high-level period for logic 0, high-level period for logic 1, and high-level period for a burst	DT0MAX	C	H'0D	0.68 ms	0.73 ms
Minimum low-level period for logic 1	DT1MIN	D	H'0F	0.78 ms	0.78 ms
Maximum low-level period for logic 1	DT1MAX	D	H'1B	1.42 ms	1.46 ms
Minimum low-level period for a repeat header	RMIN	E	H'1F	1.62 ms	1.58 ms
Maximum low-level period for a repeat header	RMAX	E	H'37	2.88 ms	2.92 ms

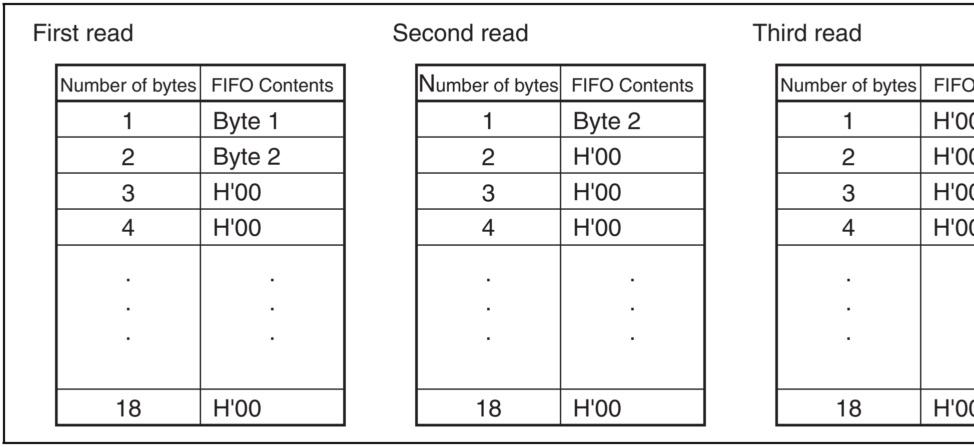
Note: The above table shows the values when the system clock is 10MHz, CLK1, CLK0
and BRR = H'82 (when the error is 30%).



1	Byte 0
2	H'00
3	H'00
4	H'00
. . .	. . .
18	H'00

1	Byte 0
2	Byte 1
3	Byte 2
4	H'00
. . .	. . .
18	H'00

Figure 16.6 Operation when FIFO Data is Received



First read

Number of bytes	FIFO Contents
1	Byte 1
2	Byte 2
3	H'00
4	H'00
. . .	. . .
18	H'00

Second read

Number of bytes	FIFO Contents
1	Byte 2
2	H'00
3	H'00
4	H'00
. . .	. . .
18	H'00

Third read

Number of bytes	FIFO
1	H'0
2	H'0
3	H'0
4	H'0
. . .	
18	H'0

Figure 16.7 Operation when FIFO Data is Read

- Select the subclock (ϕsub) as the operating clock for the CIR module.
- Enable the CIR header-detected interrupt.

For a transition from watch mode to high-speed mode, the CIR module generates an inte detection of a received header, in accord with the settings before the transition.

The module is released from watch mode when the interrupt is generated, and makes the to the high- or medium-speed mode.

16.4.4 Switching between System Clock and Sub Clock

If the operating clock is switched from the system clock to the subclock (ϕsub) while the module is operating, operation may not proceed correctly. To switch the operating clock to stop the CIR module (by clearing the CIRE bit) beforehand.

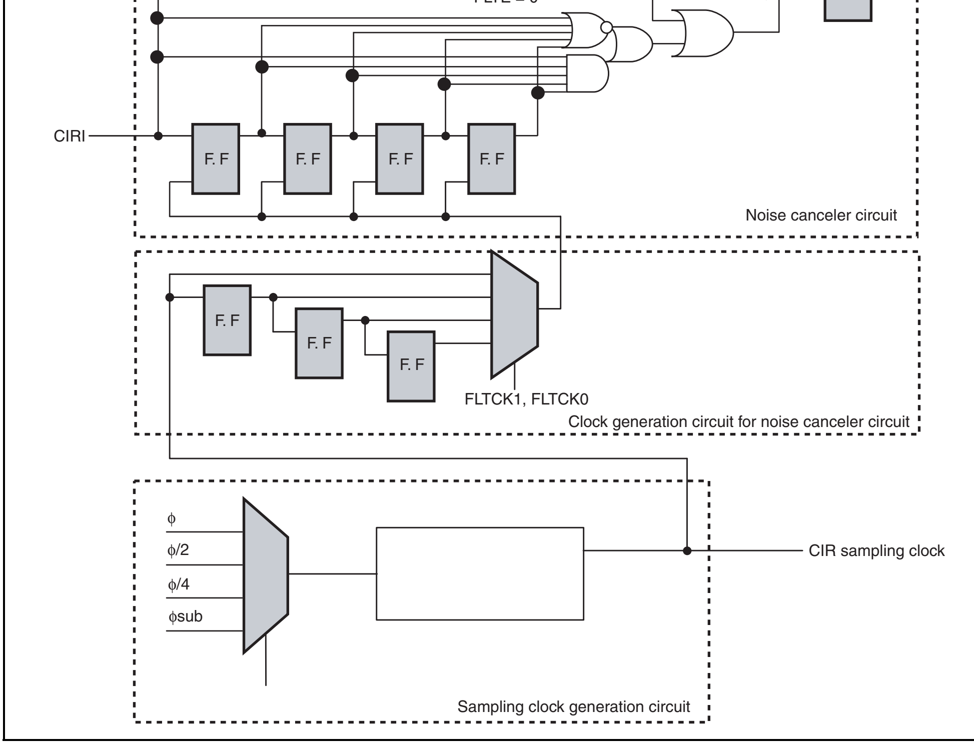


Figure 16.8 Noise Canceler Circuit

					4	64.5
			Divided by 2	25.8 µs	0	25.8
					2	77.4
					4	129
			Divided by 4	51.6 µs	0	51.6
					2	154.
					4	258
			Divided by 8	103.2 µs	0	103.
					2	309.
					4	516
—	φsub	H'00	Not divided	31.3 µs	0	31.3
					1	62.5
					2	93.8
					3	125
					4	156
			Divided by 2	62.5 µs	0	62.5
					2	187.
					4	312.
			Divided by 4	125 µs	0	125
					2	375
					4	625
			Divided by 8	250 µs	0	250
					2	750
					4	1.25

	CEIR	in HHMIN	CIRRDR	CSTR	Block
System reset	Initialized	Initialized	Initialized	Initialized	Initialized
SRES software reset	Retained	Initialized	Initialized	Initialized	Initialized
Abort	Retained	Retained	Retained	Retained * (CIRBUSY is initialized.)	Initialized

16.7 Interrupt Sources

The CIR has six interrupt source flags for this LSI. Setting the corresponding enable bit t enables the relevant interrupt request to be issued. Since the six interrupt requests are allo one vector address, it is necessary for the CPU to check the interrupt request flags in orde determine which interrupt source has caused the interrupt to be requested.

Table 16.7 Interrupt Sources

Interrupt Name	Interrupt Source Flags		Interrupt Enab
RENDI	REND	Receive end	RENDIE
OVEI	OVRF	Overrun error	OVEIE
REPI	REPF	Repeat detection	REPIE
FREI	FRF	Framing error	FREIE
ABI	ABF	Abort	ABIE
HEADFI	HEADF	Header detection	HEADFIE

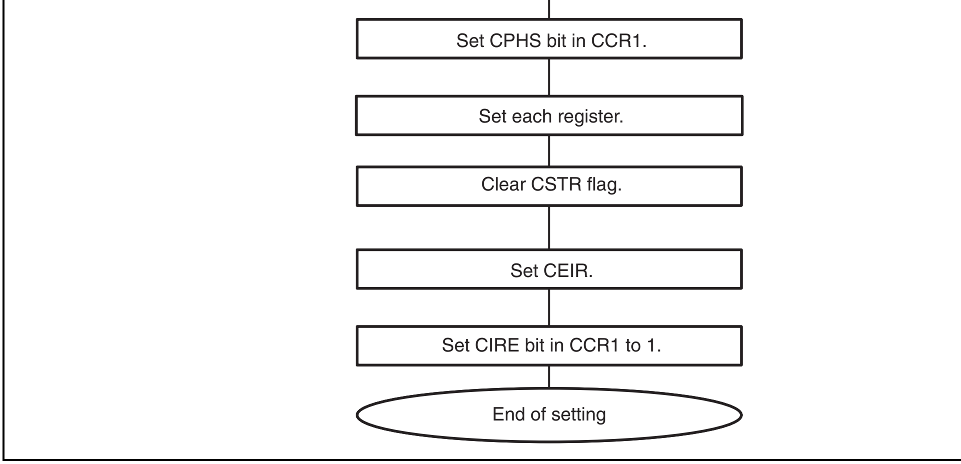


Figure 16.9 CIR Setting Flow

The CPHS bit in CCR1 should be set before starting reception. When the CIRI pin is hig
idle state, set the CPHS bit to 1. When it is low in the idle state, clear the bit to 0. The B
register is initialized to H'FF by setting the SRES bit in CCR1 to 1. After setting each re
the CIR, set the CIRE bit in CCR1 to 1 to enable the CIR reception.

(2) Switching between System Clock and Sub Clock

The CIR is capable of remote-control reception by using the sub clock in watch mode. B
switching between the system clock and the sub clock, the CIR must be stopped by clear
CIRE bit to 0.

In addition, the SCIF can be connected to the LPC interface for direct control from the I

17.1 Features

- Full-duplex communication:

  The transmitter and receiver are independent, enabling transmission and reception to executed simultaneously. Both the transmitter and receiver use 16-stage FIFO buffer enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
- Modem control function
- Data length: Selectable from 5, 6, 7, and 8 bits
- Parity: Selectable from even parity, odd parity, and no parity
- Stop bit length: Selectable from 1, 1.5, and 2 bits
- Receive error detection: Parity, overrun, and framing errors
- Break detection

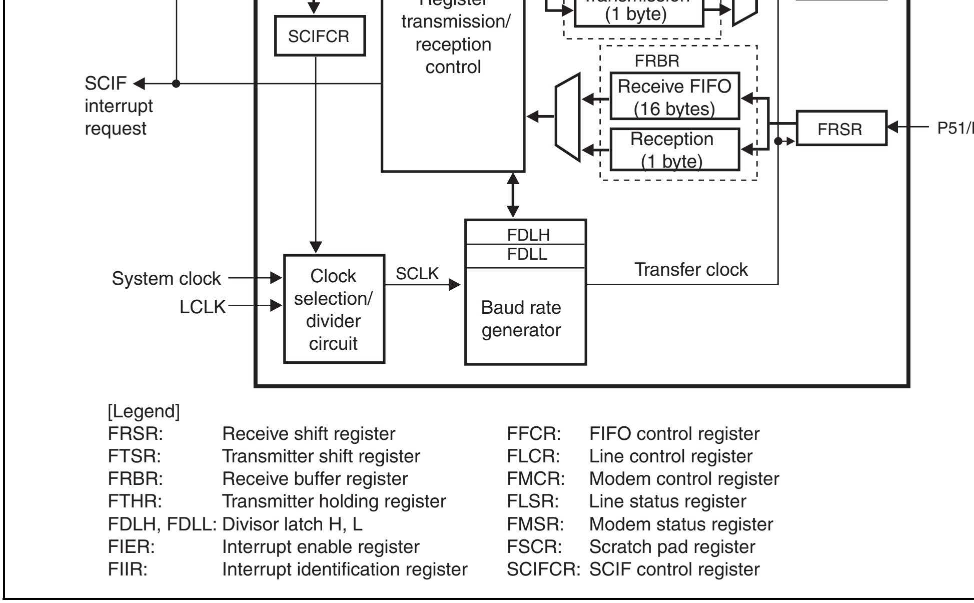

Register
transmission/
reception
control
SCIFCR
transmission
(1 byte)
FRBR
Receive FIFO
(16 bytes)
Reception
(1 byte)
FRSR
P51/
SCIF
interrupt
request
System clock
LCLK
Clock
selection/
divider
circuit
SCLK
FDLH
FDLL
Baud rate
generator
Transfer clock
[Legend]
FRSR: Receive shift register
FTSR: Transmitter shift register
FRBR: Receive buffer register
FTHR: Transmitter holding register
FDLH, FDLL: Divisor latch H, L
FIER: Interrupt enable register
FIIR: Interrupt identification register
FFCR: FIFO control register
FLCR: Line control register
FMCR: Modem control register
FLSR: Line status register
FMSR: Modem status register
FSCR: Scratch pad register
SCIFCR: SCIF control register

Figure 17.1 Block Diagram of SCIF

DCD	PB3	Input	Data carrier detect input
$\overline{\text{DSR}}$	PB4	Input	Data set ready input
$\overline{\text{DTR}}$	PB5	Output	Data terminal ready output
$\overline{\text{CTS}}$	PB6	Input	Transmission permission input
$\overline{\text{RTS}}$	PB7	Output	Transmission request output

Register Name	Abbreviation	R/W	Initial Value	Address	
Host interface control register 5	HICR5	R/W	H'00	H'FFFE33	8
Module stop control register B	MSTPCRB	R/W	H'00	H'FFFE7F	8
Receive buffer register	FRBR	R	H'00	H'FFFC20	8
Transmitter holding register	FTHR	W	—		
Divisor latch L	FDLL	R/W	H'00		
Interrupt enable register	FIER	R/W	H'00	H'FFFC21	8
Divisor latch H	FDLH	R/W	H'00		
Interrupt identification register	FIIR	R	H'01	H'FFFC22	8
FIFO control register	FFCR	W	H'00		
Line control register	FLCR	R/W	H'00	H'FFFC23	8
Modem control register	FMCR	R/W	H'00	H'FFFC24	8
Line status register	FLSR	R	H'60	H'FFFC25	8
Modem status register	FMSR	R	—	H'FFFC26	8
Scratch pad register	FSCR	R/W	H'00	H'FFFC27	8
SCIF control register	SCIFCR	R/W	H'00	H'FFFC28	8
SCIF address register H	SCIFADRH	R/W	H'03	H'FFFDC4	8
SCIF address register L	SCIFADRL	R/W	H'F8	H'FFFDC5	8
Serial IRQ control register 4	SIRQCR4	R/W	H'00	H'FFFE3B	8

17.3.1 Receive Shift Register (FRSR)

FRSR is a register that receives data and converts serial data input from the FRxD pin to data. It stores the data in the order received from the LSB (bit 0). When one frame of ser has been received, the data is transferred to FRBR.

FRSR cannot be read from the CPU/LPC interface.

17.3.2 Receive Buffer Register (FRBR)

FRBR is an 8-bit read-only register that stores received serial data. It can read data corre the DR bit in FLSR is set.

When the FIFO is disabled, the data in FRBR must be read before the next data is receiv data is received before the remaining data is read, the data is overwritten, resulting in an error.

When this register is read with the FIFO enabled, the first buffer of the receive FIFO is r When the receive FIFO becomes full, the subsequent receive data is lost, resulting in an error.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R	Stores received serial data. The data is 16 bytes when the FIFO is ena

DLAB bit in FLCR is 0. Write transmit data while the THRE bit in FLCR is set to 1.

Data can be written to FTHR when the THRE bit is set with the FIFO disabled. If data is FTHR when the THRE bit is not set, the data is overwritten.

While the THRE bit is set with the FIFO enabled, up to 16 bytes of data can be written. If written with the FIFO full, the written data is lost.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	—	W	Stores serial data to be transmitted. The data is 16 bytes when the FIFO is enal

17.3.5 Divisor Latch H, L (FDLH, FDLL)

The FDLH and FDLL are registers used to set the baud rate. They are accessible when th bit in FLCR is 1. Frequency division ranging from 1 to (2^{16} – 1) can be set with these regi The frequency divider circuit stops when both of FDLH and FDLL are 0 (initial value).

- FDLH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Upper 8 bits of divisor latch

FIER is a register that enables or disables interrupts. It is accessible when the DLAB bit
is 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved This bit is always read as 0 and cannot be
3	EDSSI	0	R/W	Modem Status Interrupt Enable 0: Modem status interrupt disabled 1: Modem status interrupt enabled
2	ELSI	0	R/W	Receive Line Status Interrupt Enable 0: Receive line status interrupt disabled 1: Receive line status interrupt enabled
1	ETBEI	0	R/W	FTHR Empty Interrupt Enable 0: FTHR empty interrupt disabled 1: FTHR empty interrupt enabled
0	ERBFI	0	R/W	Receive Data Ready Interrupt Enable A character timeout interrupt is included w FIFO is enabled. 0: Receive data ready interrupt disabled 1: Receive data ready interrupt enabled

				These bits are always read as 0 and canno modified.
3	INTID2	0	R	Interrupt ID2, ID1, ID0
2	INTID1	0	R	These bits Indicate the interrupt of the high priority among the pending interrupts.
1	INTID0	0	R	000: Modem status 001: FTHR empty 010: Receive data ready 011: Receive line status 110: Character timeout (when the FIFO is e
0	INTPEND	1	R	Interrupt Pending Indicates whether one or more interrupts ar pending. 0: Interrupt pending 1: No interrupt pending

						remaining, FIFO trigger level	receive below t level.
1	1	0	0	2	Character timeout (with FIFO enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.	FRBR
0	0	1	0	3	FTHR empty	FTHR empty	FIIR re FTHR
0	0	0	0	4 (low)	Modem status	CTS, DSR, RI, DCD	FMSR

				10: 8 bytes 11: 14 bytes
5, 4	–	—	—	Reserved These bits cannot be modified.
3	DMAMODE	0	—	DMA Mode This bit is not supported and cannot be mo
2	XMITFRST	0	W	Transmit FIFO Reset The transmit FIFO data is cleared when 1 i However, FTSR data is not cleared. This bi automatically cleared.
1	RCVRFRST	0	W	Receive FIFO Reset The receive FIFO data is cleared when 1 is However, FRSR data is not cleared. This bit is automatically cleared.
0	FIFOE	0	W	FIFO Enable 0: Transmit/receive FIFOs disabled All bytes of these FIFOs are cleared. 1: Transmit/receive FIFOs enabled

				1: FDLL and FDLH access enabled
6	BREAK	0	R/W	Break Control Generates a break by driving the serial ou FTxD low. The break state is released by clearing thi 0: Break released 1: Break generated
5	STICK PARITY	0	R	Stick Parity These bits are not supported in this LSI. These bits are always read as 0 and cann modified.
4	EPS	0	R/W	Parity Select Selects even or odd parity when the PEN 0: Odd parity 1: Even parity
3	PEN	0	R/W	Parity Enable Selects whether to add a parity bit for data transmission and whether to perform a par for data reception. 0: No parity bit added/parity check disable 1: Parity bit added/parity check enabled

				length. 00: Data length is 5 bits 01: Data length is 6 bits 10: Data length is 7 bits 11: Data length is 8 bits

17.3.10 Modem Control Register (FMCR)

FMCR controls output signals.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 1 and canno modified.
4	LOOP BACK	0	R/W	Loopback Test The transmit data output is internally conne the receive data input, and the transmit data pin (FRxD) becomes 1. The receive data inp disconnected from external sources. The fo control input pins ($\overline{\text{DSR}}$, $\overline{\text{CTS}}$, $\overline{\text{RI}}$, and $\overline{\text{DCD}}$ disconnected from external sources, and th internally connected to the four modem con signals ($\overline{\text{DTR}}$, $\overline{\text{RTS}}$, $\overline{\text{OUT1}}$, and $\overline{\text{OUT2}}$), resp The transmit data is received immediately in loopback mode. Enabling/disabling of interr by the OUT2LOOP bit in SCIFCR and FIER 0: Loopback function disabled 1: Loopback function enabled

				• Normal operation No effect on operation • Loopback test Internally connected to the $\overline{\text{RI}}$ input pin.
1	RTS	0	R/W	Request to Send Controls the $\overline{\text{RTS}}$ output. 0: $\overline{\text{RTS}}$ output is high level 1: $\overline{\text{RTS}}$ output is low level
0	DTR	0	R/W	Data Terminal Ready Controls the $\overline{\text{DTR}}$ output. 0: $\overline{\text{DTR}}$ output is high level 1: $\overline{\text{DTR}}$ output is low level

				[Clearing condition] When FRBR is read or FLSR is read while no remaining data that could cause an erro FIFO clear. 1: A receive FIFO error [Setting condition] When at least one data error (parity error, error, or break interrupt) has occurred in th
6	TEMT	1	R	Transmitter Empty Indicates whether transmit data remains. • When the FIFO is disabled 0: Transmit data remains in FTHR or FTSR [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in FTHR and F [Setting condition] When no transmit data remains in FTHR a • When the FIFO is enabled 0: Transmit data remains in the transmit FI FTSR. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in the transmit and FTSR. [Setting condition] When no transmit data remains in the trans and FTSR

				[Setting condition] When the transmit FIFO becomes empty • When the FIFO is disabled 0: Transmit data remains in FTHR. [Clearing condition] Transmit data is written to FTHR 1: No transmit data in FTHR [Setting condition] When data transfer from FTHR to FTSR i completed
4	BI	0	R	Break Interrupt Indicates detection of the receive data bre When the FIFO is enabled, a break interru in any receive data in the FIFO, and this b when the receive data is in the first FIFO Reception of the next data starts after the receive data becomes mark and a valid st received. 0: Break signal not detected [Clearing condition] FLSR read 1: Break signal detected [Setting condition] When input receive data stays at space (l for a reception time exceeding the length frame

				[Clearing condition] FLSR read 1: A framing error [Setting condition] Invalid stop bit in the receive data
2	PE	0	R	Parity Error This bit indicates a parity error in the receiv when the PEN bit in FLCR is 1. When the l enabled, this error occurs in any receive da FIFO, and this bit is set when the receive d the first FIFO buffer. 0: No parity error [Clearing condition] FLSR read If this bit is set during an overrun error, rea twice. 1: A parity error [Setting condition] Detection of parity error in receive data

				has been completed, an overrun error occ FIFO data is retained, but the last receive lost. 0: No overrun error [Clearing condition] FLSR read 1: An overrun error [Setting condition] Occurrence of an overrun error
0	DR	0	R	Data Ready Indicates that receive data is stored in FR FIFO. 0: No receive data [Clearing condition] FRBR is read or all of the FIFO data is re 1: Receive data remains. [Setting condition] Reception of data

				Indicates the inverted state of the $\overline{\text{DSR}}$ inp
4	CTS	Undefined	R	Clear to Send Indicates the inverted state of the $\overline{\text{CTS}}$ inpu
3	DDCD	0	R	Delta Data Carrier Indicator Indicates a change in the $\overline{\text{DCD}}$ input signal DDCD bit is read. 0: No change in the $\overline{\text{DCD}}$ input signal after read [Clearing condition] FMSR read 1: A change in the $\overline{\text{DCD}}$ input signal after F read [Setting condition] A change in the $\overline{\text{DCD}}$ input signal
2	TERI	0	R	Trailing Edge Ring Indicator Indicates a rise in the $\overline{\text{RI}}$ input signal after t bit is read. 0: No change in the $\overline{\text{RI}}$ input signal after FM [Clearing condition] FMSR read 1: A rise in the $\overline{\text{RI}}$ input signal after FMSR r [Setting condition] A rise in the $\overline{\text{RI}}$ input pin

				[Setting condition] A change in the $\overline{\text{DSR}}$ input signal
0	DCTS	0	R	Delta Clear to Send Indicator Indicates a change in the $\overline{\text{CTS}}$ input signa DCTS bit is read. 0: No change in the $\overline{\text{CTS}}$ input signal afte read [Clearing condition] FMSR read 1: A change in the $\overline{\text{CTS}}$ input signal after read [Setting condition] A change in the $\overline{\text{CTS}}$ input signal

17.3.13 Scratch Pad Register (FSCR)

FSCR is not used for SCIF control, but is used to temporarily store program data.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Temporarily stores program data.

4	OUT2LOOP	0	R/W	Enables or disables interrupts during a loop test. 0: Interrupt enabled 1: Interrupt disabled
3	CKSEL1	0	R/W	These bits select the clock (SCLK) to be in baud rate generator. 00: LCLK divided by 18 01: System clock divided by 11 10: Reserved for LCLK (not selectable) 11: Reserved for system clock (not selecta
2	CKSEL0	0	R/W	
1	SCIFRST	0	R/W	Resets the baud rate generator, FRSR, an 0: Normal operation 1: Reset
0	REGRST	0	R/W	Resets registers (except SCIFCR) accessi the H8S CPU or LPC interface. 0: Normal operation 1: Reset

and P50 pins are set to PORT.

CKSEL1, CKSEL0	LCLK (33 MHz) divided by 18		System Clock (20 MHz) divided by 11		System Cl (10 MHz) divide
Baud rate	FDLH, FDLL (Hex)	Error (%)	FDLH, FDLL (Hex)	Error (%)	FDLH, FDLL (Hex)
50	0900	0.54 %	0900	1.36 %	0480
75	0600	0.54 %	0600	1.36 %	0300
110	0417	0.54 %	0417	1.36 %	—
300	0180	0.54 %	0180	1.36 %	00C0
600	00C0	0.54 %	00C0	1.36 %	0060
1200	0060	0.54 %	0060	1.36 %	0030
1800	0040	0.54 %	0040	1.36 %	0020
2400	0030	0.54 %	0030	1.36 %	0018
4800	0018	0.54 %	0018	1.36 %	000C
9600	000C	0.54 %	000C	1.36 %	0006
14400	0008	0.54 %	0008	1.36 %	0004
19200	0006	0.54 %	0006	1.36 %	0003
38400	0003	0.54 %	0003	1.36 %	—
57600	0002	0.54 %	0002	1.36 %	0001
115200	0001	0.54 %	0001	1.36 %	—

enabling continuous data transmission and reception.

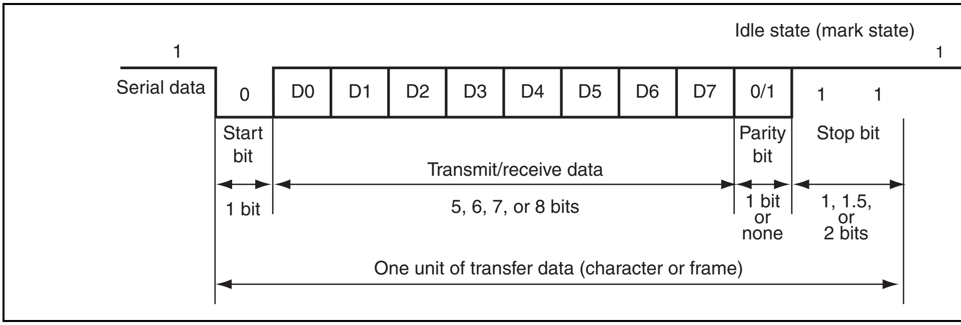
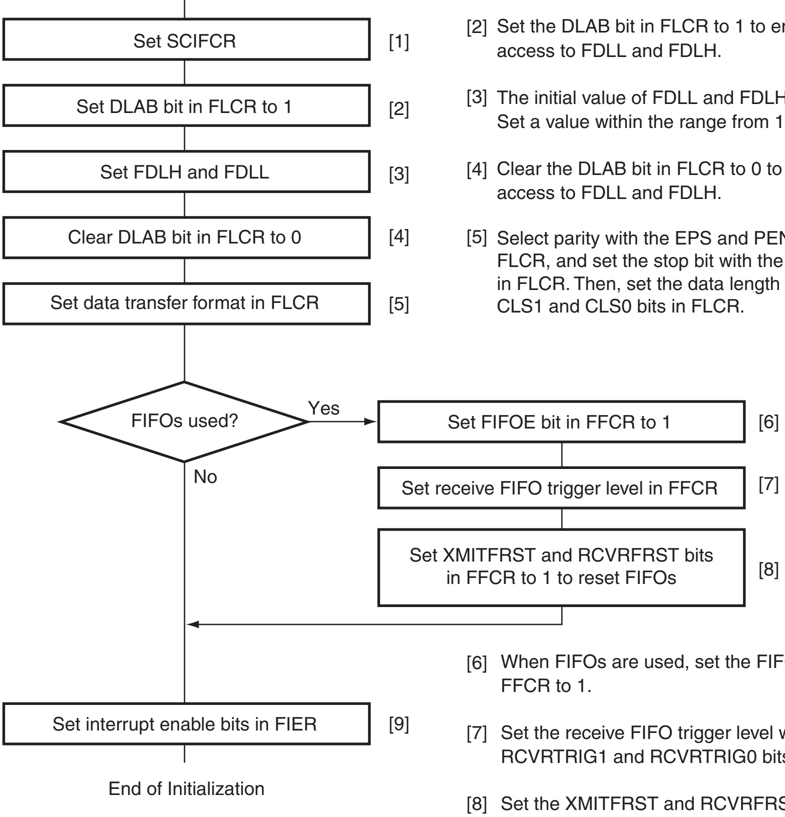


Figure 17.2 Data Format in Serial Transmission/Reception (Example with 8-Bit Data, Parity and 2 Stop Bits)


Set SCIFCR
[1]
Set DLAB bit in FLCR to 1
[2]
Set FDLH and FDLL
[3]
Clear DLAB bit in FLCR to 0
[4]
Set data transfer format in FLCR
[5]
FIFOs used?
Yes
No
Set FIFOE bit in FFCR to 1
[6]
Set receive FIFO trigger level in FFCR
[7]
Set XMITFRST and RCVRFRST bits in FFCR to 1 to reset FIFOs
[8]
Set interrupt enable bits in FIER
[9]
End of Initialization

[2] Set the DLAB bit in FLCR to 1 to enable access to FDLL and FDLH.

[3] The initial value of FDLL and FDLH is 0. Set a value within the range from 1 to 65535

[4] Clear the DLAB bit in FLCR to 0 to disable access to FDLL and FDLH.

[5] Select parity with the EPS and PEN bits in FLCR, and set the stop bit with the STOP bit in FLCR. Then, set the data length with the CLS1 and CLS0 bits in FLCR.

[6] When FIFOs are used, set the FIFOE bit in FFCR to 1.

[7] Set the receive FIFO trigger level with the RCVRTRIG1 and RCVRTRIG0 bits in FFCR

[8] Set the XMITFRST and RCVRFRST bits in FFCR to 1 to reset the FIFOs.

[9] Enable or disable an interrupt with the EDSSI, ELSI, ETBEI, and ERBFI bits in FIER and the OUT2 bit in FMCR.

Figure 17.3 Example of Initialization Flowchart

1 to ensure that all transmit data has been transmi

[3] To output a break at the end of serial transmission BREAK bit in FLCR to 1. After completion of the br clear the BREAK bit in FLCR to 0 to clear the brea

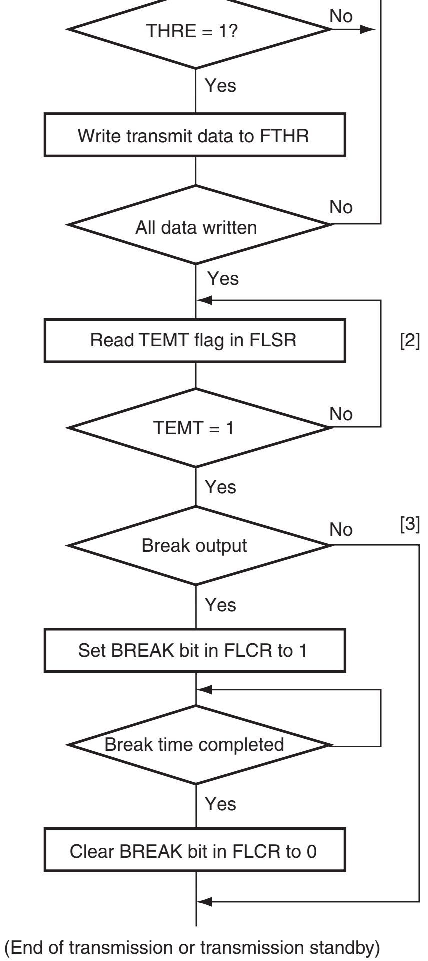
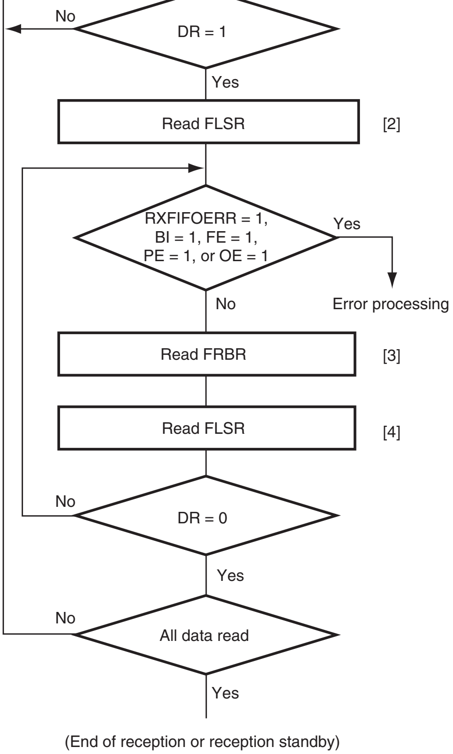

THRE = 1?
No
Yes
Write transmit data to FTHR
All data written
No
Yes
Read TEMT flag in FLSR
[2]
TEMT = 1
No
Yes
Break output
No
[3]
Yes
Set BREAK bit in FLCR to 1
Break time completed
Yes
Clear BREAK bit in FLCR to 0
(End of transmission or transmission standby)

Figure 17.4 Example of Data Transmission Flowchart

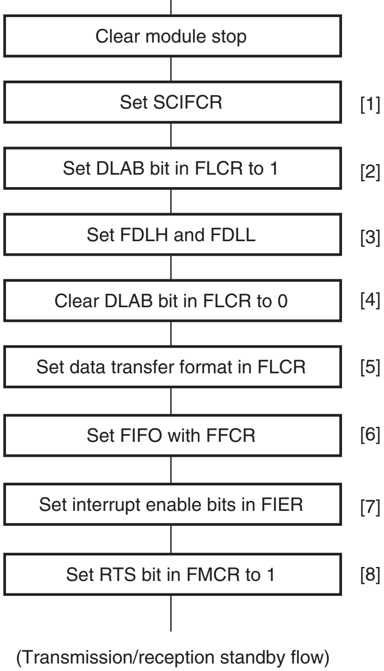


set to 1, a receive line status interrupt occurs.

[3] Read the receive data in FRBR.

[4] Check the DR flag in FLSR. When the DR flag is cleared to 0 and all data has been read, data recepti is complete.

Figure 17.5 Example of Data Reception Flowchart


Clear module stop
Set SCIFCR
[1]
Set DLAB bit in FLCR to 1
[2]
Set FDLH and FDLL
[3]
Clear DLAB bit in FLCR to 0
[4]
Set data transfer format in FLCR
[5]
Set FIFO with FFCR
[6]
Set interrupt enable bits in FIER
[7]
Set RTS bit in FMCR to 1
[8]
(Transmission/reception standby flow)

bits in SCIFCR. Set the SCIF input/output pins w
the SCIFOE1 and SCIFOE0 bits in SCIFCR.

[2] Set the DLAB bit in FLCR to 1 to enable access
FDLL and FDLH.

[3] The initial value of FDLL and FDLH is 0. Set a va
within the range from 1 to 65535.

[4] Clear the DLAB bit in FLCR to 0 to disable acces
FDLL and FDLH.

[5] Select parity with the EPS and PEN bits in FLCR
set the stop bit with the STOP bit in FLCR. Then
the data length with the CLS1 and CLS0 bits in F

[6] Set the FIFOE bit in FFCR to 1 to enable the FIF

Set the receive FIFO trigger level with the RCVR
and RCVRTRIG0 bits in FFCR. Select the best t
level to prevent an overflow of the receive FIFO.

[7] Set the EDSSI and ERBFI bits in FIER to 1 to en
modem status interrupt and receive data ready i

[8] Set the RTS bit in FMCR to 1.

Figure 17.6 Example of Initialization Flowchart

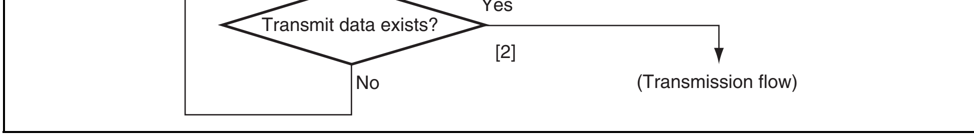


Figure 17.7 Example of Data Transmission/Reception Standby Flowchart

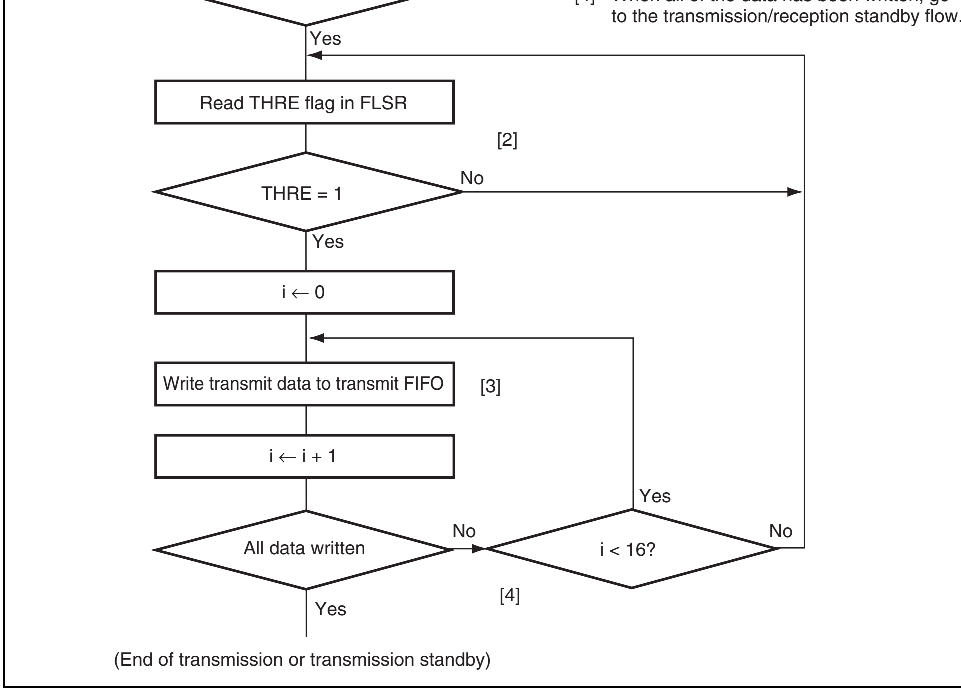
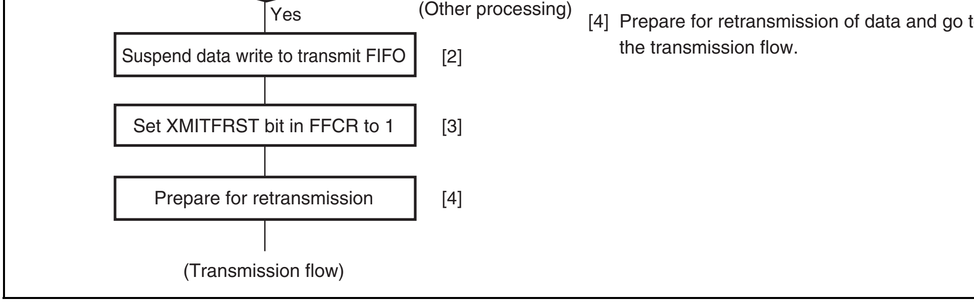
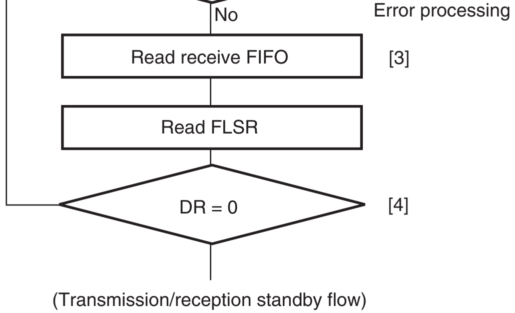

[4] When all of the data has been written, go
to the transmission/reception standby flow.
Yes
Read THRE flag in FLSR
[2]
THRE = 1
No
Yes
i ← 0
Write transmit data to transmit FIFO
[3]
i ← i + 1
Yes
All data written
No
i < 16?
No
Yes
[4]
(End of transmission or transmission standby)

Figure 17.8 Example of Data Transmission Flowchart


Yes
(Other processing)
Suspend data write to transmit FIFO
[2]
Set XMITFRST bit in FFCR to 1
[3]
Prepare for retransmission
[4]
(Transmission flow)

[4] Prepare for retransmission of data and go t the transmission flow.

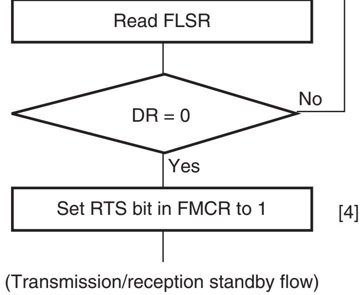
Figure 17.9 Example of Data Transmission Suspension Flowchart



[3] Read the receive FIFO.

[4] Check the DR flag in FLSR. When the D
is cleared and all of the data has been re
reception is complete.

Figure 17.10 Example of Data Reception Flowchart



[4] Set the RTS bit in FMCR to 1, and then go to the transmission/reception standby flow.

Figure 17.11 Example of Data Reception Suspension Flowchart

LPC Interface I/O Address						SCI
Bits 15 to 3	Bit 2	Bit 1	Bit 0	R/W	Condition	Reg
SCIFADR (bits 15 to 3)	0	0	0	R	FLCR[7] = 0	FRE
				W	FLCR[7] = 0	FTH
				R/W	FLCR[7] = 1	FDL
SCIFADR (bits 15 to 3)	0	0	1	R/W	FLCR[7] = 0	FIE
				R/W	FLCR[7] = 1	FDL
SCIFADR (bits 15 to 3)	0	1	0	R	—	FIIR
				W	—	FFC
SCIFADR (bits 15 to 3)	0	1	1	R/W	—	FLC
SCIFADR (bits 15 to 3)	1	0	0	R/W	—	FMC
SCIFADR (bits 15 to 3)	1	0	1	R	—	FLS
SCIFADR (bits 15 to 3)	1	1	0	R	—	FMS
SCIFADR (bits 15 to 3)	1	1	1	R/W	—	FSC

	SCSIRQ3 to 0					
SCIFCR	SCIFOE1, SCIFOE0, OUT2LOOP, CKSEL1, CKSEL0, SCIFRST, REGRST	Initialized	Retained	Retained	Retained	Retained
FRBR	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained
FTHR	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained
FDLL	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained
FDLH	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained
FIIR	FIFOE1, FIFOE0, INTID2 to INTID0, INTPEND	Initialized	Retained	Initialized	Initialized	Retained
FFCR	RCVRTRIG1, RCVRTRIG0, XMITFRST, RCVRFRST, FIFOE	Initialized	Retained	Initialized	Initialized	Retained
FLCR	DLAB, TREAK, EPS, PEN, STOP, CLS1, CLS0	Initialized	Retained	Initialized	Initialized	Retained

FSCR	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained
SCIF transmission sequencer (inner state)	—	Initialized	Initialized	Retained	Initialized	Retained

Receive line status	Overrun error, parity error, framing error, break interrupt
Receive data ready	Acceptance of receive data, FIFO trigger level
Character timeout (when FIFO is enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.
FTHR empty	FTHR empty
Modem status	CTS, DSR, RI, DCD

Table 17.10 shows the interrupt source, vector address, and interrupt priority.

Table 17.10 Interrupt Source, Vector Address, and Interrupt Priority

Interrupt		Vector	Vector
Origin of Interrupt Source	**Interrupt Name**	**Number**	**Address**
SCIF	SCIF (SCIF interrupt)	82	H'000148

17.6 Usage Note

17.6.1 Power-Down Mode When LCLK Is Selected for SCLK

To switch to watch mode or software standby mode when LCLK divided by 18 has been for SCLK, use the shutdown function of the LPC interface to stop LCLK.

17.6.2 FLCR Access During Serial Transmission and Reception

Set FLCR to its initial value and do not write to it during serial transmission or reception.

— Clocked synchronous serial format: non-addressing format without an acknowle master operation only

- Conforms to Philips I^{2}C bus interface (I^{2}C bus format)
- Two ways of setting slave address (I^{2}C bus format)
- Start and stop conditions generated automatically in master mode (I^{2}C bus format)
- Selection of the acknowledge output level in reception (I^{2}C bus format)
- Automatic loading of an acknowledge bit in transmission (I^{2}C bus format)
- Wait function in master mode (I^{2}C bus format)

— A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement.

— The wait can be cleared by clearing the interrupt flag.

- Wait function (I^{2}C bus format)

— A wait request can be generated by driving the SCL pin low after data transfer.

— The wait request is cleared when the next transfer becomes possible.

- Interrupt sources

— Data transfer end (including when a transition to transmit mode with I^{2}C bus for when ICDR data is transferred from ICDRT to ICDRS or from ICDRS to ICDR during a wait state)

— Address match: When any slave address matches or the general call address is re slave receive mode with I^{2}C bus format (including address reception after loss of arbitration)

— Arbitration lost

— Start condition detection (in master mode)

— Stop condition detection (in slave mode)

Figure 18.1 shows a block diagram of the I^{2}C bus interface. Figure 18.2 shows an exampl pin connections to external circuits. Since I^{2}C bus interface I/O pins are different in struct normal port pins, they have different specifications for permissible applied voltages. For see section 28, Electrical Characteristics.

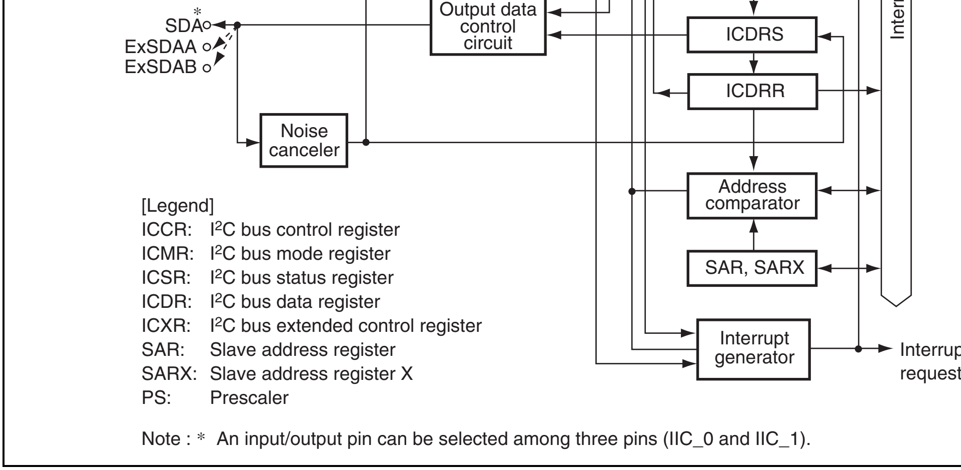

*
SDA
ExSDAA
ExSDAB
Output data control circuit
ICDRS
ICDRR
Noise canceler
Address comparator
SAR, SARX
Interrupt generator
Interrup
request
Inter
[Legend]
ICCR: I2C bus control register
ICMR: I2C bus mode register
ICSR: I2C bus status register
ICDR: I2C bus data register
ICXR: I2C bus extended control register
SAR: Slave address register
SARX: Slave address register X
PS: Prescaler
Note : * An input/output pin can be selected among three pins (IIC_0 and IIC_1).

Figure 18.1 Block Diagram of I^{2}C Bus Interface

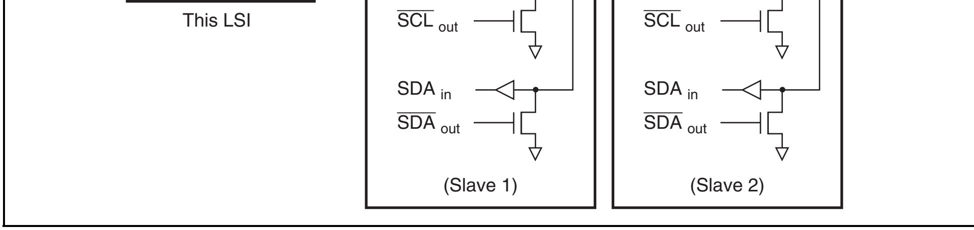


Figure 18.2 I²C Bus Interface Connections (Example: This LSI as Master)

Channel	Symbol	Input/Output	Function
0	SCL0	Input/Output	Serial clock input/output pin of IIC_0
	SDA0	Input/Output	Serial data input/output pin of IIC_0
1	SCL1	Input/Output	Serial clock input/output pin of IIC_1
	SDA1	Input/Output	Serial data input/output pin of IIC_1
2	SCL2	Input/Output	Serial clock input/output pin of IIC_2
	SDA2	Input/Output	Serial data input/output pin of IIC_2
—	ExSCLA	Input/Output	Serial clock input/output pin of IIC_0
	ExSDAA	Input/Output	Serial data input/output pin of IIC_0
	ExSCLB	Input/Output	Serial clock input/output pin of IIC_0
	ExSDAB	Input/Output	Serial data input/output pin of IIC_0

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used

Channel	Register Name	Abbreviation	R/W	Value	Address	W
Channel 0	I²C bus extended control register_0	ICXR_0	R/W	H'00	H'FED4	8
	I²C bus control register_0	ICCR_0	R/W	H'01	H'FFD8	8
	I²C bus status register_0	ICSR_0	R/W	H'00	H'FFD9	8
	I²C bus data register_0	ICDR_0	R/W	—	H'FFDE	8
	Second slave address register_0	SARX_0	R/W	H'01	H'FFDE	8
	I²C bus mode register_0	ICMR_0	R/W	H'00	H'FFDF	8
	Slave address register_0	SAR_0	R/W	H'00	H'FFDF	8
	I²C bus control initialization register_0	ICRES_0	R/W	H'0F	H'FEE6	8
Channel 1	I²C bus extended control register_1	ICXR_1	R/W	H'00	H'FED5	8
	I²C bus control register_1	ICCR_1	R/W	H'01	H'FF88 H'FED0*	8
	I²C bus status register_1	ICSR_1	R/W	H'00	H'FF89 H'FED1*	8
	I²C bus data register_1	ICDR_1	R/W	—	H'FF8E H'FECE*	8
	Second slave address register_1	SARX_1	R/W	H'01	H'FF8E H'FECE*	8
	I²C bus mode register_1	ICMR_1	R/W	H'00	H'FF8F H'FECF*	8
	Slave address register_1	SAR_1	R/W	H'00	H'FF8F H'FECF*	8

I²C bus control initialization register_2	ICRES_2	R/W	H'0F	H'FE8A

Note: * Upper address: when RELOCATE = 0
Lower address: when RELOCATE = 1

and the TRS bit is automatically changed to 1.

In transmit mode (TRS = 1), transmit data can be written to ICDRT when the ICDRE flag After the transmit data has been written to ICDRT, the ICDRE flag is cleared to 0. Then, ICDRS becomes empty on completion of the previous transmission, the data are automati transferred from ICDRT to ICDRS and the ICDRE flag is set to 1. As long as ICDRS con data to be transmitted or data being transmitted, data written to ICDRT are retained there

In receive mode (TRS = 0), data is not transferred from ICDRT to ICDRS. Thus, do not v ICDRT when in this mode.

In receive mode (TRS = 0), data received in ICDRR can be read when the ICDRF flag is the data has been read from ICDRR, the ICDRF flag is cleared to 0. Each time ICDRS co data on completion of one round of reception, the data is automatically transferred from I ICDRR and the ICDRF flag is set to 1. If ICDRR contains receive data that hasn't been re any further receive data is retained in ICDRS.

Since data are not transferred from ICDRS to ICDRR in transmit mode (TRS = 1), do not ICDRR in transmit mode (excluding the case where final receive data is read out in the recommended operation flow of master receive mode).

If the number of bits in a frame, excluding the acknowledge bit, is less than eight, transm and receive data are stored differently. Transmit data should be written justified toward th side when MLS = 0 in ICMR, and toward the LSB side when MLS = 1. Receive data bits be read from the LSB side when MLS = 0, and from the MSB side when MLS = 1.

ICDR can be written to and read from only when the ICE bit is set to 1 in ICCR. The initi of ICDR is undefined.

6	SVA5	0	R/W	Set a slave address.
5	SVA4	0	R/W	
4	SVA3	0	R/W	
3	SVA2	0	R/W	
2	SVA1	0	R/W	
1	SVA0	0	R/W	
0	FS	0	R/W	Format Select Selects the communication format together w FSX bit in SARX. See table 18.3. This bit should be set to 0 when general call a recognition is performed.

6	SVAX5	0	R/W	Set the second slave address.
5	SVAX4	0	R/W	
4	SVAX3	0	R/W	
3	SVAX2	0	R/W	
2	SVAX1	0	R/W	
1	SVAX0	0	R/W	
0	FSX	1	R/W	Format Select X Selects the communication format together wit bit in SAR. See table 18.3.

		• General call address recognized
1	0	I^2C bus format • SAR slave address ignored • SARX slave address recognized • General call address ignored
	1	Clocked synchronous serial format • SAR and SARX slave addresses ignored • General call address ignored

- I^2C bus format: addressing format with an acknowledge bit
- Clocked synchronous serial format: non-addressing format without an acknowledge master mode only

				Set this bit to 0 when the I^{2}C bus format is use
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I^{2}C format. 0: Data and the acknowledge bit are transferre consecutively with no wait inserted. 1: After the fall of the clock for the final data bit clock), the IRIC flag is set to 1 in ICCR, and state begins (with SCL at the low level). Whe IRIC flag is cleared to 0 in ICCR, the wait en the acknowledge bit is transferred. For details, see section 18.4.7, IRIC Setting Ti SCL Control.
5	CKS2	0	R/W	Transfer Clock Select 2 to 0
4	CKS1	0	R/W	These bits are used only in master mode.
3	CKS0	0	R/W	These bits select the required transfer rate, tog with the IICX2 (IIC_2), IICX1 (IIC_1), and IICX bits in STCR. See table 18.4.

I^{2}C Bus Format	Clocked Synchronous Seria
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bits
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

0	0	1	1	ϕ/64	125 kHz	156 kHz	250 kHz	313 k
0	1	0	0	ϕ/80	100 kHz	125 kHz	200 kHz	250 k
0	1	0	1	ϕ/100	80.0 kHz	100 kHz	160 kHz	200 k
0	1	1	0	ϕ/112	71.4 kHz	89.3 kHz	143 kHz	179 k
0	1	1	1	ϕ/128	62.5 kHz	78.1 kHz	125 kHz	156 k
1	0	0	0	ϕ/56	143 kHz	179 kHz	286 kHz	357 k
1	0	0	1	ϕ/80	100 kHz	125 kHz	200 kHz	250 k
1	0	1	0	ϕ/96	83.3 kHz	104 kHz	167 kHz	208 k
1	0	1	1	ϕ/128	62.5 kHz	78.1 kHz	125 kHz	156 k
1	1	0	0	ϕ/160	50.0 kHz	62.5 kHz	100 kHz	125 k
1	1	0	1	ϕ/200	40.0 kHz	50.0 kHz	80.0 kHz	100 k
1	1	1	0	ϕ/224	35.7 kHz	44.6 kHz	71.4 kHz	89.3
1	1	1	1	ϕ/256	31.3 kHz	39.1 kHz	62.5 kHz	78.1

Notes: n = 0, 1, or 2

* Correct operation cannot be guaranteed since the transfer rate is beyond the I^2 interface specification (normal mode: maximum 100 kHz, high-speed mode: m 400 kHz).

				operation, and the ports function as the SC SDA input/output pins. ICMR and ICDR ca accessed.
6	IEIC	0	R/W	I^{2}C Bus Interface Interrupt Enable 0: Disables interrupts from the I^{2}C bus interfac CPU 1: Enables interrupts from the I^{2}C bus interfac CPU.
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select MST TRS 0 0: Slave receive mode 0 1: Slave transmit mode 1 0: Master receive mode 1 1: Master transmit mode Both these bits will be cleared by hardware w lose in a bus contention in master mode with bus format. In slave receive mode with I^{2}C bu the R/$\overline{\text{W}}$ bit in the first frame immediately afte condition sets these bits in receive mode or tr mode automatically by hardware. Modification of the TRS bit during transfer is d until transfer is completed, and the changeove after completion of the transfer (at the rising e the 9th clock).

				MST clearing condition 2) [TRS clearing conditions] 1. When 0 is written by software (except for T setting condition 3) 2. When 0 is written in TRS after reading TRS TRS setting condition 3) 3. When lost in bus contention in I^{2}C bus form master mode [TRS setting conditions] 1. When 1 is written by software (except for T clearing condition 3) 2. When 1 is written in TRS after reading TRS TRS clearing condition 3) 3. When 1 is received as the R/$\overline{W}$ bit after the frame address matching in I^{2}C bus format s mode
3	ACKE	0	R/W	Acknowledge Bit Decision and Selection 0: The value of the acknowledge bit is ignored, continuous transfer is performed. The value received acknowledge bit is not indicated by ACKB bit in ICSR, which is always 0. 1: If the received acknowledge bit is 1, continu transfer is halted. Depending on the receiving device, the acknow bit may be significant, in indicating completion processing of the received data, for instance, o be fixed at 1 and have no significance.

- Writing to the BBSY flag is disabled.

[BBSY setting condition]

When the SDA level changes from high to low the condition of SCL = high, assuming that th condition has been issued.

[BBSY clearing condition]

When the SDA level changes from low to high the condition of SCL = high, assuming that th condition has been issued.

To issue a start/stop condition, use the MOV instruction.

The I^{2}C bus interface must be set in master tr mode before the issue of a start condition. Se 1 and TRS to 1 before writing 1 in BBSY and

The BBSY flag can be read to check whether bus (SCL, SDA) is busy or free.

The SCP bit is always read as 1. If 0 is written is not stored.

Note: * The value in BBSY flag does not change even if written.

[Setting conditions]

All operating modes:

1. When a start condition is detected in transm
   and the ICDRE flag is set to 1
2. When data is transferred from ICDRT to IC
   transmit mode and the ICDRE flag is set to
3. When data is transferred from ICDRS to IC
   receive mode and the ICDRF flag is set to
4. If 1 is received as the acknowledge bit (whe
   ACKE bit is 1 in transmit mode) at the comp
   data transmission

I^{2}C bus format master mode:

1. When a wait is inserted between the data a
   acknowledge bit when the WAIT bit is 1
2. When the AL flag is set to 1 after bus arbitr
   lost while the ALIE bit is 1

I^{2}C bus format slave mode:

1. When the slave address (SVA or SVAX) m
   after the reception of the first frame followin
   start condition and the AAS flag or AASX fl
   to 1
2. When the general call address is detected
   reception of the first frame following the sta
   condition and the ADZ flag is set to 1 (the F
   SAR is 0)
3. When a stop condition is detected (when th
   or ESTP flag is set to 1) while the STOPIM

the IRIC flag.

[Clearing condition]

- When 0 is written in IRIC after reading IRI

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

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
1	1	0	0	0	0	0↓	0	0↓	0↓	0	—	0	Idle state (fla required)
1	1	1↑	0	0	1↑	0	0	0	0	0	—	1↑	Start conditic
1	—	1	0	0	—	0	0	0	0	—	—	—	Wait state
1	1	1	0	0	—	0	0	0	0	1↑	—	—	Transmissior (ACKE=1 anc
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Transmissior ICDRE=0
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write w above state
1	1	1	0	0	—	0	0	0	0	0	—	1	Transmissior ICDRE=1
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write w above state c condition det
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Automatic da from ICDRT t with the abov
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Reception er ICDRF=0
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read w above state
1	0	1	0	0	—	0	0	0	0	—	1	—	Reception er ICDRF=1
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read w above state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Automatic da from ICDRS with the abov

0	1	1	0	0	—	—	—	—	0	1↑	—	—	Transmission end and ACKB=1)
0	1	1	0	0	1↑/0 *2	—	—	—	0	0	—	1↑	Transmission end ICDRE=0
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	ICDR write with th state
0	1	1	0	0	—	—	—	—	1	0		1	Transmission end ICDRE=1
0	1	1	0	0	—	—	0↓	0↓	0	0		0↓	ICDR write with th state
0	1	1	0	0	1↑/0 *2	—	0	0	0	0		1↑	Automatic data tra ICDRT to ICDRS above state
0	0	1	0	0	1↑/0 *2	—	—	—	—	—	1↑	—	Reception end wi
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with th state
0	0	1	0	0	—	—	—	—	—	—	1	—	Reception end wi
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with th state
0	0	1	0	0	1↑/0 *2	—	0	0	0	—	1↑	—	Automatic data tra ICDRS to ICDRR above state
0	—	0↓	1↑/0 *3	0/1↑ *3	—	—	—	—	—	—	—	0↓	Stop condition de

[Legend]

0: 0-state retained

1: 1-state retained

—: Previous state retained

0↓: Cleared to 0

1↑: Set to 1

Notes: 1. Set to 1 when 1 is received as a R/$\overline{\text{W}}$ bit following an address.

2. Set to 1 when the AASX bit is set to 1.

3. When ESTP=1, STOP is 0, or when STOP=1, ESTP is 0.

				[Clearing conditions] • When 0 is written in ESTP after reading E • When the IRIC flag in ICCR is cleared to 0
6	STOP	0	R/(W)*	Normal Stop Condition Detection Flag This bit is valid in I^{2}C bus format slave mode. [Setting condition] When a stop condition is detected after frame completion. [Clearing conditions] • When 0 is written in STOP after reading S • When the IRIC flag is cleared to 0
5	IRTR	0	R/(W)*	I^{2}C Bus Interface Continuous Transfer Interru Request Flag Indicates that the I^{2}C bus interface has issued interrupt request to the CPU, and the source i completion of reception/transmission of one fr When the IRTR flag is set to 1, the IRIC flag i to 1 at the same time. [Setting conditions] I^{2}C bus format slave mode: • When the ICDRE or ICDRF flag in ICDR i when AASX = 1 Master mode or clocked synchronous serial f mode with I^{2}C bus format: • When the ICDRE or ICDRF flag is set to 1 [Clearing conditions] • When 0 is written after reading IRTR = 1 • When the IRIC flag is cleared to 0 while I

				• When 0 is written in AASX after reading A • When a start condition is detected • In master mode
3	AL	0	R/(W)*	Arbitration Lost Flag Indicates that arbitration was lost in master mo [Setting conditions] When ALSL=0 • If the internal SDA and SDA pin disagree a of SCL in master transmit mode • If the internal SCL line is high at the fall of S master mode When ALSL=1 • If the internal SDA and SDA pin disagree a of SCL in master transmit mode • If the SDA pin is driven low by another devi before the I^{2}C bus interface drives the SDA after the start condition instruction was exe master transmit mode [Clearing conditions] • When ICDR is written to (transmit mode) o from (receive mode) • When 0 is written in AL after reading AL = 1

				[Clearing conditions] • When ICDR is written to (transmit mode) o from (receive mode) • When 0 is written in AAS after reading AA • In master mode
1	ADZ	0	R/(W)*	General Call Address Recognition Flag In I^2C bus format slave receive mode, this flag 1 if the first frame following a start condition is general call address (H'00). [Setting condition] When the general call address (one frame inc R/$\overline{W}$ bit is H'00) is detected in slave receive m FS = 0 or FSX = 0 [Clearing conditions] • When ICDR is written to (transmit mode) o from (receive mode) • When 0 is written in ADZ after reading AD • In master mode If a general call address is detected while FS= FSX=0, the ADZ flag is set to 1; however, the call address is not recognized (AAS flag is no

When 1 is received as the acknowledge bit wh
ACKE = 1 in transmit mode

[Clearing conditions]

- When 0 is received as the acknowledge bit
ACKE = 1 in transmit mode
- When 0 is written to the ACKE bit

Receive mode:

Sets the acknowledge data to be returned to th
transmitting device.

0: Returns 0 as acknowledge data after data re

1: Returns 1 as acknowledge data after data re

When this bit is read, the value loaded from the
line (returned by the receiving device) is read in
transmission (when TRS = 1). In reception (wh
= 0), the value set by internal software is read.

When this bit is written, acknowledge data that
returned after receiving is rewritten regardless
TRS value.

Note: When, in transmit mode, this bit has bee
overwritten by a bit manipulation instruction wit
value other than that of the ACKB flag in ICSR
value of the ACKB bit as the acknowledge data
for receive mode is overwritten by this value. T
always reset the acknowledge data when switc
receive mode.
Write 0 to the ACKE bit to clear the ACKB flag
the following cases:
in master mode—before transmission is ended
stop condition is generated; and
in slave mode—before transmission is ended a
is released to allow a master device to issue a
condition.

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

2	CLR2	1	W*	Controls initialization of the internal state of II
1	CLR1	1	W*	IIC_1. (ICRES_0)
0	CLR0	1	W*	00--: Setting prohibited 0100: Setting prohibited 0101: IIC_0 internal latch cleared 0110: IIC_1 internal latch cleared 0111: IIC_0 and IIC_1 internal latches cleared 1---: Invalid setting Controls initialization of the internal state of II (ICRES_2) 00--: Setting prohibited 0100: Setting prohibited 0101: IIC_2 internal latch cleared 0110: Setting prohibited 0111: IIC_2 internal latch cleared 1---: Invalid setting When a write operation is performed on these clear signal is generated for the internal latch the corresponding module, and the internal st IIC module is initialized. These bits can only be written to; they are alw as 1. Write data to this bit is not retained. To perform IIC clearance, bits CLR3 to CLR0 written to simultaneously using an MOV instru not use a bit manipulation instruction such as When clearing is required again, all the bits m written to in accordance with the setting.

Note: * This bit is always read as 1.

				0: Enables IRIC flag setting and interrupt gene when the stop condition is detected (STOP = ESTP = 1) in slave mode. 1: Disables IRIC flag setting and interrupt gene when the stop condition is detected.
6	HNDS	0	R/W	Enables or disables handshake control in recei for the selection of reception with handshaking 0: Disables handshake control 1: Enables handshake control Note: When the IIC module is in use, be sure t bit to 1. When the HNDS bit is cleared to 0 and a round reception is completed with ICDRR empty (the flag is 0), successive reception will proceed wit next round of reception. At the same time, a cl continuously supplied over the SCL line. In this case, the sequence of operations shoul such that unnecessary clock cycles are not out the bus after reception of the last of the data. When the HNDS bit is set to 1, SCL is fixed lo clock output stops on completion of reception. released and reception of the next frame is en reading the receive data from ICDR.

- When data is received successfully and tr from ICDRS to ICDRR.

(1) When data is received successfully while 0 (at the rise of the 9th clock pulse).

(2) When ICDR is read successfully in receiv after data was received while ICDRF = 1.

[Clearing conditions]

- When ICDR (ICDRR) is read.
- When 0 is written to the ICE bit.
- When the IIC is internally initialized using to CLR0 bits in DDCSWR.

When ICDRF is set due to the condition (2) al ICDRF is temporarily cleared to 0 when ICDR is read; however, since data is transferred fro to ICDRR immediately, ICDRF is set to 1 aga

Note that ICDR cannot be read successfully in mode (TRS = 1) because data is not transferr ICDRS to ICDRR. Be sure to read data from I receive mode (TRS = 0).

[Setting conditions]

- When the start condition is detected from th
  line state with I^{2}C bus format or serial forma
- When data is transferred from ICDRT to IC
  1. When data transmission completed whi
     ICDRE = 0 (at the rise of the 9th clock p
  2. When data is written to ICDR in transm
     after data transmission was completed
     ICDRE = 1.

[Clearing conditions]

- When data is written to ICDR (ICDRT).
- When the stop condition is detected with I^{2}C
  format or serial format.
- When 0 is written to the ICE bit.
- When the IIC is internally initialized using th
  to CLR0 bits in DDCSWR.

Note that if the ACKE bit is set to 1 with I^{2}C bus
thus enabling acknowledge bit decision, ICDRE
set when data transmission is completed while
acknowledge bit is 1.

When ICDRE is set due to the condition (2) ab
ICDRE is temporarily cleared to 0 when data is
to ICDR (ICDRT); however, since data is trans
from ICDRT to ICDRS immediately, ICDRE is s
again. Do not write data to ICDR when TRS =
because the ICDRE flag value is invalid during
time.

				IIC bus interface outputs at the rise of SCL the SCL pin is driven low by another devic 1: When the SDA pin state disagrees with the IIC bus interface outputs at the rise of SCL the SDA line is driven low by another devic state or after the start condition instruction executed.
1	FNC1	0	R/W	Function 1, 0
0	FNC0	0	R/W	These bits cancel some restrictions on usage FNC0 FNC1 0 0: Restrictions on operation canceled 0 1: Setting prohibited 1 0: Setting prohibited 1 1: Restrictions on operation remaining Note: When the IIC module is used, make sur both of the bits to 1.

18.4.

Figure 18.5 shows the I^{2}C bus timing.

The symbols used in figures 18.3 to 18.5 are explained in table 18.7.

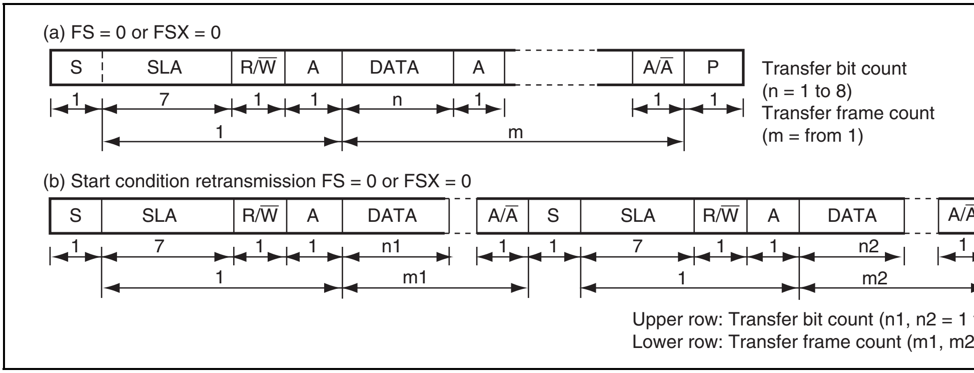


Figure 18.3 I^{2}C Bus Data Format (I^{2}C Bus Format)

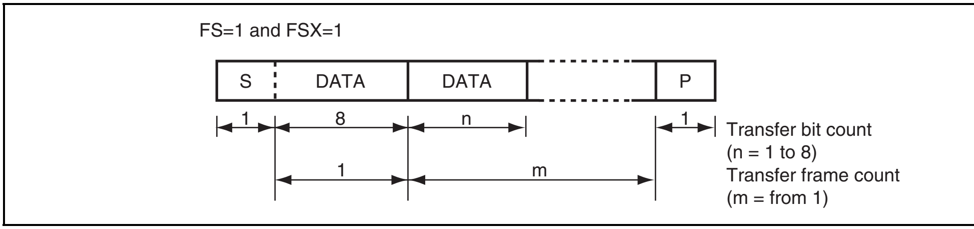


Figure 18.4 I^{2}C Bus Data Format (Serial Format)

S	Start condition. The master device drives SDA from high to low while SCL is
SLA	Slave address. The master device selects the slave device.
$R/\overline{W}$	Indicates the direction of data transfer: from the slave device to the master d when $R/\overline{W}$ is 1, or from the master device to the slave device when $R/\overline{W}$ is 0
A	Acknowledge. The receiving device drives SDA low to acknowledge a transf slave device returns acknowledge in master transmit mode, and the master returns acknowledge in master receive mode.)
DATA	Transferred data. The bit length of transferred data is set with the BC2 to BC ICMR. The MSB first or LSB first is switched with the MLS bit in ICMR.
P	Stop condition. The master device drives SDA from low to high while SCL is

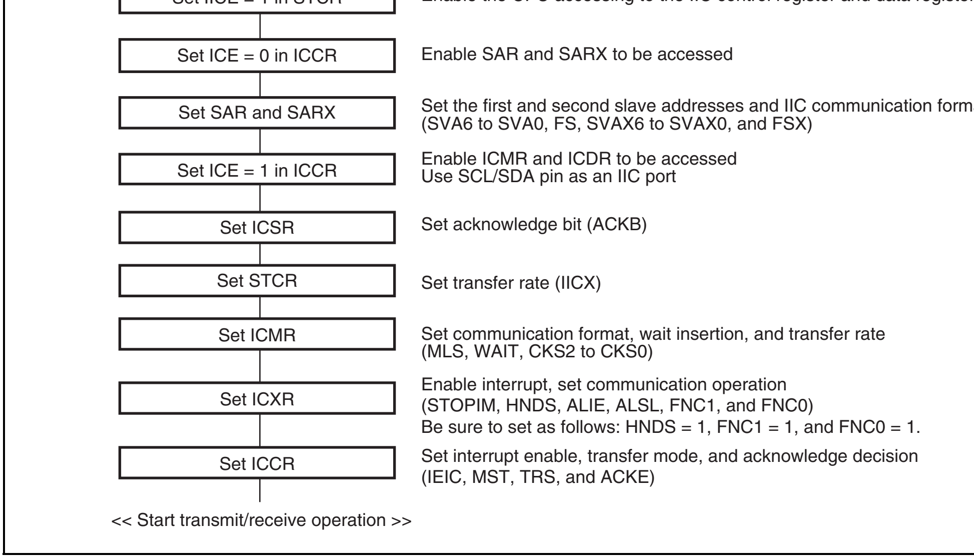

Set IICE = 1 in STCR
Enable the CPU accessing to the IIC control register and data register
Set ICE = 0 in ICCR
Enable SAR and SARX to be accessed
Set SAR and SARX
Set the first and second slave addresses and IIC communication form
(SVA6 to SVA0, FS, SVAX6 to SVAX0, and FSX)
Set ICE = 1 in ICCR
Enable ICMR and ICDR to be accessed
Use SCL/SDA pin as an IIC port
Set ICSR
Set acknowledge bit (ACKB)
Set STCR
Set transfer rate (IICX)
Set ICMR
Set communication format, wait insertion, and transfer rate
(MLS, WAIT, CKS2 to CKS0)
Set ICXR
Enable interrupt, set communication operation
(STOPIM, HNDS, ALIE, ALSL, FNC1, and FNC0)
Be sure to set as follows: HNDS = 1, FNC1 = 1, and FNC0 = 1.
Set ICCR
Set interrupt enable, transfer mode, and acknowledge decision
(IEIC, MST, TRS, and ACKE)
<< Start transmit/receive operation >>

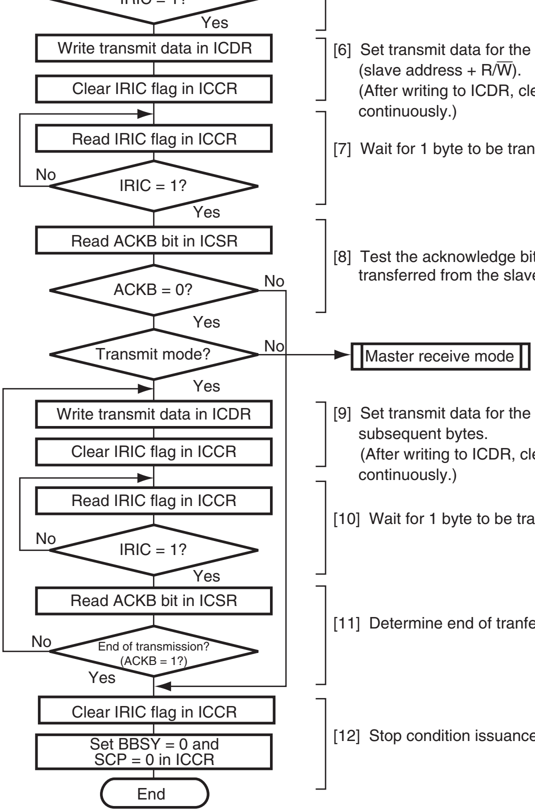
Figure 18.6 Sample Flowchart for IIC Initialization

Note: Be sure to modify the ICMR register after transmit/receive operation has been co
If the ICMR register is modified during transmit/receive operation, bit counter BC
BC0 will be modified erroneously, thus causing incorrect operation.

18.4.3 Master Transmit Operation

In I^{2}C bus format master transmit mode, the master device outputs the transmit clock and data, and the slave device returns an acknowledge signal.

Figure 18.7 shows the sample flowchart for the operations in master transmit mode.



[6] Set transmit data for the first byte (slave address + R/$\overline{W}$). (After writing to ICDR, clear IRIC flag continuously.)

[7] Wait for 1 byte to be transmitted.

[8] Test the acknowledge bit transferred from the slave device.

[9] Set transmit data for the second and subsequent bytes. (After writing to ICDR, clear IRIC flag continuously.)

[10] Wait for 1 byte to be transmitted.

[11] Determine end of tranfer

[12] Stop condition issuance

Figure 18.7 Sample Flowchart for Operations in Master Transmit Mode

With the I^{2}C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the fir data following the start condition indicates the 7-bit slave address and transmit/receiv direction (R/$\overline{\text{W}}$).

To determine the end of the transfer, the IRIC flag is cleared to 0. After writing to ICI IRIC continuously so no other interrupt handling routine is executed. If the time for transmission of one frame of data has passed before the IRIC clearing, the end of tran cannot be determined. The master device sequentially sends the transmission clock an data written to ICDR. The selected slave device (i.e. the slave device with the matchin address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge s

7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixe synchronization with the internal clock until the next transmit data is written.

8. Read the ACKB bit in ICSR to confirm that ACKB is cleared to 0. When the slave de not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and retry th transmit operation.

9. Write the transmit data to ICDR.

   As indicating the end of the transfer, the IRIC flag is cleared to 0. Perform the ICDR the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next frame performed in synchronization with the internal clock.

10. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixe synchronization with the internal clock until the next transmit data is written.

11. Read the ACKB bit in ICSR.

    Confirm that the slave device has been acknowledged (ACKB bit is 0). When there is to be transmitted, go to step [9] to continue the next transmission operation. When the device has not acknowledged (ACKB bit is set to 1), operate step [12] to end transmis

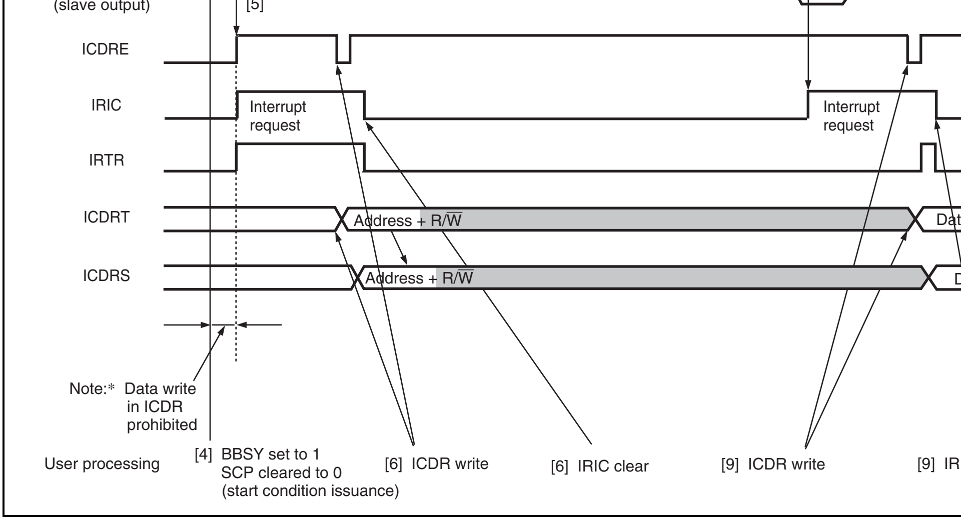


Figure 18.8 Example of Operation Timing in Master Transmit Mode (MLS = W

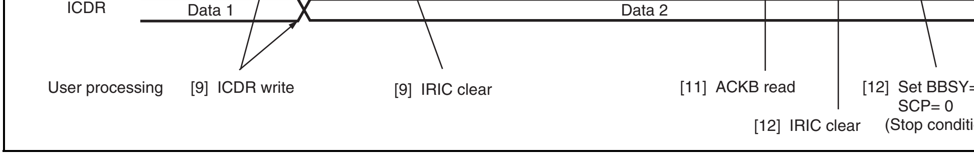


Figure 18.9 Example of Stop Condition Issuance Operation Timing in Master Transmit Mode (MLS = WAIT = 0)

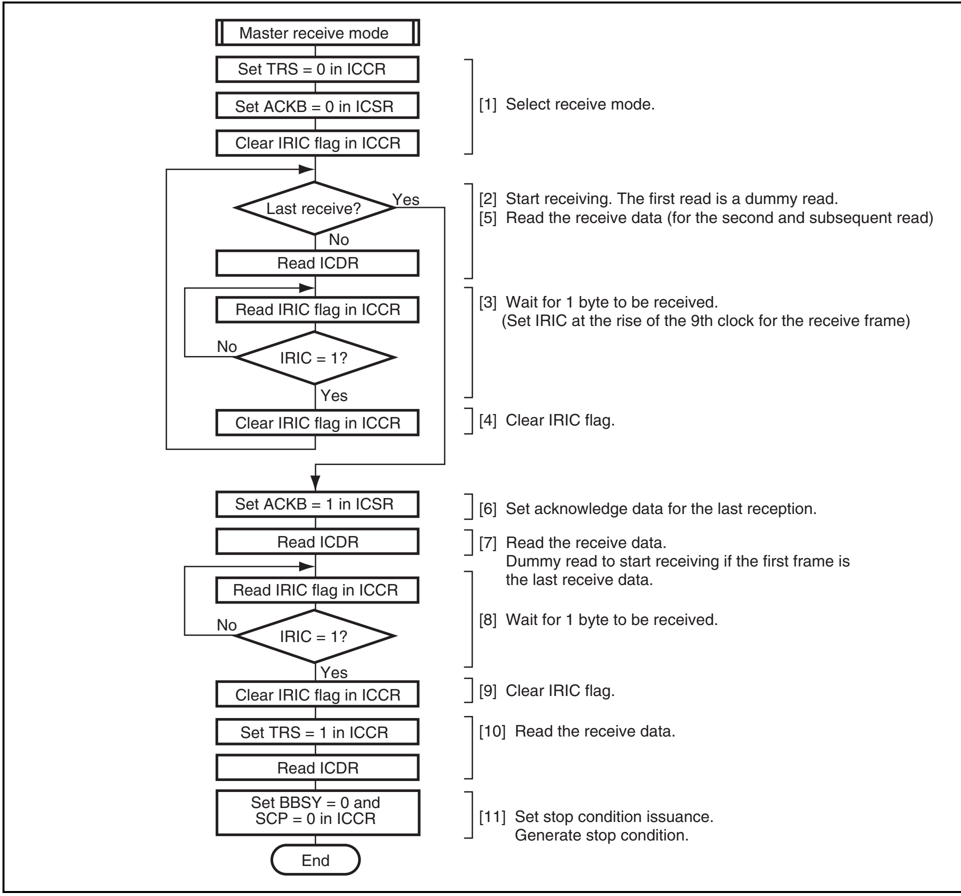

Master receive mode
Set TRS = 0 in ICCR
Set ACKB = 0 in ICSR
Clear IRIC flag in ICCR
[1] Select receive mode.
Last receive?
Yes
No
Read ICDR
[2] Start receiving. The first read is a dummy read.
[5] Read the receive data (for the second and subsequent read)
Read IRIC flag in ICCR
No
IRIC = 1?
Yes
[3] Wait for 1 byte to be received.
(Set IRIC at the rise of the 9th clock for the receive frame)
Clear IRIC flag in ICCR
[4] Clear IRIC flag.
Set ACKB = 1 in ICSR
[6] Set acknowledge data for the last reception.
Read ICDR
[7] Read the receive data.
Dummy read to start receiving if the first frame is the last receive data.
Read IRIC flag in ICCR
No
IRIC = 1?
Yes
[8] Wait for 1 byte to be received.
Clear IRIC flag in ICCR
[9] Clear IRIC flag.
Set TRS = 1 in ICCR
Read ICDR
[10] Read the receive data.
Set BBSY = 0 and SCP = 0 in ICCR
[11] Set stop condition issuance.
Generate stop condition.
End

Figure 18.10 Sample Flowchart for Operations in Master Receive Mode

pulse. The receive data is transferred from ICDRS to ICDRR at the rise of the 9th cloc
setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set to 1, an inte
request is sent to the CPU.

The master device drives SCL low from the fall of the 9th receive clock pulse to the I
reading.

4. Clear the IRIC flag to determine the next interrupt.

   Go to step [6] to halt reception operation if the next frame is the last receive data.

5. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs t
   receive clock continuously to receive the next data.

Data can be received continuously by repeating steps [3] to [5].

6. Set the ACKB bit to 1 so as to return the acknowledge data for the last reception.
7. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs t
   receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are set t
   rise of the 9th receive clock pulse.
9. Clear the IRIC flag to 0.
10. Read ICDR receive data after setting the TRS bit. This clears the ICDRF flag to 0.
11. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high w
    is high, and generates the stop condition.

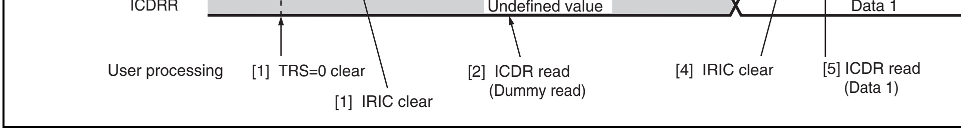


Figure 18.11 Example of Operation Timing in Master Receive Mode (MLS = WAIT = 0)

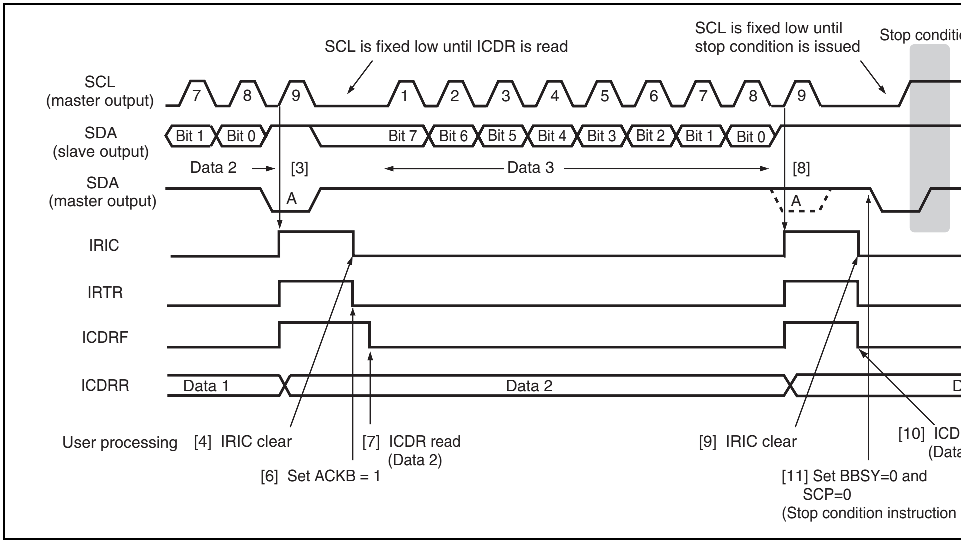


Figure 18.12 Example of Stop Condition Issuance Operation Timing in Master Receive Mode (MLS = WAIT = 0)

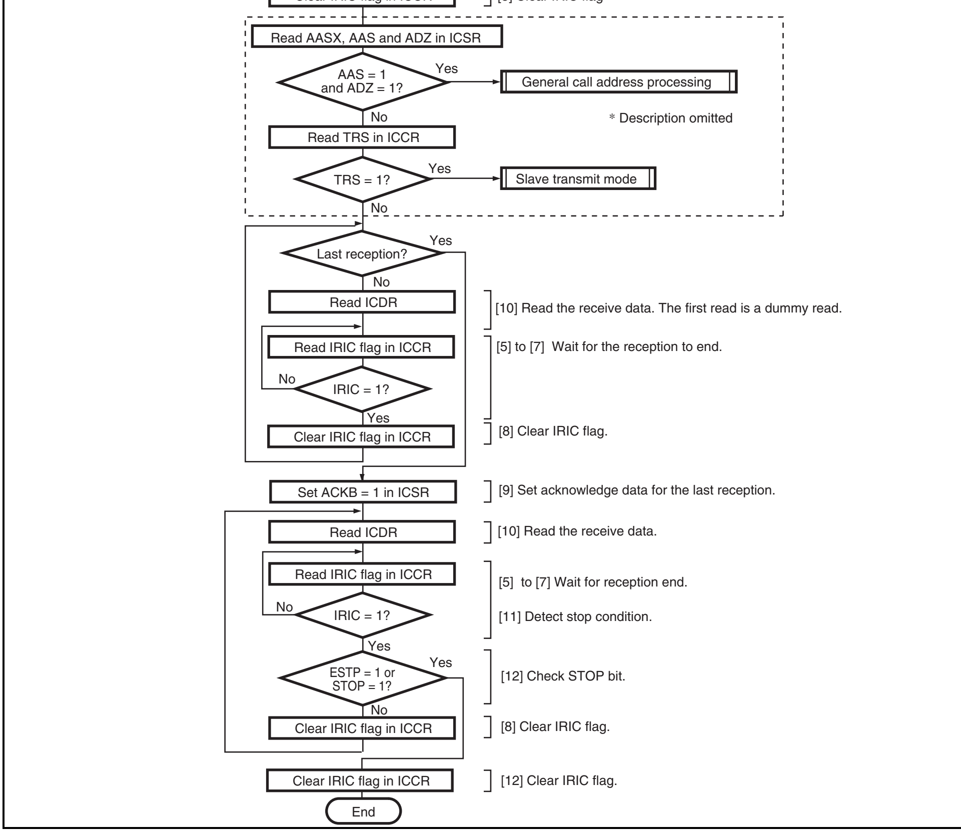


Figure 18.13 Sample Flowchart for Operations in Slave Receive Mode

4. When the slave address matches in the first frame following the start condition, the de
operates as the slave device specified by the master device. If the 8th data bit ($R/\overline{W}$) is
TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th data
($R/\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When th
address does not match, receive operation is halted until the next start condition is det
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the AC
as an acknowledge signal.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set
interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDR
setting the ICDRF flag to 1. The slave device drives SCL low from the fall of the 9th
clock pulse until data is read from ICDR.
8. Confirm that the STOP bit is cleared to 0, and clear the IRIC flag to 0.
9. If the next frame is the last receive frame, set the ACKB bit to 1.
10. If ICDR is read, the ICDRF flag is cleared to 0, releasing the SCL bus line. This enab
master device to transfer the next data.

Receive operations can be performed continuously by repeating steps [5] to [10].

11. When the stop condition is detected (SDA is changed from low to high when SCL is h
BBSY flag is cleared to 0 and the STOP bit is set to 1. If the STOPIM bit has been cle
0, the IRIC flag is set to 1.
12. Confirm that the STOP bit is set to 1, and clear the IRIC flag to 0.

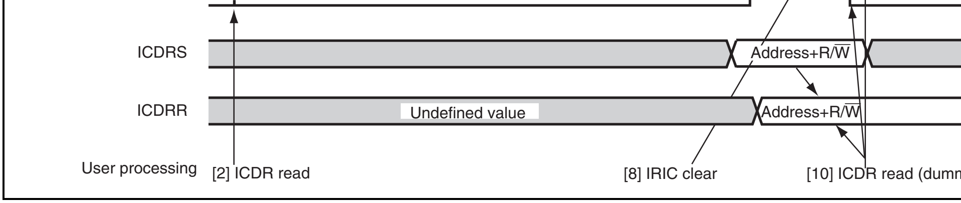


Figure 18.14 Example of Slave Receive Mode Operation Timing (1) (MLS =

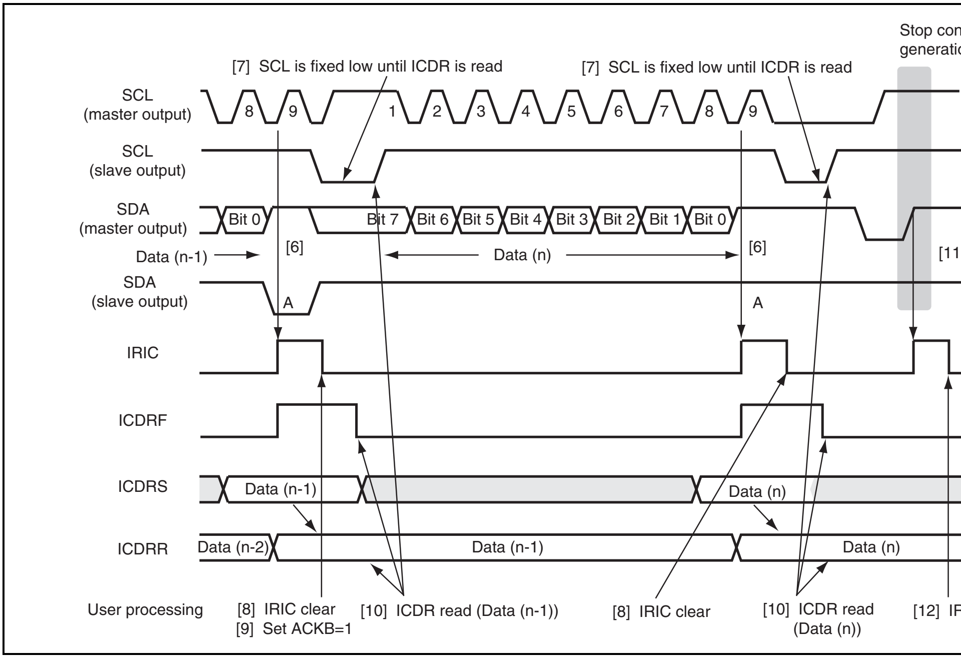
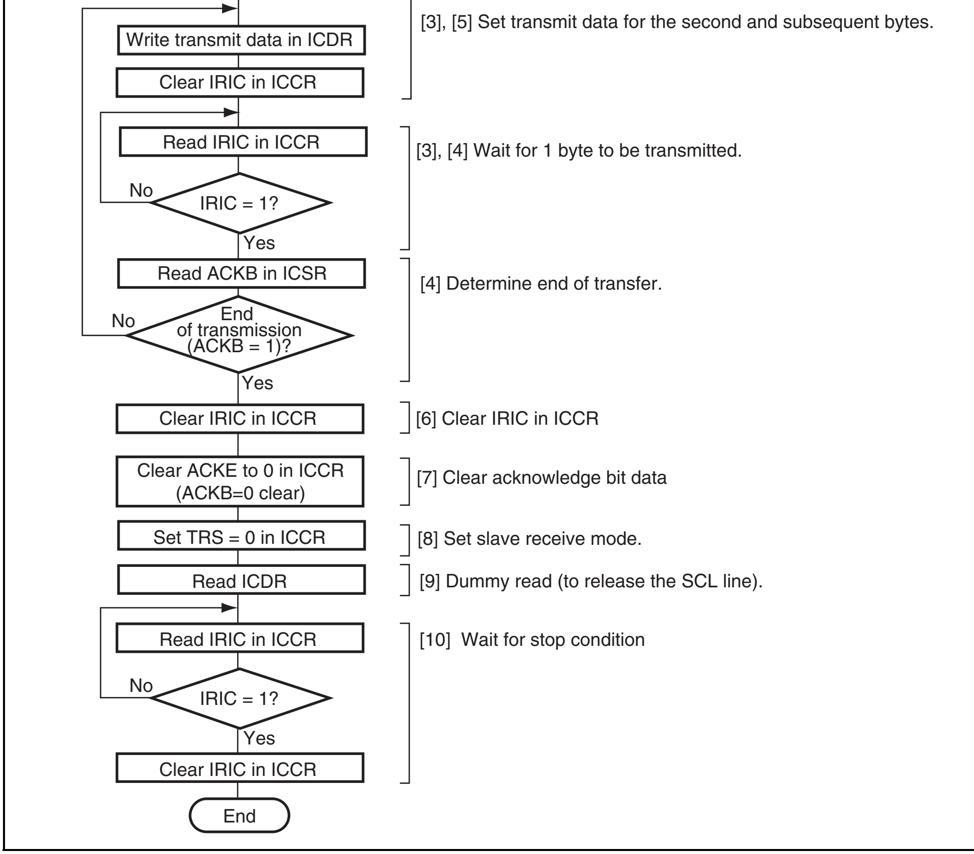


Figure 18.15 Example of Slave Receive Mode Operation Timing (2) (MLS =



Write transmit data in ICDR
Clear IRIC in ICCR
[3], [5] Set transmit data for the second and subsequent bytes.
Read IRIC in ICCR
No
IRIC = 1?
Yes
[3], [4] Wait for 1 byte to be transmitted.
Read ACKB in ICSR
No
End of transmission (ACKB = 1)?
Yes
[4] Determine end of transfer.
Clear IRIC in ICCR
[6] Clear IRIC in ICCR
Clear ACKE to 0 in ICCR (ACKB=0 clear)
[7] Clear acknowledge bit data
Set TRS = 0 in ICCR
[8] Set slave receive mode.
Read ICDR
[9] Dummy read (to release the SCL line).
Read IRIC in ICCR
No
IRIC = 1?
Yes
Clear IRIC in ICCR
[10] Wait for stop condition
End

Figure 18.16 Sample Flowchart for Slave Transmit Mode

until ICDR data is written, to disable the master device to output the next transfer cl

3. After clearing the IRIC flag to 0, write data to ICDR. At this time, the ICDRE flag is 0. The written data is transferred to ICDRS, and the ICDRE and IRIC flags are set to The slave device sequentially sends the data written into ICDRS in accordance with output by the master device.

   The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the I register writing to the IRIC flag clearing should be performed continuously. Prevent interrupt processing from being inserted.

4. The master device drives SDA low at the 9th clock pulse, and returns an acknowled As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed successfully. When one fra has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to ICDR transmission starts, and the ICDRE and IRIC flags are set to 1 again. If the ICDRE f been set to 1, this slave device drives SCL low from the fall of the 9th transmit clock is written to ICDR.

5. To continue transmission, write the next data to be transmitted into ICDR. The ICD cleared to 0. The IRIC flag is cleared to 0 to detect the end of transmission. Processi the ICDR writing to the IRIC flag clearing should be performed continuously. Preve other interrupt processing from being inserted.

Transmit operations can be performed continuously by repeating steps [4] and [5].

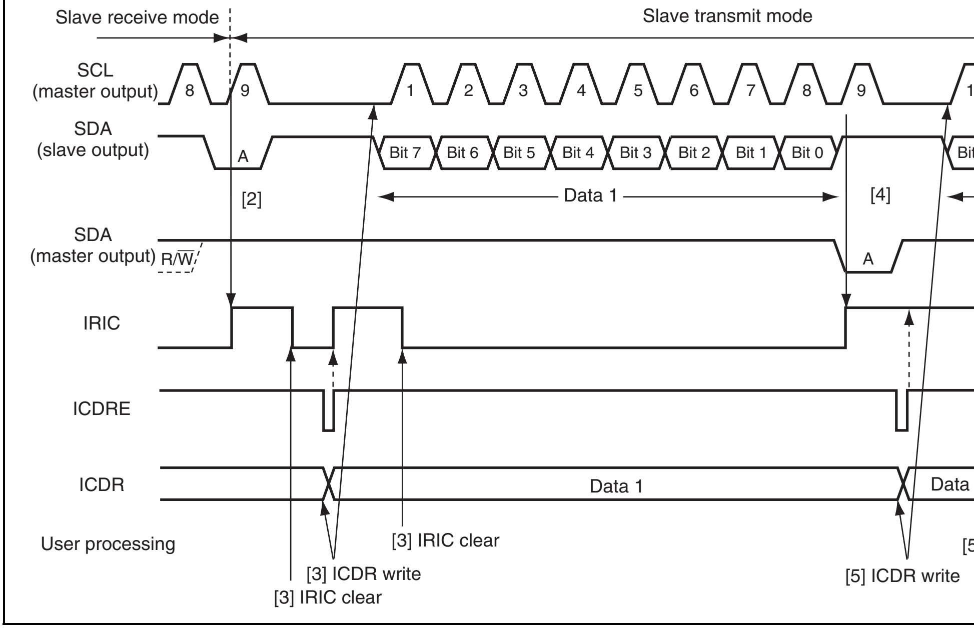
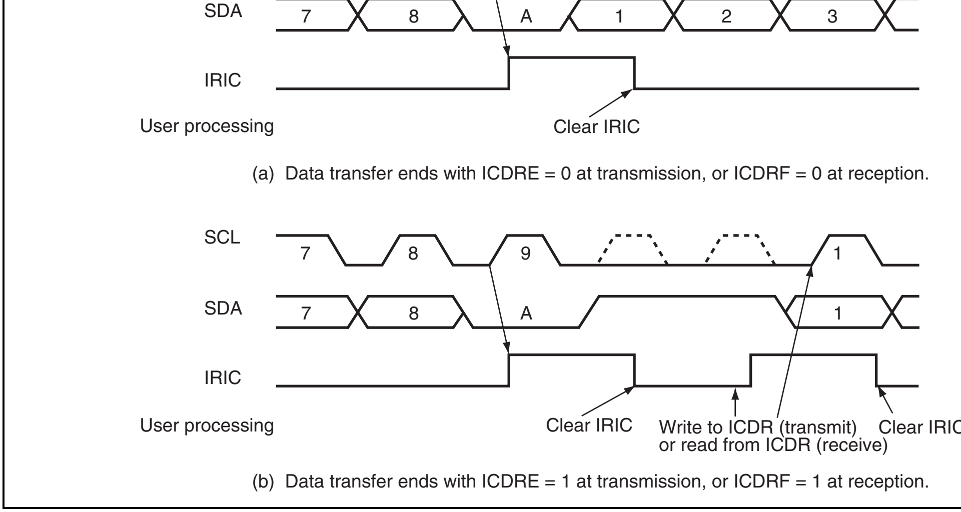


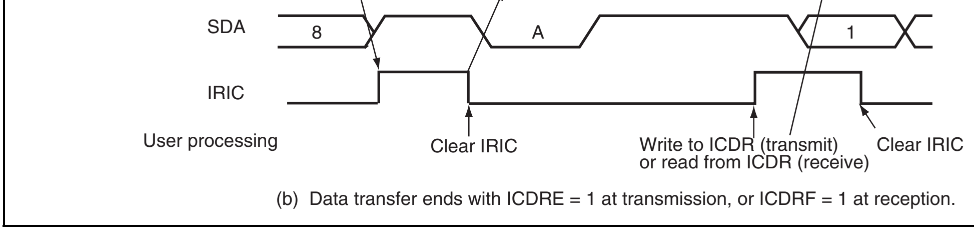
Figure 18.17 Example of Slave Transmit Mode Operation Timing (MLS = 0)



(a) Data transfer ends with ICDRE = 0 at transmission, or ICDRF = 0 at reception.

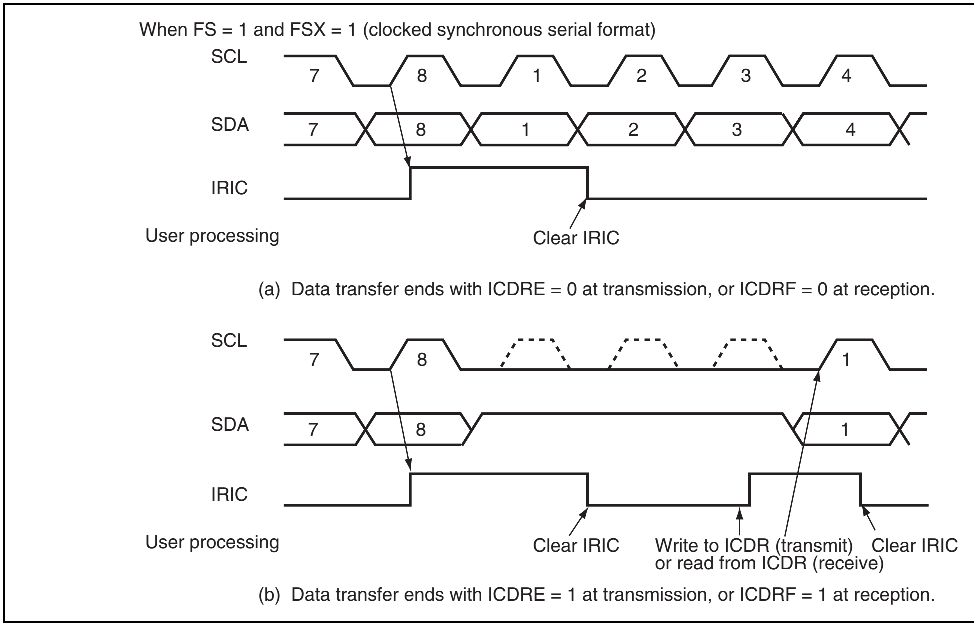
(b) Data transfer ends with ICDRE = 1 at transmission, or ICDRF = 1 at reception.

Figure 18.18 IRIC Setting Timing and SCL Control (1)



(b) Data transfer ends with ICDRE = 1 at transmission, or ICDRF = 1 at reception.

Figure 18.19 IRIC Setting Timing and SCL Control (2)


When FS = 1 and FSX = 1 (clocked synchronous serial format)

(a) Data transfer ends with ICDRE = 0 at transmission, or ICDRF = 0 at reception.

(b) Data transfer ends with ICDRE = 1 at transmission, or ICDRF = 1 at reception.

Figure 18.20 IRIC Setting Timing and SCL Control (3)

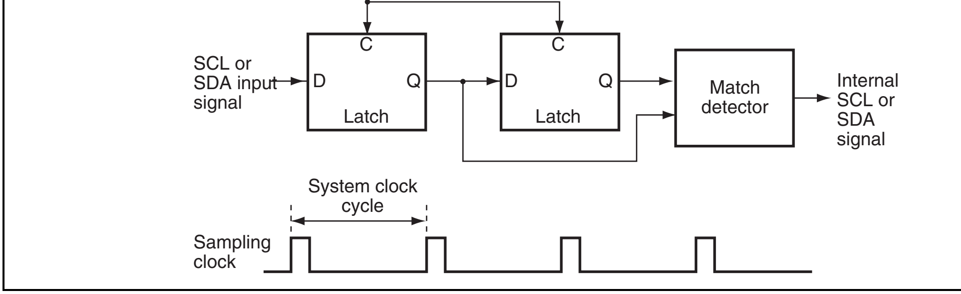


Figure 18.21 Block Diagram of Noise Canceler

18.4.9 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs communication.

Initialization is executed in accordance with the setting of bits CLR3 to CLR0 in ICRES clearing ICE bit. For details on the setting of bits CLR3 to CLR0, see section 18.3.7, I^{2}C Control Initialization Register (ICRES).

(1) Scope of Initialization

The initialization executed by this function covers the following items:

- ICDRE and ICDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, output, etc.)

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures mu taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures mu taken as necessary.
- When initialization is executed by ICRES, the write data for bits CLR3 to CLR0 is no retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneo using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.
- Similarly, when clearing is required again, all the bits must be written to simultaneous accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will st transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but stop condition pin waveform is generated according to the state and release timing of the SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bit flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used wh initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CL ICE bit clearing.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the bit to 0, and wait for two transfer rate clock cycles.
3. Re-execute initialization of the internal state according to the setting of bits CLR3 to ICE bit clearing.
4. Initialize (re-set) the IIC registers.

0	IICI0	IEIC	I^2C bus interface interrupt request	IRIC	H
1	IICI1	IEIC	I^2C bus interface interrupt request	IRIC	
2	IICI2	IEIC	I^2C bus interface interrupt request	IRIC	L

ICDRR)

3. Table 18.9 shows the timing of SCL and SDA outputs in synchronization with the int clock. Timings on the bus are determined by the rise and fall times of signals affected bus load capacitance, series resistance, and parallel resistance.

Table 18.9 I²C Bus Timing (SCL and SDA Outputs)

Item	Symbol	Output Timing	Unit	Not
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	See
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	28. refe
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	—
Data output setup time (slave)		$1t_{SCLL}$ – ($6t_{cyc}$ or $12t_{cyc}$*)		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: * $6t_{cyc}$ when IICX is 0, $12t_{cyc}$ when 1.

IICX	t_{cyc} Indication		I²C Bus Specification (Max.)	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 2
0	7.5 t_{cyc}	Standard mode	1000	937	750	468	3
		High-speed mode	300	300	300	300	3
1	17.5 t_{cyc}	Standard mode	1000	1000	1000	1000	8
		High-speed mode	300	300	300	300	3

5. The I^{2}C bus interface specifications for the SCL and SDA rise and fall times are und and 300 ns. The I^{2}C bus interface SCL and SDA output timing is prescribed by t_{cyc}, a table 18.11. However, because of the rise and fall times, the I^{2}C bus interface specifi may not be satisfied at the maximum transfer rate. Table 18.11 shows output timing calculations for different operating frequencies, including the worst-case influence o fall times.

   t_{BUFO} fails to meet the I^{2}C bus interface specifications at any frequency. The solution i to provide coding to secure the necessary interval (approximately 1 µs) between issu stop condition and issuance of a start condition, or (b) to select devices whose input permits this output timing for use as slave devices connected to the I^{2}C bus.

   t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I^{2}C bus interfa specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resisto capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecti whose input timing permits this output timing for use as slave devices connected to t bus.

t_{BUFO}	$0.5\ t_{SCLO} –1\ t_{cyc}$ (–t_{Sr})	Standard mode	–1000	4700	3875*1	3900*1	3939*1
		High-speed mode	–300	1300	825*1	850*1	888*1
t_{STAHO}	$0.5\ t_{SCLO} –1\ t_{cyc}$ (–t_{Sf})	Standard mode	–250	4000	4625	4650	4688
		High-speed mode	–250	600	875	900	938
t_{STASO}	$1\ t_{SCLO}$ (–t_{Sr})	Standard mode	–1000	4700	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200
t_{STOSO}	$0.5\ t_{SCLO} + 2\ t_{cyc}$ (–t_{Sr})	Standard mode	–1000	4000	4250	4200	4125
		High-speed mode	–300	600	1200	1150	1075
t_{SDASO} (master)	$1\ t_{SCLLO}$*3 $–3\ t_{cyc}$ (–t_{Sr})	Standard mode	–1000	250	3325	3400	3513
		High-speed mode	–300	100	625	700	813
t_{SDASO} (slave)	$1\ t_{SCLL}$*3 $–12\ t_{cyc}$*2 (–t_{Sr})	Standard mode	–1000	250	2200	2500	2950
		High-speed mode	–300	100	–500*1	–200*1	250
t_{SDAHO}	$3\ t_{cyc}$	Standard mode	0	0	375	300	188
		High-speed mode	0	0	375	300	188

Notes: 1. Does not meet the I^2C bus interface specification. Remedial action such as the
is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the
fall times by means of a pull-up resistor and capacitive load; (c) reduce the trar
(d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bi
CKS0 to CKS2. Depending on the frequency it may not be possible to achieve
maximum transfer rate; therefore, whether or not the I^2C bus interface specifica
met must be determined in accordance with the actual setting conditions.

2. Value when the IICX bit is set to 1. When the IICX bit is cleared to 0, the value
$6\ t_{cyc}$).

3. Calculated using the I^2C bus specification values (standard mode: 4700 ns mir
speed mode: 1300 ns min.).

address does not match. Similarly, if the start condition or address is transmitted fro master device in slave transmit mode (TRS = 1), the IRIC flag may be set after the I is set and 1 received as the acknowledge bit value (ACKB = 1), thus causing an inter source even when the address does not match.

To use the I^{2}C bus interface module in slave mode, be sure to follow the procedures

A. When having received 1 as the acknowledge bit value for the last transmit data at of a series of transmit operation, clear the ACKE bit in ICCR once to initialize th bit to 0.

B. Set receive mode (TRS = 0) before the next start condition is input in slave mode Complete transmit operation by the procedure shown in figure 18.16, in order to from slave transmit mode to slave receive mode.

18.6.1 Module Stop Mode Setting

The IIC operation can be enabled or disabled using the module stop control register. The setting is for the IIC operation to be halted. Register access is enabled by canceling mod mode. For details, see section 26, Power-Down Modes.

- Conforms to PS/2 interface specifications
- Direct bus drive (via the KCLK and KD pins)
- Interrupt sources: on completion of data reception/transmission, on detection of cloc edge, and on detection of the first falling edge of a clock
- Error detection: parity error, stop bit monitoring, and receive notify monitoring

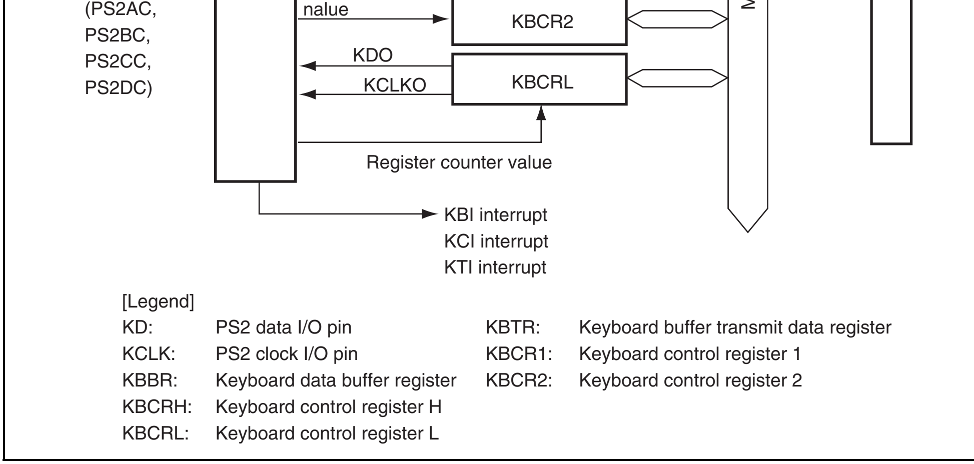

(PS2AC,
PS2BC,
PS2CC,
PS2DC)
nalue
KBCR2
KDO
KCLKO
KBCRL
Register counter value
KBI interrupt
KCI interrupt
KTI interrupt
[Legend]
KD: PS2 data I/O pin
KCLK: PS2 clock I/O pin
KBBR: Keyboard data buffer register
KBCRH: Keyboard control register H
KBCRL: Keyboard control register L
KBTR: Keyboard buffer transmit data register
KBCR1: Keyboard control register 1
KBCR2: Keyboard control register 2

Figure 19.1 Block Diagram of PS2

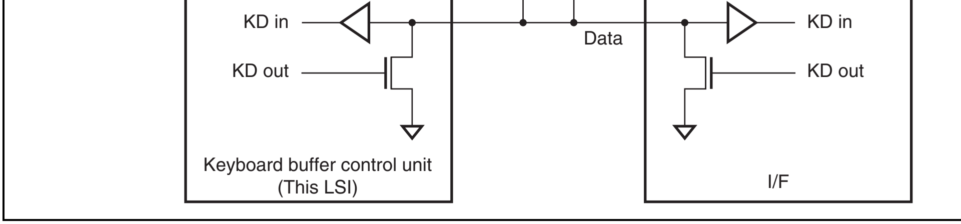


Figure 19.2 PS2 Connection

19.2 Input/Output Pins

Table 19.1 lists the input/output pins used by the keyboard buffer control unit.

Table 19.1 Pin Configuration

Channel	Name	Abbreviation*	I/O	Function
0	PS2 clock I/O pin (KCLK0)	PS2AC	I/O	PS2 clock inpu
	PS2 data I/O pin (KD0)	PS2AD	I/O	PS2 data inpu
1	PS2 clock I/O pin (KCLK1)	PS2BC	I/O	PS2 clock inpu
	PS2 data I/O pin (KD1)	PS2BD	I/O	PS2 data inpu
2	PS2 clock I/O pin (KCLK2)	PS2CC	I/O	PS2 clock inpu
	PS2 data I/O pin (KD2)	PS2CD	I/O	PS2 data inpu
3	PS2 clock I/O pin (KCLK3)	PS2DC	I/O	PS2 clock inpu
	PS2 data I/O pin (KD3)	PS2DD	I/O	PS2 data inpu

Note: * These are the external I/O pin names. In the text, clock I/O pins are referred t and data I/O pins as KD, omitting the channel designations.

	register_0					
	Keyboard control register H_0	KBCRH_0	R/W	H'70	H'FED8	8
	Keyboard control register L_0	KBCRL_0	R/W	H'70	H'FED9	8
	Keyboard data buffer register_0	KBBR_0	R	H'00	H'FEDA	8
Channel 1	Keyboard control register 1_1	KBCR1_1	R/W	H'00	H'FEC2	8
	Keyboard control register 2_1	KBCR2_1	R/W	H'F0	H'FEDF	8
	Keyboard buffer transmit data register_1	KBTR_1	R/W	H'FF	H'FEC3	8
	Keyboard control register H_1	KBCRH_1	R/W	H'70	H'FEDC	8
	Keyboard control register L_1	KBCRL_1	R/W	H'70	H'FEDD	8
	Keyboard data buffer register_1	KBBR_1	R	H'00	H'FEDE	8
Channel 2	Keyboard control register 1_2	KBCR1_2	R/W	H'00	H'FEC4	8
	Keyboard control register 2_2	KBCR2_2	R/W	H'F0	H'FEE3	8
	Keyboard buffer transmit data register_2	KBTR_2	R/W	H'FF	H'FEC5	8
	Keyboard control register H_2	KBCRH_2	R/W	H'70	H'FEE0	8
	Keyboard control register L_2	KBCRL_2	R/W	H'70	H'FEE1	8
	Keyboard data buffer register_2	KBBR_2	R	H'00	H'FEE2	8
Channel 3	Keyboard control register 1_3	KBCR1_3	R/W	H'00	H'FED2	8
	Keyboard control register 2_3	KBCR2_3	R/W	H'F0	H'FFE3	8
	Keyboard buffer transmit data register_3	KBTR_3	R/W	H'FF	H'FED3	8
	Keyboard control register H_3	KBCRH_3	R/W	H'70	H'FFE0	8
	Keyboard control register L_3	KBCRL_3	R/W	H'70	H'FFE1	8
	Keyboard data buffer register_3	KBBR_3	R	H'00	H'FFE2	8

				• When 0 is written • When the KBTE is set to 1 • When the KBIOE is cleared to 0 1: Starts data transmission [Setting condition] When 1 is written after reading the KBTS = 0
6	PS	0	R/W	Transmit Parity Selection Selects even or odd parity. 0: Selects odd parity 1: Selects even parity
5	KCIE	0	R/W	First KCLK Falling Interrupt Enable Selects whether an interrupt at the first fallin KCLK is enabled or disabled. 0: Disables first KCLK falling interrupt 1: Enables first KCLK falling interrupt
4	KTIE	0	R/W	Transmit Completion Interrupt Enable Selects whether a transmit completion interru enabled or disabled. 0: Disables transmit completion interrupt 1: Enables transmit completion interrupt
3	—	0	—	Reserved The initial value should not be changed.

				Note that this flag cannot be set when softwar standby mode or watch mode is cancelled. (H internal flag is set.)
1	KBTE	0	R/(W)*	Transmit Completion Flag Indicates that data transmission is completed. KTIE and KBTE are set to 1, requests the CP interrupt. 0: [Clearing condition] After reading KBTE = 1, 0 is written 1: [Setting Condition] When all KBTR data has been transmitted (Se eleventh rising edge of the KCLK signal)
0	KTER	0	R	Transmit Error Stores a notification of receive completion. Va when KBTE = 1. 0: 0 received as a notification of receive comp 1: 1 received as a notification of receive comp

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

3	TXCR3	0	R	Transmit Counter
2	TXCR2	0	R	Indicates bit of transmit data. Counter is incre at the falling edge of KCLK. The transmit cou initialized by a reset, when the KBTS is clear the KBIOE is cleared to 0, or the KBTE is se 0000: Clear 0001: KBT0 0010: KBT1 0011: KBT2 0100: KBT3 0101: KBT4 0110: KBT5 0111: KBT6 1000: KBT7 1001: Parity bit 1010: Stop bit 1011: Transmit completion notification
1	TXCR1	0	R	
0	TXCR0	0	R	

				1: The keyboard buffer control unit is enabled transmission and reception (KCLK and KD pins are in the bus drive state)
6	KCLKI	1	R	Keyboard Clock In Monitors the KCLK I/O pin. This bit cannot be modified. 0: KCLK I/O pin is low 1: KCLK I/O pin is high
5	KDI	1	R	Keyboard Data In Monitors the KDI I/O pin. This bit cannot be m 0: KD I/O pin is low 1: KD I/O pin is high
4	KBFSEL	1	R/W	Keyboard Buffer Register Full Select Selects whether the KBF bit is used as the ke buffer register full flag or as the KCLK fall inte flag. When KBF bit is used as the KCLK fall in flag, the KBE bit in KBCRL should be cleared disable reception. 0: KBF bit is used as KCLK fall interrupt flag 1: KBF bit is used as keyboard buffer register
3	KBIE	0	R/W	Keyboard Interrupt Enable Enables or disables interrupts from the keybo buffer control unit to the CPU. 0: Interrupt requests are disabled 1: Interrupt requests are enabled

				When data has been received normally been transferred to KBBR while KBFSEL (keyboard buffer register full flag) • When a KCLK falling edge is detected wl KBFSEL = 0 (KCLK interrupt flag)
1	PER	0	R/(W)*	Parity Error Indicates that an odd parity error has occurre 0: [Clearing condition] Read PER when PER =1, then write 0 in PE 1: [Setting condition] When an odd parity error occurs
0	KBS	0	R	Keyboard Stop Indicates the receive data stop bit. Valid only KBF = 1. 0: 0 stop bit received 1: 1 stop bit received

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

				1: Loading of receive data into KBBR is enabl
6	KCLKO	1	R/W	Keyboard Clock Out Controls PS2 clock I/O pin output. 0: PS2 clock I/O pin is low 1: PS2 clock I/O pin is high
5	KDO	1	R/W	Keyboard Data Out Controls PS2 data I/O pin output. 0: PS2 data I/O pin is low 1: PS2 data I/O pin is high When the start bit (KDO) is automatically clea (KDO = 1) by means of automatic transmissio written after reading 1.
4	—	1	—	Reserved This bit is always read as 1 and cannot be mo

0010: KB0

0011: KB1

0100: KB2

0101: KB3

0110: KB4

0111: KB5

1000: KB6

1001: KB7

1010: Parity bit

1011: —

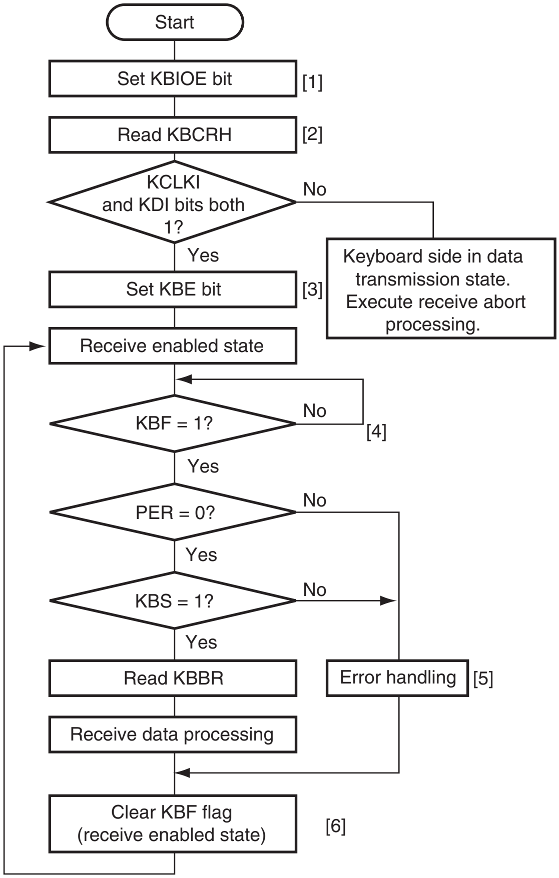
11- -: —

3	KB3	0	R
2	KB2	0	R
1	KB1	0	R
0	KB0	0	R

19.3.6 Keyboard Buffer Transmit Data Register (KBTR)

KBTR stores transmit data.

Bit	Bit Name	Initial Value	R/W	Description
7	KBT7	1	R/W	Keyboard Buffer Transmit Data Register 7 to 0
6	KBT6	1	R/W	Initialized to H'FF at reset.
5	KBT5	1	R/W	
4	KBT4	1	R/W	
3	KBT3	1	R/W	
2	KBT2	1	R/W	
1	KBT1	1	R/W	
0	KBT0	1	R/W	



[1] Set the KBIOE bit to 1 in KBCRL.

[2] Read KBCRH, and if the KCLKI and
are both 1, set the KBE bit (receive e
state).

[3] Detect the start bit output on the keyb
side and receive data in synchronizat
the fall of KCLK.

[4] When a stop bit is received, the keyb
buffer controller drives KCLK low to d
keyboard transmission (automatic I/C
If the KBIE bit is set to 1 in KBCRH, a
interrupt request is sent to the CPU a
same time.

[5] Perform receive data processing.

[6] Clear the KBF flag to 0 in KBCRL.
At the same time, the system automa
drives KCLK high, setting the receive
state.

The receive operation can be continued

Figure 19.3 Sample Receive Processing Flowchart

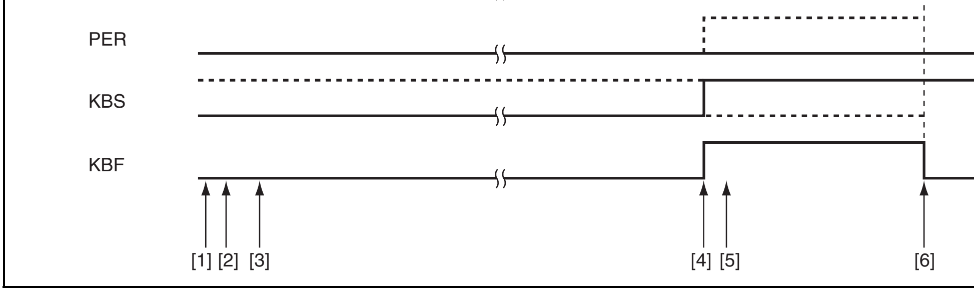
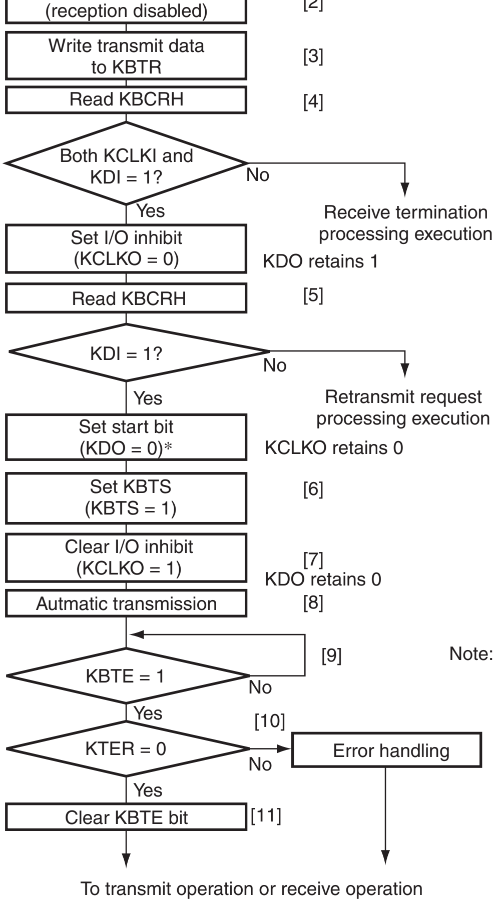


Figure 19.4 Receive Timing



[2] Clear the KBE bit (reception disabled).

[3] Write transmit data to KBTR.

[4] Read KBCRH, and when both the KCLKI and KDI bits are 1, write 0 to the KCLKO bit to set the I/O inhibit. 60 µs or more is required for I/O inhibit.

[5] Read KBCRH, and when the KDI bit is 1, write 0 to the KDO (set start bit).

[6] Write 1 to the KBTS bit to enter the transmit enabled state.

[7] Write 1 to the KCLKO bit to clear the I/O inhibi

[8] Check D0 to D7, the parity bit, the stop bit, an receive completion notification (send data at th falling edge of the KCLK signal).

[9] The KBTE bit is set to 1 at the eleventh rising edge of the KCLK signal. When KTIE = 1, a CPU interrupt occurs.

[10] When KTER = 0, transmission is successfully

[11] Clear the KBTE bit to 0.

Note: * The start bit (KDO = 0) is automatically initialized when automatic transmission is started. After initia to write 0 to KDO, read 1 before writing 0 to it.

Figure 19.5 Sample Transmit Processing Flowchart

Figure 19.6 Transmit Timing

19.4.3 Receive Abort

This LSI (system side) can forcibly abort transmission from the device connected to it (ke
side) in the event of a protocol error, etc. In this case, the system holds the clock low. Du
reception, the keyboard also outputs a clock for synchronization, and the clock is monito
the keyboard output clock is high. If the clock is low at this time, the keyboard judges tha
an abort request from the system, and data transmission from the keyboard is aborted. Th
system can abort reception by holding the clock low for a certain period. A sample receiv
processing flowchart is shown in figure 19.7, and the receive abort timing in figure 19.8.

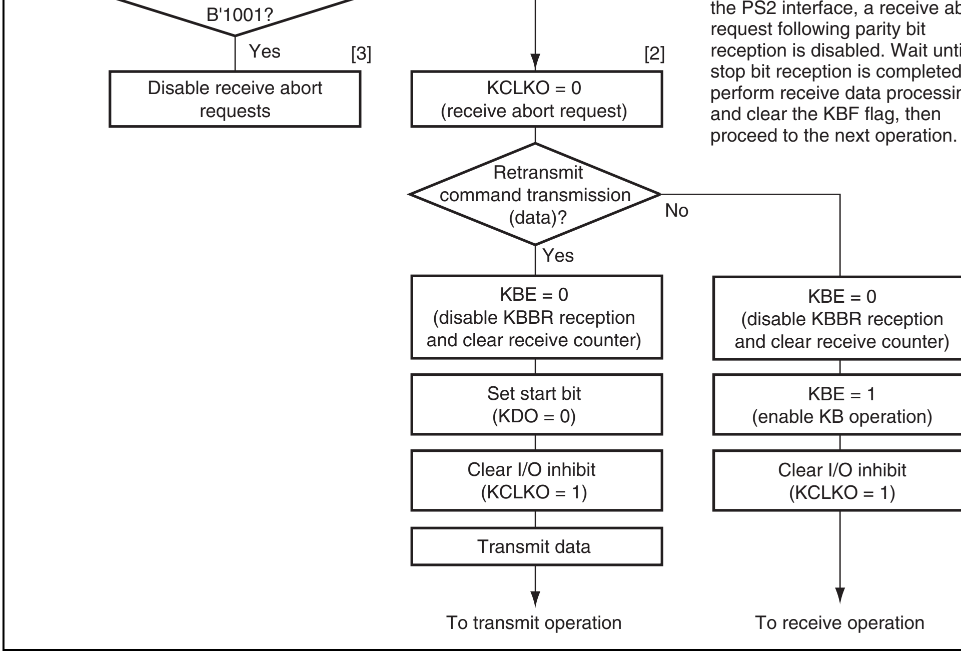

B'1001?
Yes
[3]
Disable receive abort requests
[2]
KCLKO = 0
(receive abort request)
Retransmit command transmission (data)?
No
Yes
KBE = 0
(disable KBBR reception and clear receive counter)
Set start bit
(KDO = 0)
Clear I/O inhibit
(KCLKO = 1)
Transmit data
To transmit operation
KBE = 0
(disable KBBR reception and clear receive counter)
KBE = 1
(enable KB operation)
Clear I/O inhibit
(KCLKO = 1)
To receive operation
the PS2 interface, a receive ab
request following parity bit
reception is disabled. Wait unti
stop bit reception is completed
perform receive data processin
and clear the KBF flag, then
proceed to the next operation.

Figure 19.7 Sample Receive Abort Processing Flowchart (1)

the data is transmitted.

Figure 19.7 Sample Receive Abort Processing Flowchart (2)

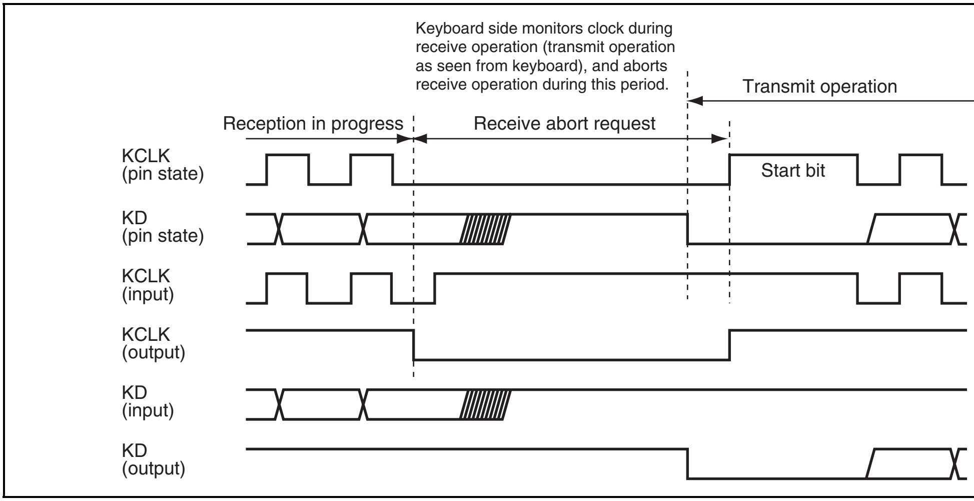
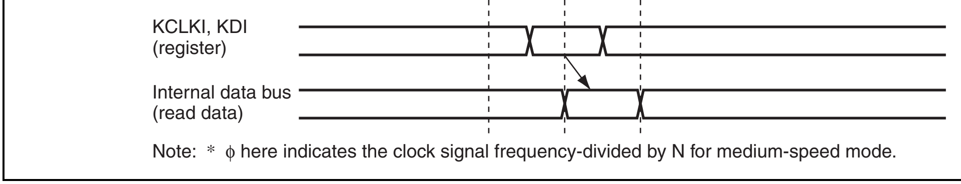

Keyboard side monitors clock during receive operation (transmit operation as seen from keyboard), and aborts receive operation during this period.
Transmit operation
Reception in progress
Receive abort request
KCLK (pin state)
Start bit
KD (pin state)
KCLK (input)
KCLK (output)
KD (input)
KD (output)

Figure 19.8 Receive Abort and Transmit Start (Transmission/Reception Switchover) Timing

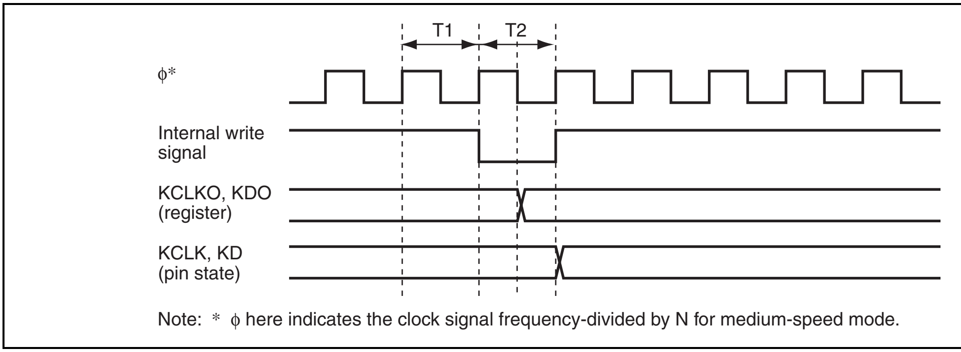


Note: * ϕ here indicates the clock signal frequency-divided by N for medium-speed mode.

Figure 19.9 KCLKI and KDI Read Timing

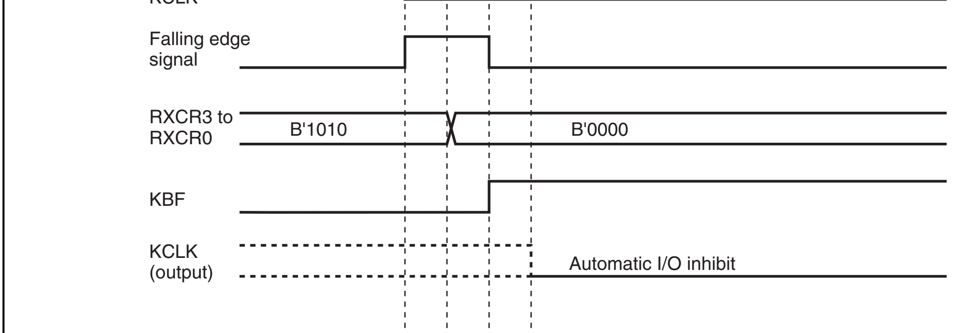
19.4.5 KCLKO and KDO Write Timing

Figure 19.10 shows the KLCKO and KDO write timing and the KCLK and KD pin state



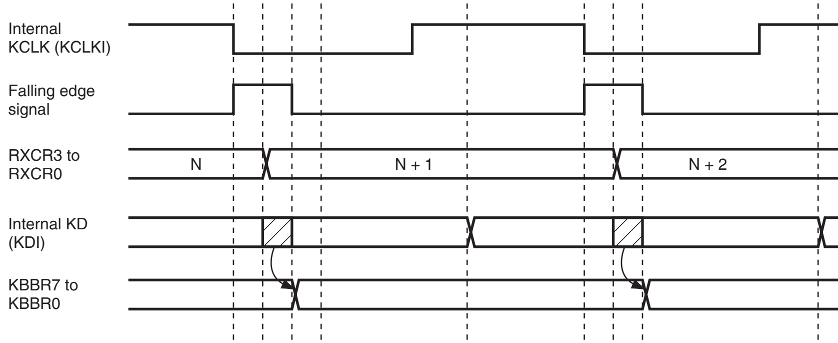
Note: * ϕ here indicates the clock signal frequency-divided by N for medium-speed mode.

Figure 19.10 KCLKO and KDO Write Timing



Note: * φ here indicates the clock signal frequency-divided by N for medium-speed mode.

Figure 19.11   KBF Setting and KCLK Automatic I/O Inhibit Generation Tim



Note: * ϕ here indicates the clock signal frequency-divided by N for medium-speed mode.

Figure 19.12 Receive Counter and KBBR Data Load Timing

19.4.8 Operation during Data Reception

If the KBS bit in KBCRH is set to 1 with other keyboard buffer control units in receptio
KCLK is automatically pulled down. Figure 19.13 shows receive timing and the KCLK.

Note: * Period from the first falling edge of KCLK to completion of reception (KBF

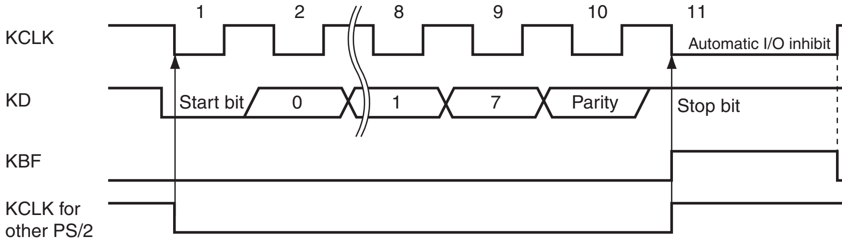
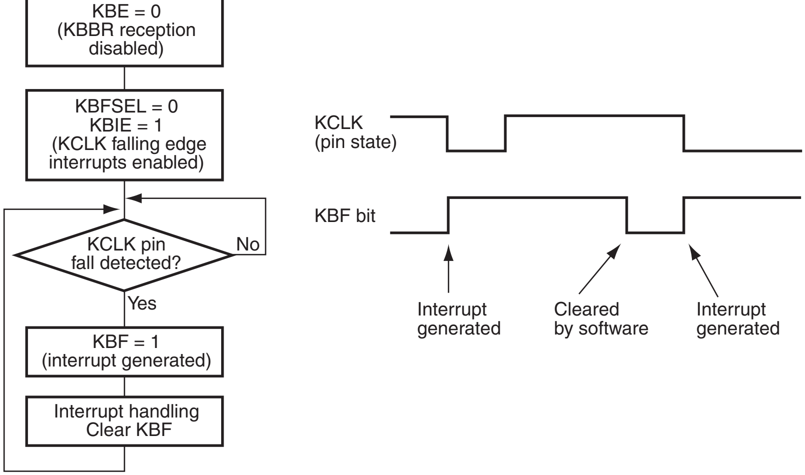


Figure 19.13 Receive Timing and KCLK



Note: * The KBF setting timing is the same as the timing of KBF setting and KCLK automatic I/O inhibit bit generation in figure 19.11. When the KBF bit is used as the KCLK input fall interrupt flag, the automatic I/O inhibit function does not operate.

Figure 19.14 Example of KCLK Input Fall Interrupt Operation

KCIF is set at the same time when the RXCR3 to RXCR0 bits in KBCRL are increm
from B'0000 to B'0001.

- Transmission

  When both KBIOE and KBTS are set to 1, the KCIF is set after the first falling edge has been detected.

  At this time, if KCIE is set to 1, the CPU is requested an interrupt.

  KCIF is set at the same time when the TXCR3 to TXCR0 bits in KBCR2 are increm B'0000 to B'0001.

- Determining interrupt generation

  By checking the KBE, KBTS, and KBTE bits, it can be determined whether the first falling interrupt is occurred during reception or transmission.

  During reception: KBE = 1

  During transmission: KBTS = 1 or KBTE = 1 (Check KBTE = 1 because the KBTS automatically cleared after transfer has been completed.)

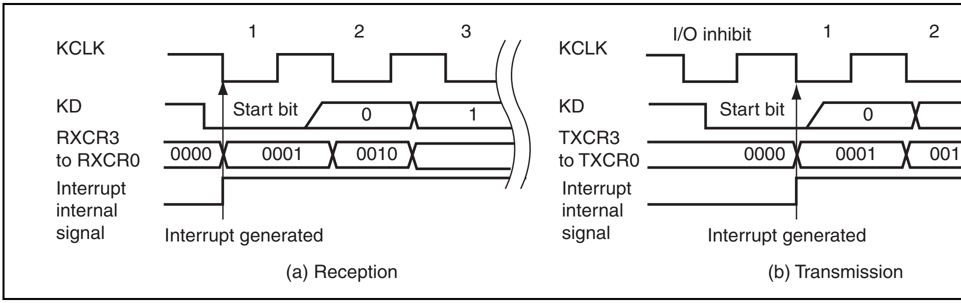


Figure 19.15 Timing of First KCLK Interrupt

In the first KCLK interrupt handling routine, the KCIF bit is checked. If the KCIF
indicates that the interrupt is generated after software standby mode and watch mo
been cancelled.

— When software standby mode or watch mode is cancelled by receiving a receive c
reception is ignored. Execute reception terminating processing by an interrupt han
routine, and then request retransfer.

— When transition to software standby mode or watch mode is made and the mode i
canceled by a first KCLK falling interrupt during data transmission, state before
performing mode transition is held immediately after canceling the mode. Therefo
initialization by an interrupt handling routine is required. Precautions as (b) and (
are shown in figure 19.17 should be applied on interrupt generation.

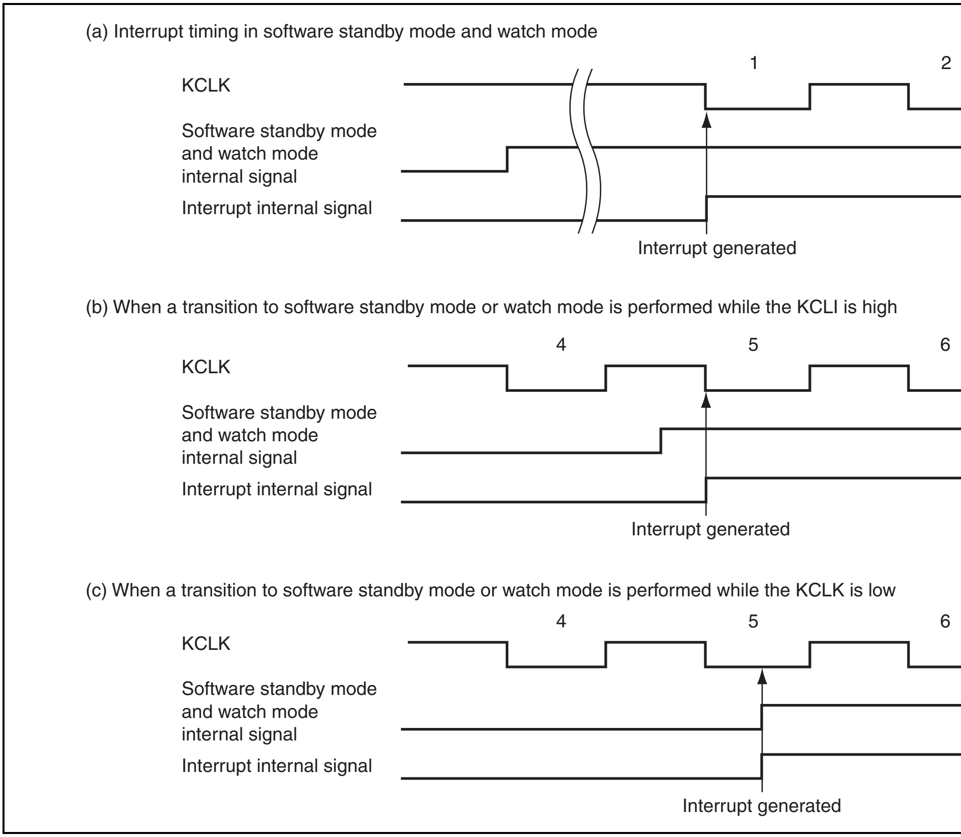
— Priority of canceling software standby mode and watch mode is decided by the se
ICR.

— The interrupt signal path and flag setting of the first KCLK interrupt in normal op
differ from those in software standby mode and watch mode. Figure 19.6 shows th
interrupt signal paths of the first KCLK interrupt.

Signal A: Interrupt signal in normal operation

Signal B: Interrupt signal in software standby mode and watch mode

— KCLK is input directly to the interrupt control block, not through the PS2, in soft
standby mode and watch mode, and then an interrupt is generated by detection of
edge. Therefore, the KCIF flag is not set. In this case, a flag that is in the interrupt
block is set. The internal flag is automatically cleared after an interrupt request is
the CPU. Figure 19.18 shows setting and clearing timing.



(a) Interrupt timing in software standby mode and watch mode

(b) When a transition to software standby mode or watch mode is performed while the KCLI is high

(c) When a transition to software standby mode or watch mode is performed while the KCLK is low

1
2
KCLK
Software standby mode and watch mode internal signal
Interrupt internal signal
Interrupt generated
4
5
6
KCLK
Software standby mode and watch mode internal signal
Interrupt internal signal
Interrupt generated
4
5
6
KCLK
Software standby mode and watch mode internal signal
Interrupt internal signal
Interrupt generated

Figure 19.17 Interrupt Timing in Software Standby Mode and Watch Mo

Watch Mode

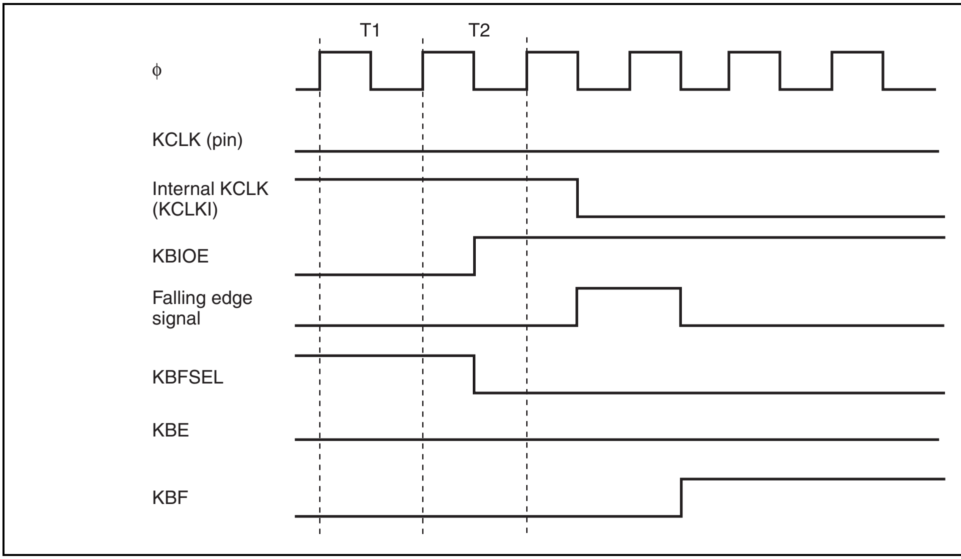


Figure 19.19 KBIOE Setting and KCLK Falling Edge Detection Timing

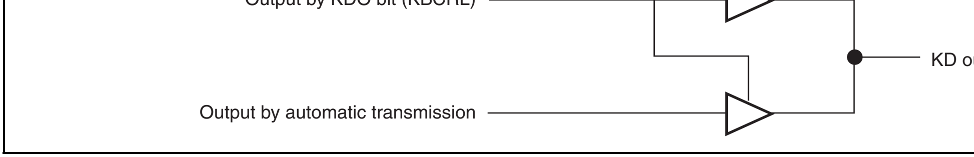


Figure 19.20 KDO Output

19.5.3 Module Stop Mode Setting

Keyboard buffer control unit operation can be enabled or disabled using the module stop
register. The initial setting is for keyboard buffer control unit operation to be halted. Regi
access is enabled by canceling module stop mode. For details, see section 26, Power-Dov
Modes.

19.5.4 Medium-Speed Mode

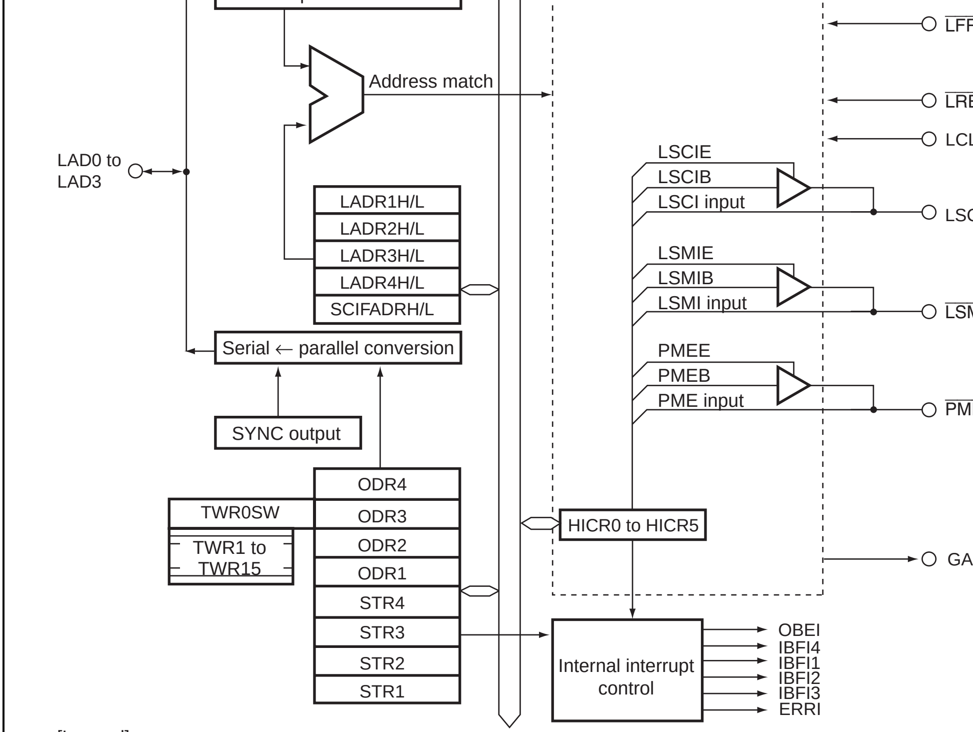
In medium-speed mode, the PS2 operates with the medium-speed clock. For normal oper
the PS2, set the medium-speed clock to a frequency of 300 kHz or higher.

19.5.5 Transmit Completion Flag (KBTE)

When TXCR3 to TXCR0 are 1011 (transmit completion notification) and then the TXCR
TXCR0 are initialized by clearing KBIOE or KBTS to 0, the transmit completion flag (K
set. In this case, KTER is invalid.

20.1 Features

- Supports LPC interface I/O read and I/O write cycles
  - Uses four signal lines (LAD3 to LAD0) to transfer the cycle type, address, and d
  - Uses three control signals: clock (LCLK), reset ($\overline{\text{LRESET}}$), and frame ($\overline{\text{LFRAM}}$
- Four register sets comprising data and status registers
  - The basic register set comprises three bytes: an input register (IDR), output regis and status register (STR).
  - I/O addresses from H'0000 to H'FFFF are selected for channels 1 to 4.
  - A fast Gate A20 function is provided for channel 1.
  - For channel 3, sixteen bidirectional data register bytes can be manipulated in add the basic register set.
- Supports SCIF
  - The LPC interface is connected to the SCIF, allowing direct control of the SCIF LPC host.
- Supports SERIRQ
  - Host interrupt requests are transferred serially on a single signal line (SERIRQ).
  - On channel 1, HIRQ1 and HIRQ12 can be generated.
  - On channels 2, 3 and 4, SMI, HIRQ6, and HIRQ9 to HIRQ11 can be generated.
  - In the SCIF, HIRQ1, SMI, and HIRQ3 to HIRQ15 can be generated.
  - Operation can be switched between quiet mode and continuous mode.
  - The $\overline{\text{CLKRUN}}$ signal can be manipulated to restart the PCI clock (LCLK).
- Power-down modes and interrupts
  - The LPC module can be shut down by inputting the $\overline{\text{LPCPD}}$ signal.
  - Three pins, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, and LSCI, are provided for general input/output.



[Legend]

HICR0 to HICR5:	Host interface control registers 0 to 5	STR1 to STR4:	Status registers 1 to 4
LADR1H/L to 4H/L:	LPC channel 1 to 4 address registers H and L	TWR0MW:	Bidirectional data register
		TWR0SW:	Bidirectional data register
SCIFADRH/L:	SCIF address register H and L	TWR1 to TWR15:	Bidirectional data registers
IDR1 to IDR4:	Input data registers 1 to 4	SIRQCR0 to SIRQCR4:	SERIRQ control registers
ODR1 to ODR4:	Output data registers 1 to 4	HISEL:	Host interface select regis

Figure 20.1 Block Diagram of LPC

				termination signal
LPC reset	$\overline{\text{LRESET}}$	P35	Input*1	LPC interface reset signal
LPC clock	LCLK	P36	Input	33-MHz PCI clock signal
Serialized interrupt request	SERIRQ	P37	I/O*1	Serialized host interrupt re signal in synchronization w
LSCI general output	LSCI	PB1	Output*1, *2	General output
LSMI general output	$\overline{\text{LSMI}}$	PB0	Output*1, *2	General output
PME general output	$\overline{\text{PME}}$	P80	Output*1, *2	General output
GATE A20	GA20	P81	Output*1, *2	Gate A20 control signal ou
LPC clock run	$\overline{\text{CLKRUN}}$	P82	I/O*1, *2	LCLK restart request sign serial host interrupt is requ
LPC power-down	$\overline{\text{LPCPD}}$	P83	Input*1	LPC module shutdown sig

Notes: 1. Pin state monitoring input is possible in addition to the LPC interface control input/output function.

2. Only 0 can be output. If 1 is output, the pin is in the high-impedance state, so external resistor is necessary to pull the signal up to VCC.

Host interface control register 2	HICR2	R/W	—	—	H'FE42	8
Host interface control register 3	HICR3	R	—	—	H'FE43	8
Host interface control register 4	HICR4	R/W	—	H'00	H'FDD9	8
Host interface control register 5	HICR5	R/W	—	H'00	H'FE33	8
LPC channel 1 address register H	LADR1H	R/W	—	H'00	H'FDC0	8
LPC channel 1 address register L	LADR1L	R/W	—	H'60	H'FDC1	8
LPC channel 2 address register H	LADR2H	R/W	—	H'00	H'FDC2	8
LPC channel 2 address register L	LADR2L	R/W	—	H'62	H'FDC3	8
LPC channel 3 address register H	LADR3H	R/W	—	H'00	H'FE34	8
LPC channel 3 address register L	LADR3L	R/W	—	H'00	H'FE35	8
LPC channel 4 address register H	LADR4H	R/W	—	H'00	H'FDD4	8
LPC channel 4 address register L	LADR4L	R/W	—	H'00	H'FDD5	8
Input data register 1	IDR1	R	W	H'00	H'FE38	8
Input data register 2	IDR2	R	W	H'00	H'FE3C	8
Input data register 3	IDR3	R	W	H'00	H'FE30	8
Input data register 4	IDR4	R	W	H'00	H'FDD6	8
Output data register 1	ODR1	R/W	R	H'00	H'FE39	8
Output data register 2	ODR2	R/W	R	H'00	H'FE3D	8
Output data register 3	ODR3	R/W	R	H'00	H'FE31	8
Output data register 4	ODR4	R/W	R	H'00	H'FDD7	8
Status register 1	STR1	R/W	R	H'00	H'FE3A	8
Status register 2	STR2	R/W	R	H'00	H'FE3E	8
Status register 3	STR3	R/W	R	H'00	H'FE32	8
Status register 4	STR4	R/W	R	H'00	H'FDD8	8

Bidirectional data register 6	TWR6	R/W	R/W	H'00	H'FE26
Bidirectional data register 7	TWR7	R/W	R/W	H'00	H'FE27
Bidirectional data register 8	TWR8	R/W	R/W	H'00	H'FE28
Bidirectional data register 9	TWR9	R/W	R/W	H'00	H'FE29
Bidirectional data register 10	TWR10	R/W	R/W	H'00	H'FE2A
Bidirectional data register 11	TWR11	R/W	R/W	H'00	H'FE2B
Bidirectional data register 12	TWR12	R/W	R/W	H'00	H'FE2C
Bidirectional data register 13	TWR13	R/W	R/W	H'00	H'FE2D
Bidirectional data register 14	TWR14	R/W	R/W	H'00	H'FE2E
Bidirectional data register 15	TWR15	R/W	R/W	H'00	H'FE2F
SERIRQ control register 0	SIRQCR0	R/W	—	H'00	H'FE36
SERIRQ control register 1	SIRQCR1	R/W	—	H'00	H'FE37
SERIRQ control register 2	SIRQCR2	R/W	—	H'00	H'FDDA
SERIRQ control register 3	SIRQCR3	R/W	—	H'00	H'FDDB
SERIRQ control register 4	SIRQCR4	R/W	—	H'00	H'FE3B
Host interface select register	HISEL	R/W	—	H'03	H'FE3F
SCIF address register H	SCIFADRH	R/W	—	H'03	H'FDC4
SCIF address register L	SCIFADRL	R/W	—	H'F8	H'FDC5

Notes: R/W in the register description means as follows:

1. R/W slave indicates access from the slave (this LSI).
2. R/W host indicates access from the host.

7	LPC3E	0	R/W		LPC Enables 3 to 1
6	LPC2E	0	R/W	—	Enable or disable the LPC interface function. V LPC interface is enabled (one of the three bits 1), processing for data transfer between the sl LSI) and the host is performed using pins LAD LAD0, $\overline{\text{LFRAME}}$, $\overline{\text{LRESET}}$, LCLK, SERIRQ, $\overline{\text{C}}$ and $\overline{\text{LPCPD}}$. • LPC3E 0: LPC channel 3 operation is disabled No address (LADR3) matches for IDR3, O STR3, or TWR0 to TWR15 1: LPC channel 3 operation is enabled • LPC2E 0: LPC channel 2 operation is disabled No address (LADR2) matches for IDR2, O STR2 1: LPC channel 2 operation is enabled • LPC1E 0: LPC channel 1 operation is disabled No address (LADR1) matches for IDR1, O STR1 1: LPC channel 1 operation is enabled
5	LPC1E	0	R/W	—	

					1: Fast Gate A20 function enabled GA20 pin output is open-drain (external p resistor (Vcc) required)
3	SDWNE	0	R/W	—	LPC Software Shutdown Enable Controls LPC interface shutdown. For details LPC shutdown function, and the scope of init by an LPC reset and an LPC shutdown, see s 20.4.4, LPC Interface Shutdown Function (LF 0: Normal state, LPC software shutdown set enabled [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software rese • LPC hardware shutdown release (rising e $\overline{\text{LPCPD}}$ signal) 1: LPC hardware shutdown state setting ena Hardware shutdown state when $\overline{\text{LPCPD}}$ s low level [Setting condition] Writing 1 after reading SDWNE = 0

					output goes to 0 level 1 1 : PME output enabled, $\overline{\text{PME}}$ output is high-impedance
1	LSMIE	0	R/W	—	LSMI output Enable Controls LSMI output in combination with the L bit in HICR1. $\overline{\text{LSMI}}$ pin output is open-drain, ar external pull-up resistor (Vcc) is needed. LSMIE LSMIB 0 X : LSMI output disabled, other of pin is enabled 1 0 : LSMI output enabled, LSMI output goes to 0 level 1 1 : LSMI output enabled, LSMI output is Hi-Z
0	LSCIE	0	R/W	—	LSCI output Enable Controls LSCI output in combination with the L in HICR1. LSCI pin output is open-drain, and a external pull-up resistor (Vcc) is needed. LSCIE LSCIB 0 X : LSCI output disabled, other of pin is enabled 1 0 : LSCI output enabled, LSCI output goes to 0 level 1 1 : LSCI output enabled, LSCI output is high-impedance

[Legend]

X: Don't care

					• Cycle type or address indeterminate during t cycle [Clearing conditions] • LPC hardware reset or LPC software reset • LPC hardware shutdown or LPC software sh • Forced termination (abort) of transfer cycle s processing • Normal termination of transfer cycle subject t processing 1: LPC interface is performing transfer cycle pr [Setting condition] Match of cycle type and address
6	CLKREQ	0	R	—	LCLK Request Indicates that the LPC interface's SERIRQ outpu requesting a restart of LCLK. 0: No LCLK restart request [Clearing conditions] • LPC hardware reset or LPC software reset • LPC hardware shutdown or LPC software sh • There are no further interrupts for transfer to in quiet mode in which SERIRQ is set to cont mode 1: LCLK restart request issued [Setting condition] In quiet mode, SERIRQ interrupt output become necessary while LCLK is stopped

					1: SERIRQ transfer processing in progress [Setting condition] Start of SERIRQ transfer frame
4	LRSTB	0	R/W	—	LPC Software Reset Bit Resets the LPC interface. For the scope of initializ an LPC reset, see section 20.4.4, LPC Interface S Function (LPCPD). 0: Normal state [Clearing conditions] • Writing 0 • LPC hardware reset 1: LPC software reset state [Setting condition] Writing 1 after reading LRSTB = 0

					• LPC hardware reset or LPC software reset • LPC hardware shutdown (falling edge of $\overline{\text{LPCPD}}$ signal when SDWNE • LPC hardware shutdown release (rising edge of $\overline{\text{LPCPD}}$ signal when SDWNE 1: LPC software shutdown state [Setting condition] Writing 1 after reading SDWNB = 0
2	PMEB	0	R/W	—	PME Output Bit Controls PME output in combination with the PM For details, refer to description on the PMEE bit
1	LSMIB	0	R/W	—	LSMI Output Bit Controls LSMI output in combination with the LS For details, refer to description on the LSMIE bit
0	LSCIB	0	R/W	—	LSCI output Bit Controls LSCI output in combination with the LS For details, refer to description on the LSCIE bit

Bit	Bit Name	Value	Slave	Host	Description
7	GA20	Undefined	R	—	GA20 Pin Monitor
6	LRST	0	R/(W)*	—	LPC Reset Interrupt Flag This bit is a flag that generates an ERRI inter when an LPC hardware reset occurs. 0: [Clearing condition] Writing 0 after reading LRST = 1 1: [Setting condition] $\overline{\text{LRESET}}$ pin falling edge detection
5	SDWN	0	R/(W)*	—	LPC Shutdown Interrupt Flag This bit is a flag that generates an ERRI inter when an LPC hardware shutdown request is generated. 0: [Clearing conditions] • Writing 0 after reading SDWN = 1 • LPC hardware reset ($\overline{\text{LRESET}}$ pin falling edge detection) • LPC software reset (LRSTB = 1) 1: [Setting condition] $\overline{\text{LPCPD}}$ pin falling edge detection

					• LPC software reset (LRSTB = 1) • LPC hardware shutdown (SDWNE = 1 and $\overline{\text{LPCPD}}$ pin falling edg detection) • LPC software shutdown (SDWNB = 1) 1: [Setting condition] $\overline{\text{LFRAME}}$ pin falling edge detection during L transfer cycle
3	IBFIE3	0	R/W	—	IDR3 and TWR Receive Complete interrupt Enables or disables IBFI3 interrupt to the sla LSI). 0: Input data register IDR3 and TWR recei complete interrupt requests disabled 1: [When TWRE = 0 in LADR3] Input data register (IDR3) receive co interrupt requests enabled [When TWRE = 1 in LADR3] Input data register (IDR3) and TWR complete interrupt requests enabled
2	IBFIE2	0	R/W	—	IDR2 Receive Complete interrupt Enable Enables or disables IBFI2 interrupt to the sla LSI). 0: Input data register (IDR2) receive comple interrupt requests disabled 1: Input data register (IDR2) receive comple interrupt requests enabled

Enables or disables ERRI interrupt to the sla
LSI).

0: Error interrupt requests disabled

1: Error interrupt requests enabled

Note: * Only 0 can be written to bits 6 to 4, to clear the flag.

- HICR3

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	LFRAME	Undefined	R	—	$\overline{\text{LFRAME}}$ Pin Monitor
6	CLKRUN	Undefined	R	—	$\overline{\text{CLKRUN}}$ Pin Monitor
5	SERIRQ	Undefined	R	—	SERIRQ Pin Monitor
4	LRESET	Undefined	R	—	$\overline{\text{LRESET}}$ Pin Monitor
3	LPCPD	Undefined	R	—	$\overline{\text{LPCPD}}$ Pin Monitor
2	PME	Undefined	R	—	$\overline{\text{PME}}$ Pin Monitor
1	LSMI	Undefined	R	—	$\overline{\text{LSMI}}$ Pin Monitor
0	LSCI	Undefined	R	—	LSCI Pin Monitor

					0: LPC channel 4 is disabled For IDR4, ODR4, and STR4, address (match is not occurred. 1: LPC channel 4 enabled
5	IBFIE4	0	R/W	—	IDR4 Receive Completion Interrupt Enable Enables or disables IBFI4 interrupt to the s LSI). 0: Input data register (IDR4) receive comp interrupt requests disabled 1: Input data register (IDR4) receive comp interrupt requests enabled
4 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

6	OBEI	0	R/W	—	Output Buffer Empty Interrupt Flag 0: [Clearing conditions] • Writing 0 after reading OBEI = 1 • LPC hardware reset or LPC software re 1: [Setting condition] When one of OBF1, OBF2, OBF3A, OBF3I OBF4 is cleared
5 to 4	—	All 0	R/W	—	Reserved The initial value bit should not be changed.
3	SCIFE	0	R/W	—	SCIF Enable Enables or disables access from the LPC h the SCIF. 0: Disables access from the LPC host of th 1: Enables access from the LPC host of the
2 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

6	Bit 14	0	R/W	—	Set the LPC channel 1 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR1L

			R/W		
Bit	Bit Name	Initial Value	Slave	Host	Description
7	Bit 7	0	R/W	—	Channel 1 Address Bits 7 to 3
6	Bit 6	1	R/W	—	Set the LPC channel 1 host address.
5	Bit 5	1	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit 2	0	R/W	—	Reserved This bit is ignored when an address match decided.
1	Bit 1	0	R/W	—	Channel 1 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 1 host address.

20.3.6 LPC Channel 2 Address Registers H and L (LADR2H and LADR2L)

LADR2 sets the LPC channel 2 host address. The LADR2 contents must not be changed channel 2 is operating (while LPC2E is set to 1).

- LADR2H

		Initial	R/W		
Bit	**Bit Name**	**Value**	**Slave**	**Host**	**Description**
7	Bit 15	0	R/W	—	Channel 2 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 2 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

					This bit is ignored when an address match i
1	Bit 1	1	R/W	—	Channel 2 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 2 host address.

- Host select register

I/O Address			Transfer Cycle	Host Select Re
Bits 5 to 3	**Bit 2**	**Bits 1 and 0**		
Bits 15 to 3 in LADR2	0	Bits 1 and 0 in LADR2	I/O write	IDR2 write (dat
Bits 15 to 3 in LADR2	1	Bits 1 and 0 in LADR2	I/O write	IDR2 write (cor
Bits 15 to 3 in LADR2	0	Bits 1 and 0 in LADR2	I/O read	ODR2 read
Bits 15 to 3 in LADR2	1	Bits 1 and 0 in LADR2	I/O read	STR2 read

Note: * When channel 2 is used, the content of LADR2 must be set so that the addre channels 1, 3, 4, and SCIF are different.

6	Bit 14	0	R/W	—	Set the LPC channel 3 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR3L

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 7	0	R/W	—	Channel 3 Address Bits 7 to 3
6	Bit 6	0	R/W	—	Set the LPC channel 3 host address.
5	Bit 5	0	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	—	0	R/W	—	Reserved The initial value should not be changed.
1	Bit 1	0	R/W	—	Channel 3 Address Bit 1 Sets the LPC channel 3 host address.
0	TWRE	0	R/W	—	Bidirectional Data Register Enable Enables or disables bidirectional data regist operation. 0: TWR operation is disabled TWR-related I/O address match determin halted 1: TWR operation is enabled

Bit 4	Bit 3	0	Bit 1	0	I/O read	ODR3 read
Bit 4	Bit 3	1	Bit 1	0	I/O read	STR3 read
$\overline{\text{Bit}}$ $\overline{4}$	0	0	0	0	I/O write	TWR0MW write
$\overline{\text{Bit}}$ $\overline{4}$	0 : 1	0 : 1	0 : 1	1 : 1	I/O write	TWR1 to TWR15 write
$\overline{\text{Bit}}$ $\overline{4}$	0	0	0	0	I/O read	TWR0SW read
$\overline{\text{Bit}}$ $\overline{4}$	0 : 1	0 : 1	0 : 1	1 : 1	I/O read	TWR1 to TWR15 read

Note: * When channel 3 is used, the content of LADR3 must be set so that the addre channels 1, 2, 4, and SCIF are different.

6	Bit 14	0	R/W	—	Set the LPC channel 4 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR4L

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 7	0	R/W	—	Channel 4 Address Bits 7 to 3
6	Bit 6	0	R/W	—	Set the LPC channel 4 host address.
5	Bit 5	0	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit2	0	R/W	—	Reserved This bit is ignored when an address match is decided.
1	Bit 1	0	R/W	—	Channel 4 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 4 host address.

20.3.9 Input Data Registers 1 to 4 (IDR1 to IDR4)

IDR1 to IDR4 are 8-bit read-only registers for the slave (this LSI), and 8-bit write-only r
for the host. The registers selected from the host according to the I/O address are shown
following table. Data transferred in an LPC I/O write cycle is written to the selected regi
value of bit 2 of the I/O address is latched into the C/$\overline{\text{D}}$ bit in STR, to indicate whether th
information is a command or data. The initial values of IDR1 to IDR4 are H'00.

I/O Address					Transfer	
Bits 15 to 4	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Se**
Bits 15 to 4	Bit 3	0	Bit 1	Bit 0	I/O write	IDRn write, C/$\overline{\text{D}}$n ←
Bits 15 to 4	Bit 3	1	Bit 1	Bit 0	I/O write	IDRn write, C/$\overline{\text{D}}$n ←

n = 1 to 4

20.3.10 Output Data Registers 1 to 4 (ODR1 to ODR4)

ODR1 to ODR4 are 8-bit readable/writable registers for the slave (this LSI), and 8-bit re
registers for the host. The registers selected from the host according to the I/O address a
in the following table. In an LPC I/O read cycle, the data in the selected register is transf
the host. The initial values of ODR1 to ODR4 are H'00.

I/O Address					Transfer	
Bits 15 to 4	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Se**
Bits 15 to 4	Bit 3	0	Bit1	Bit 0	I/O read	ODRn read

n = 1 to 4

Attempts by the slave to write to TWR0SW are invalid.

When the slave has access rights, TWR0SW is selected in TWR0 and the state of TWR0S returned when the slave reads TWR0MW. Attempts by the host to write to TWR0MW ar

For the registers selected from the host according to the I/O address, see section 20.3.7, L Channel 3 Address Registers H and L (LADR3H and LADR3L).

Data transferred in an LPC I/O write cycle is written to the selected register; in an LPC I/ cycle, the data in the selected register is transferred to the host. The initial values of TWR TWR15 are H'00.

20.3.12 Status Registers 1 to 4 (STR1 to STR4)

STR1 to STR4 are 8-bit registers that indicate status information during LPC interface pr The registers selected from the host according to the I/O address are shown in the followi In an LPC I/O read cycle, the data in the selected register is transferred to the host.

I/O Address					Transfer	
Bits 15 to 4	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Sele**
Bits 15 to 4	Bit 3	1	Bit1	Bit 0	I/O read	STRn read

n = 1 to 4

					address is written into this bit to indicate wl IDR1 contains data or a command. 0: Content of input data register (IDR1) is a 1: Content of input data register (IDR1) is a command
2	DBU12	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF1	0	R	R	Input Buffer Full This bit is an internal interrupt source to the (this LSI). The IBF1 flag setting and clearin conditions are different when the fast Gate used. For details, see table 20.5. 0: [Clearing condition] When the slave reads IDR1 1: [Setting condition] When the host writes to IDR1 in I/O write c
0	OBF1	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR1 in I/O read • When the slave writes 0 to the OBF1 bi 1: [Setting condition] When the slave writes to ODR1

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

					address is written into this bit to indicate wh IDR2 contains data or a command. 0: Content of input data register (IDR2) is a 1: Content of input data register (IDR2) is a command
2	DBU22	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF2	0	R	R	Input Buffer Full This bit is an internal interrupt source to the (this LSI). 0: [Clearing condition] When the slave reads IDR2 1: [Setting condition] When the host writes to IDR2 in I/O write cy
0	OBF2	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR2 in I/O read c • When the slave writes 0 to the OBF2 bit 1: [Setting condition] When the slave writes to ODR2

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

					When the host writes to TWR15 in I/O writ
6	OBF3B	0	R/(W)*	R	Bidirectional Data Register Output Buffer F 0: [Clearing conditions] • When the host reads TWR15 in I/O rea • When the slave writes 0 to the OBF3B 1: [Setting condition] When the slave writes to TWR15
5	MWMF	0	R	R	Master Write Mode Flag 0: [Clearing condition] When the slave reads TWR15 1: [Setting condition] When the host writes to TWR0 in I/O write while SWMF = 0
4	SWMF	0	R/(W)*	R	Slave Write Mode Flag In the event of simultaneous writes by the and the slave, the master write has priority 0: [Clearing conditions] • When the host reads TWR15 in I/O rea • When the slave writes 0 to the SWMF 1: [Setting condition] When the slave writes to TWR0 while MW

					The user can use this bit as necessary.
1	IBF3A	0	R	R	Input Buffer Full This bit is an internal interrupt source to the (this LSI). 0: [Clearing condition] When the slave reads IDR3 1: [Setting condition] When the host writes to IDR3 in I/O write cy
0	OBF3A	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR3 in I/O read • When the slave writes 0 to the OBF3 bi 1: [Setting condition] When the slave writes to ODR3

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

					address is written into this bit to indicate wl IDR3 contains data or a command. 0: Content of input data register (IDR3) is a 1: Content of input data register (IDR3) is a command
2	DBU32	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF3	0	R	R	Input Buffer Full This bit is an internal interrupt source to the (this LSI). 0: [Clearing condition] When the slave reads IDR3 1: [Setting condition] When the host writes to IDR3 in I/O write c
0	OBF3	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR3 in I/O read • When the slave writes 0 to the OBF3 bi 1: [Setting condition] When the slave writes to ODR3

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

					address is written into this bit to indicate wh IDR4 contains data or a command. 0: Content of input data register (IDR4) is a 1: Content of input data register (IDR4) is a command
2	DBU42	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF4	0	R	R	Input Buffer Full This bit is an internal interrupt source to the (this LSI). 0: [Clearing condition] When the slave reads IDR4 1: [Setting condition] When the host writes to IDR4 in I/O write cy
0	OBF4	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR4 in I/O read c • When the slave writes 0 to the OBF4 bit 1: [Setting condition] When the slave writes to ODR4

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

					0: Continuous mode [Clearing conditions] • LPC hardware reset, LPC software res • Specification by SERIRQ transfer cycle frame 1: Quiet mode [Setting condition] Specification by SERIRQ transfer cycle sto
6	SELREQ	0	R/W	—	Start Frame Initiation Request Select Selects the condition of a start frame initiat request when a host interrupt request is cle quiet mode. 0: Start frame initiation is requested when a interrupt requests are cleared 1: Start frame initiation is requested when o more interrupt requests are cleared
5	IEDIR2	0	R/W	—	Interrupt Enable Direct Mode 2 Selects whether an SERIRQ interrupt gene LPC channel 2 is affected only by a host in enable bit or by an OBF flag in addition to t bit. 0: A host interrupt is generated when both bit and the corresponding OBF flag are s 1: A host interrupt is generated when the e is set

					• Clearing OBF3B to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] Host SMI interrupt request by setting OB is enabled [When IEDIR3 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE3B = 0
3	SMIE3A	0	R/W	—	Host SMI Interrupt Enable 3A Enables or disables an SMI interrupt reques OBF3A is set by an ODR3 write. 0: Host SMI interrupt request by OBF3A ar SMIE3A is disabled [Clearing conditions] • Writing 0 to SMIE3A • LPC hardware reset, LPC software rese • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] Host SMI interrupt request by setting is e [When IEDIR3 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE3A = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] Host SMI interrupt request by setting O is enabled [When IEDIR2 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE2 = 0
1	IRQ12E1	0	R/W	—	Host IRQ12 Interrupt Enable 1 Enables or disables an HIRQ12 interrupt re when OBF1 is set by an ODR1 write. 0: HIRQ12 interrupt request by OBF1 and is disabled [Clearing conditions] • Writing 0 to IRQ12E1 • LPC hardware reset, LPC software res • Clearing OBF1 to 0 1: HIRQ12 interrupt request by setting OBI enabled [Setting condition] Writing 1 after reading IRQ12E1 = 0

- Clearing OBF1 to 0

1: HIRQ1 interrupt request by setting OBF1 enabled

[Setting condition]

Writing 1 after reading IRQ1E1 = 0

					IRQE11E3 is disabled [Clearing conditions] • Writing 0 to IRQ11E3 • LPC hardware reset, LPC software res • Clearing OBF3A to 0 (when IEDIR3 = 0 1: [When IEDIR3 = 0] HIRQ11 interrupt request by setting OB is enabled [When IEDIR3 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E3 = 0
6	IRQ10E3	0	R/W	—	Host IRQ10 Interrupt Enable 3 Enables or disables an HIRQ10 interrupt re when OBF3A is set by an ODR3 write. 0: HIRQ10 interrupt request by OBF3A ar IRQE10E3 is disabled [Clearing conditions] • Writing 0 to IRQ10E3 • LPC hardware reset, LPC software res • Clearing OBF3A to 0 (when IEDIR3 = 0 1: [When IEDIR3 = 0] HIRQ10 interrupt request by setting OB is enabled [When IEDIR3 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E3 = 0

					• Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ9 interrupt request by setting OBF3 enabled [When IEDIR3 = 1] HIRQ9 interrupt is requested [Setting condition] Writing 1 after reading IRQ9E3 = 0
4	IRQ6E3	0	R/W	—	Host IRQ6 Interrupt Enable 3 Enables or disables an HIRQ6 interrupt requ when OBF3A is set by an ODR3 write. 0: HIRQ6 interrupt request by OBF3A and is disabled [Clearing conditions] • Writing 0 to IRQ6E3 • LPC hardware reset, LPC software rese • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ6 interrupt request by setting OBF3 enabled [When IEDIR3 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E3 = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ11 interrupt request by setting OB enabled [When IEDIR2 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E2 = 0
2	IRQ10E2	0	R/W	—	Host IRQ10 Interrupt Enable 2 Enables or disables an HIRQ10 interrupt re when OBF2 is set by an ODR2 write. 0: HIRQ10 interrupt request by OBF2 and IRQE10E2 is disabled [Clearing conditions] • Writing 0 to IRQ10E2 • LPC hardware reset, LPC software rese • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ10 interrupt request by setting OB enabled [When IEDIR2 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E2 = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ9 interrupt request by setting OBF2 enabled [When IEDIR2 = 1] HIRQ9 interrupt is requested [Setting condition] Writing 1 after reading IRQ9E2 = 0
0	IRQ6E2	0	R/W	—	Host IRQ6 Interrupt Enable 2 Enables or disables an HIRQ6 interrupt requ when OBF2 is set by an ODR2 write. 0: HIRQ6 interrupt request by OBF2 and IF is disabled [Clearing conditions] • Writing 0 to IRQ6E2 • LPC hardware reset, LPC software rese • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ6 interrupt request by setting OBF2 enabled [When IEDIR2 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E2 = 0

					enable bit or by an OBF flag in addition to t bit. 0: A host interrupt is generated when both enable bit and the corresponding OBF f set 1: A host interrupt is generated when the is set
6	IEDIR4	0	R/W	—	Interrupt Enable Direct Mode 4 Selects whether an SERIRQ interrupt gene LPC channel 4 is affected only by a host in enable bit or by an OBF flag in addition to t bit. 0: A host interrupt is generated when both enable bit and the corresponding OBF f set 1: A host interrupt is generated when the is set

					• Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ11 interrupt request by setting OBF enabled [When IEDIR4 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E4 = 0
4	IRQ10E4	0	R/W	—	Host IRQ10 Interrupt Enable 4 Enables or disables an HIRQ10 interrupt req when OBF4 is set by an ODR4 write. 0: HIRQ10 interrupt request by OBF4 and IRQE10E4 is disabled [Clearing conditions] • Writing 0 to IRQ10E4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ10 interrupt request by setting OBF enabled [When IEDIR4 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E4 = 0

					• Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ9 interrupt request by setting OBF enabled [When IEDIR4 = 1] HIRQ9 interrupt is requested [Setting condition] Writing 1 after reading IRQ9E4 = 0
2	IRQ6E4	0	R/W	—	Host IRQ6 Interrupt Enable 4 Enables or disables an HIRQ6 interrupt rec when OBF4 is set by an ODR4 write. 0: HIRQ6 interrupt request by OBF4 and I is disabled [Clearing conditions] • Writing 0 to IRQ6E4 • LPC hardware reset, LPC software rese • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ6 interrupt request by setting OBF enabled [When IEDIR4 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E4 = 0

					• Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] Host SMI interrupt request by setting OB is enabled [When IEDIR4 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE4 = 0
0	—	0	R/W	—	Reserved The initial value should not be changed.

20.3.16 SERIRQ Control Register 3 (SIRQCR3)

SIRQCR3 contains bits that select the host interrupt request outputs.

		Initial	R/W		
Bit	**Bit Name**	**Value**	**Slave**	**Host**	**Description**
7	SELIRQ15	0	R/W	—	Host IRQ Interrupt Select These bits select the state of the output on t SERIRQ pins. 0: SERIRQ pin output is in the Hi-Z state 1: SERIRQ pin output is low
6	SELIRQ14	0	R/W	—	
5	SELIRQ13	0	R/W	—	
4	SELIRQ8	0	R/W	—	
3	SELIRQ7	0	R/W	—	
2	SELIRQ5	0	R/W	—	
1	SELIRQ4	0	R/W	—	
0	SELIRQ3	0	R/W	—	

1	SCSIRQ1	0	R/W	—	0000: No host interrupt request
0	SCSIRQ0	0	R/W	—	0001: HIRQ1
					0010: SMI
					0011: HIRQ3
					0100: HIRQ4
					0101: HIRQ5
					0110: HIRQ6
					0111: HIRQ7
					1000: HIRQ8
					1001: HIRQ9
					1010: HIRQ10
					1011: HIRQ11
					1100: HIRQ12
					1101: HIRQ13
					1110: HIRQ14
					1111: HIRQ15

6	—	0	R/W	—	These bits set the host addresses of the SC
5	—	0	R/W	—	
4	—	0	R/W	—	
3	—	0	R/W	—	
2	—	0	R/W	—	
1	—	1	R/W	—	
0	—	1	R/W	—	

- SCIFADRL

		Initial	R/W		
Bit	**Bit Name**	**Value**	**Slave**	**Host**	**Description**
7	—	1	R/W	—	SCIF Addresses 7 to 0
6	—	1	R/W	—	These bits set the host addresses of the SC
5	—	1	R/W	—	
4	—	1	R/W	—	
3	—	1	R/W	—	
2	—	0	R/W	—	
1	—	0	R/W	—	
0	—	0	R/W	—	

Note: When the SCIF is in use, set different addresses in the SCIFADR for channels 1, 2 and 4.

					details of STR3, see section 20.3.12, Statu Registers 1 to 4 (STR1 to STR4). 0: Bits 7 to 4 in STR3 indicate processing s the LPC interface. 1: [When TWRE = 1] Bits 7 to 4 in STR3 indicate processing s the LPC interface. [When TWRE = 0] Bits 7 to 4 in STR3 are readable/writabl which user can use as necessary
6	SELIRQ11	0	R/W	—	Host IRQ Interrupt Select These bits select the state of the output on SERIRQ pins. 0: [When host interrupt request is cleared] SERIRQ pin output is in the Hi-Z state [When host interrupt request is set] SERIRQ pin output is low 1: [When host interrupt request is cleared] SERIRQ pin output is low [When host interrupt request is set] SERIRQ pin output is in the Hi-Z state.
5	SELIRQ10	0	R/W	—	
4	SELIRQ9	0	R/W	—	
3	SELIRQ6	0	R/W	—	
2	SELSMI	0	R/W	—	
1	SELIRQ12	1	R/W	—	
0	SELIRQ1	1	R/W	—	

Use the following procedure to activate the LPC interface after a reset release.

1. Read the signal line status and confirm that the LPC module can be connected. Also c the LPC module is initialized internally.
2. When using channels 1, 2 and 4, set LADR1, LADR2, and LADR4 to determine the I address.
3. When using channel 3, set LADR3 to determine the I/O address and whether bidirecti registers are to be used.
4. Set the enable bit (LPC4E to LPC1E) for the channel to be used.
5. Set the enable bits (FGA20E, PMEE, LSMIE, and LSCIE) for the additional function used.
6. Set the selection bits for other functions (SDWNE, IEDIR).
7. As a precaution, clear the interrupt flags (LRST, SDWN, ABRT, OBF, and OBEI). Re or TWR15 to clear IBF.
8. Set receive complete interrupt enable bits (IBFIE4 to IBFIE1, ERRIE, and OBEI) as n

In an I/O read cycle or I/O write cycle, transfer is carried out using LAD3 to LAD0 in th following order, in synchronization with LCLK. The host can be made to wait by sendin value other than B'0000 in the slave's synchronization return cycle, but with the LPC of value of B'0000 always returns.

If the received address matches the host address in an LPC register (IDR, ODR, STR, ar the LPC interface enters the busy state; it returns to the idle state by output of a state cou turnaround. Register and flag changes are made at this timing, so in the event of a transf forced termination (abort), registers and flags are not changed.

The timing of the $\overline{\text{LFRAME}}$, LCLK, and LAD signals is shown in figures 20.2 and 20.3

6	Address 4	Host	Bits 3 to 0	Address 4	Host	Bit
7	Turnaround (recovery)	Host	1111	Data 1	Host	Bit
8	Turnaround	None	ZZZZ	Data 2	Host	Bit
9	Synchronization	Slave	0000	Turnaround (recovery)	Host	11
10	Data 1	Slave	Bits 3 to 0	Turnaround	None	ZZ
11	Data 2	Slave	Bits 7 to 4	Synchronization	Slave	00
12	Turnaround (recovery)	Slave	1111	Turnaround (recovery)	Slave	11
13	Turnaround	None	ZZZZ	Turnaround	None	ZZ

Figure 20.2   Typical $\overline{\text{LFRAME}}$ Timing

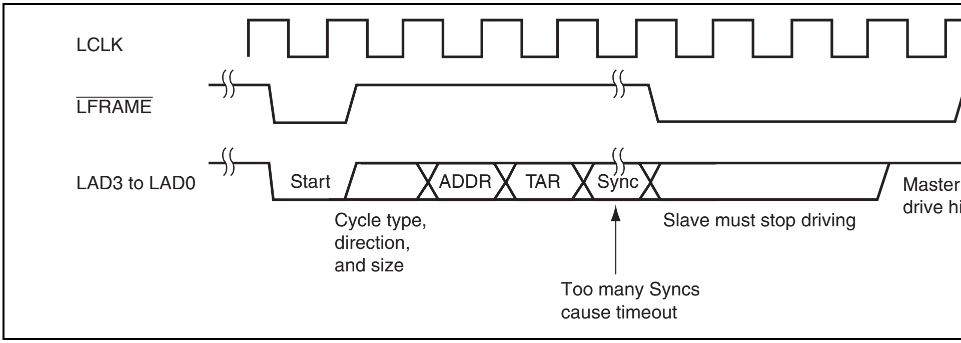


Figure 20.3   Abort Mechanism

Output of the Gate A20 signal can be controlled by an H'D1 command and data. When th (this LSI) receives data, it normally reads IDR1 in the interrupt handling routine activated IBFI1 interrupt. At this time, firmware copies bit 1 of data following an H'D1 command a outputs it on pin GA20.

(2) Fast Gate A20 Operation

The internal state of pin GA20 is initialized to 1 since the initial value of the FGA20E bit When the FGA20E bit is set to 1, pin P81/GA20 functions as the output of the fast GA20 The state of pin GA20 can be monitored by reading bit GA20 in HICR2.

The initial output from this pin is 1, which is the initial value. Afterward, the host can ma the output from this pin by sending commands and data. This function is only available v IDR1. The LPC decodes commands input from the host. When an H'D1 host command is detected, bit 1 of the data following the host command is output from pin GA20. This ope does not depend on firmware or interrupts, and is faster than the regular processing using interrupts. Table 20.4 shows the conditions that set and clear pin GA20. Figure 20.4 show GA20 output flow. Table 20.5 indicates the GA20 output signal values.

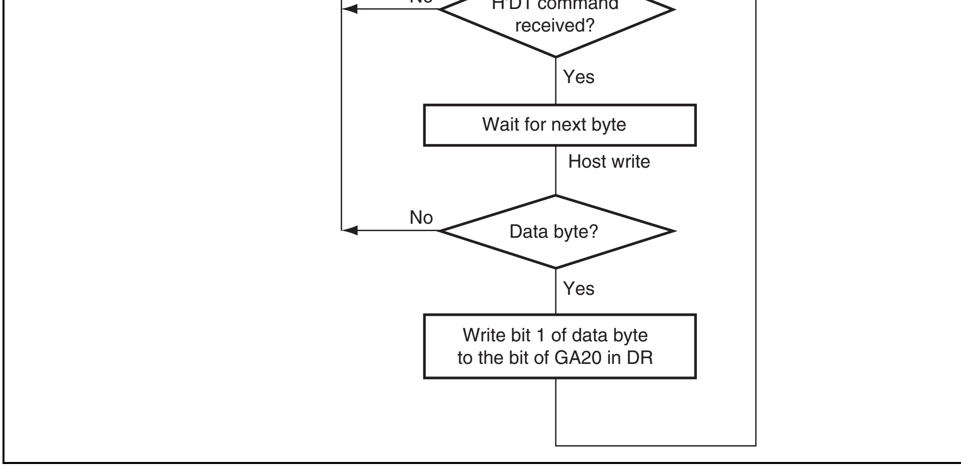


Figure 20.4 GA20 Output

1	H'FF command	0	Q (0)	
1	H'D1 command	0	Q	Turn-on sequen (abbreviated for
0	1 data*[1]	0	1	
1/0	Command other than H'FF and H'D1	1	Q (1)	
1	H'D1 command	0	Q	Turn-off sequen (abbreviated for
0	0 data*[2]	0	0	
1/0	Command other than H'FF and H'D1	1	Q (0)	
1	H'D1 command	0	Q	Cancelled sequ
1	Command other than H'D1	1	Q	
1	H'D1 command	0	Q	Retriggered sec
1	H'D1 command	0	Q	
1	H'D1 command	0	Q	Consecutively e sequences
0	Any data	0	1/0	
1	H'D1 command	0	Q (1/0)	

Notes: 1. Any data with bit 1 set to 1.

2. Any data with bit 1 cleared to 0.

for exiting software standby mode before clearing the shutdown state with the $\overline{\text{LPCPD}}$ s

If the SDWNE bit has been set to 1 beforehand, the LPC hardware shutdown state is ent same time as the $\overline{\text{LPCPD}}$ signal falls, and prior preparation is not possible. If the LPC so shutdown state is set by means of the SDWNB bit, on the other hand, the LPC software state cannot be cleared at the same time as the rising edge of the $\overline{\text{LPCPD}}$ signal. Taking points into consideration, the following operating procedure uses a combination of LPC shutdown and LPC hardware shutdown.

1. Clear the SDWNE bit to 0.
2. Set the ERRIE bit to 1 and wait for an interrupt by the SDWN flag.
3. When an ERRI interrupt is generated by the SDWN flag, check the LPC interface in status flags and perform any necessary processing.
4. Set the SDWNB bit to 1 to set LPC software standby mode.
5. Set the SDWNE bit to 1 and make a transition to LPC hardware standby mode. The bit is cleared automatically.
6. Check the state of the $\overline{\text{LPCPD}}$ signal to make sure that the $\overline{\text{LPCPD}}$ signal has not rise steps 3 to 5. If the signal has risen, clear SDWNE to 0 to return to the state in step 1.
7. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
8. When a rising edge is detected in the $\overline{\text{LPCPD}}$ signal, the SDWNE bit is automaticall to 0. If the slave has been placed in sleep mode, the mode is exited by means of $\overline{\text{LRE}}$ signal input, on completion of the LPC transfer cycle, or by some other means.

LSCI	PB1	Δ	I/O	Hi-Z, only when LSCIE = 1
$\overline{\text{LSMI}}$	PB0	Δ	I/O	Hi-Z, only when LSMIE = 1
$\overline{\text{PME}}$	P80	Δ	I/O	Hi-Z, only when PMEE = 1
GA20	P81	Δ	I/O	Hi-Z, only when FGA20E = 1
$\overline{\text{CLKRUN}}$	P82	O	Input	Hi-Z
$\overline{\text{LPCPD}}$	P83	X	Input	Needed to clear shutdown state

[Legend]

O: Pin that is shutdown by the shutdown function

Δ: Pin that is shutdown only when the LPC function is selected by register setting

X: Pin that is not shutdown

In the LPC shutdown state, the LPC's internal state and some register bits are initialized. of priority of LPC shutdown and reset states is as follows.

1. System reset (reset by $\overline{\text{RES}}$ pin input, or WDT overflow)
   All register bits, including bits LPC4E to LPC1E, are initialized.
2. LPC hardware reset (reset by $\overline{\text{LRESET}}$ pin input)
   LRSTB, SDWNE, and SDWNB bits are cleared to 0.
3. LPC software reset (reset by LRSTB)
   SDWNE and SDWNB bits are cleared to 0.
4. LPC hardware shutdown
   SDWNB bit is cleared to 0.
5. LPC software shutdown

The scope of the initialization in each mode is shown in table 20.7.

state)			
Host interrupt enable bits (IRQ1E1, IRQ12E1, SMIE2, IRQ6E2, IRQ9E2 to IRQ11E2, SMIE3B, SMIE3A, IRQ6E3, IRQ9E3 to IRQ11E3, SELREQ, SMIE4, IRQ6E4, IRQ9E4 to IRQ11E4, IEDIR2 to IEDIR4), Q/$\overline{\text{C}}$ flag	Initialized	Initialized	F
LRST flag	Initialized (0)	Can be set/cleared	C s
SDWN flag	Initialized (0)	Initialized (0)	C s
LRSTB bit	Initialized (0)	HR: 0 SR: 1	0 s
SDWNB bit	Initialized (0)	Initialized (0)	H S
SDWNE bit	Initialized (0)	Initialized (0)	H S
LPC interface operation control bits (LPC4E to LPC1E, FGA20E, LADR1 to LADR4, IBFIE1 to IBFIE4, PMEE, PMEB, LSMIE, LSMIB, LSCIE, LSCIB, TWRE, SELSTR3, SELIRQ1, SELSMI, SELIRQ3 to SELIRQ15, OBEIE, SCIFE, IDR1 to IDR4, ODR1 to ODR4, TWR0 to TWR15, SCSIRQ0 to SCSIRQ3, and SCIFADRH/L)	Initialized	Retained	F
$\overline{\text{LRESET}}$ signal	Input (port function	Input	I
$\overline{\text{LPCPD}}$ signal		Input	I
LAD3 to LAD0, $\overline{\text{LFRAME}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$ signals		Input	H
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is selected)		Output	H
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is not selected)		Port function	F

Note: System reset: Reset by $\overline{\text{RES}}$ pin input, or WDT overflow

LPC reset: Reset by LPC hardware reset (HR) or LPC software reset (SR)

LPC shutdown: Reset by LPC hardware shutdown (HS) or LPC software shutdow

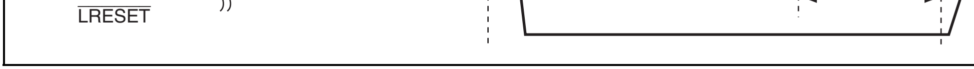


Figure 20.5 Power-Down State Termination Timing

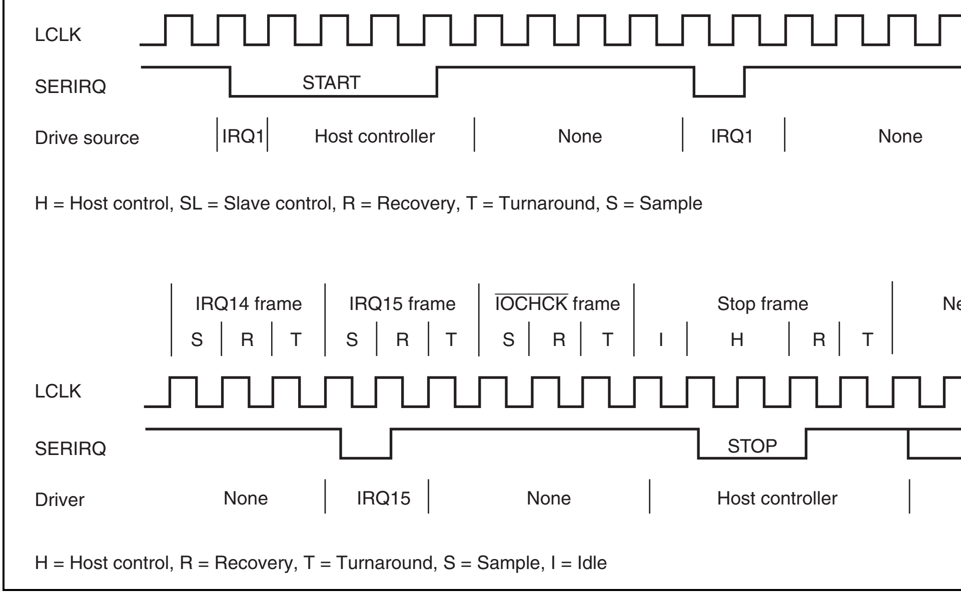


Figure 20.6 SERIRQ Timing

The serialized interrupt transfer cycle frame configuration is as follows. Two of the state comprising each frame are the recover state in which the SERIRQ signal is returned to t at the end of the frame, and the turnaround state in which the SERIRQ signal is not driv recover state must be driven by the host or slave that was driving the preceding state.

4	IRQ3	Slave	3	Drive possible in SCIF
5	IRQ4	Slave	3	Drive possible in SCIF
6	IRQ5	Slave	3	Drive possible in SCIF
7	IRQ6	Slave	3	Drive possible in LPC channels 2, 3, 4 SCIF
8	IRQ7	Slave	3	Drive possible in SCIF
9	IRQ8	Slave	3	Drive possible in SCIF
10	IRQ9	Slave	3	Drive possible in LPC channels 2, 3, 4 SCIF
11	IRQ10	Slave	3	Drive possible in LPC channels 2, 3, 4 SCIF
12	IRQ11	Slave	3	Drive possible in LPC channels 2, 3, 4 SCIF
13	IRQ12	Slave	3	Drive possible in LPC channel 1 and S
14	IRQ13	Slave	3	Drive possible in SCIF
15	IRQ14	Slave	3	Drive possible in SCIF
16	IRQ15	Slave	3	Drive possible in SCIF
17	IOCHCK	Slave	3	
18	Stop	Host	Undefined	First, 1 or more idle states, then 2 or 3 0-driven by host 2 states: Quiet mode next 3 states: Continuous mode next

20.4.6 LPC Interface Clock Start Request

A request to restart the clock (LCLK) can be sent to the host by means of the $\overline{\text{CLKRUN}}$ LPC data transfer and SERIRQ in continuous mode, a clock restart is never requested si transfer cycles are initiated by the host. With SERIRQ in quiet mode, when a host interr request is generated the $\overline{\text{CLKRUN}}$ signal is driven and a clock (LCLK) restart request is the host. The timing for this operation is shown in figure 20.7.

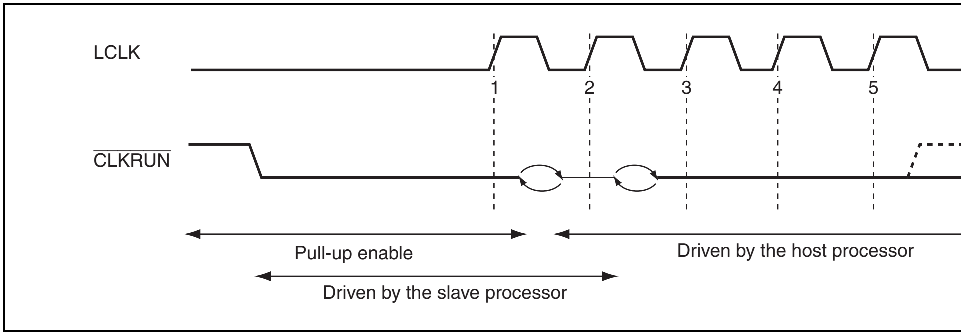

LCLK
1
2
3
4
5
$\overline{\text{CLKRUN}}$
Pull-up enable
Driven by the host processor
Driven by the slave processor

Figure 20.7 Clock Start Request Timing

Cases other than SERIRQ in quiet mode when clock restart is required must be handled different protocol, using the $\overline{\text{PME}}$ signal, etc.

20.4.7 SCIF Control from LPC Interface

Setting the SCIFE bit in HICR5 to 1 allows the LPC host to communicate with the SCIF the LPC interface can access the registers of the module SCIF other than SCIFCR. For d transmission and reception, see section 17, Serial Communication Interface with FIFO (

Table 20.9 Receive Complete Interrupts and Error Interrupt

Interrupt	Description
IBFI1	When IBFIE1 is set to 1 and IDR1 reception is completed
IBFI2	When IBFIE2 is set to 1 and IDR2 reception is completed
IBFI3	When IBFIE3 is set to 1 and IDR3 reception is completed, or when TWR IBFIE3 are set to 1 and reception is completed up to TWR15
IBFI4	When IBFIE4 is set to 1 and IDR4 reception is completed
OBEI	When OBEIE is set to 1 with OBEI set to 1.
ERRI	When ERRIE is set to 1 and one of LRST, SDWN and ABRT is set to 1

linked to the host interrupt request enable bits. When the OBF flag is cleared to 0 by a re
ODR or TWR15 by the host in the corresponding LPC channel, the corresponding host i
enable bit is automatically cleared to 0, and the host interrupt request is cleared.

When the IEDIR bit is set to 1 in SIRQCR, a host interrupt is requested by the only upon
interrupt enable bits. The host interrupt enable bit is not cleared when OBF is cleared. T
SMIE1, SMIE2, SMIE3A and SMIE3B, SMIE4, IRQ6En, IRQ9En, IRQ10En, and IRQ
their respective functional differences. In order to clear a host interrupt request, it is nec
clear the host interrupt enable bit. (n = 2 to 4.)

When the SCIF channels are used, clearing the DDCD bit in FMSR of the SCIF clears a
interrupt request.

Table 20.10 summarizes the methods of setting and clearing these bits when the LPC ch
used, and table 20.11 summarizes the methods of setting and clearing these bits when th
channels are used. Figure 20.8 shows the processing flowchart.

	SMIE3A and writes 1 • writes to TWR15, then reads 0 from bit SMIE3B and writes 1 • writes to ODR4, then reads 0 from bit SMIE4 and writes 1	reads ODR3 • writes 0 to bit SMIE3B, reads TWR15 • writes 0 to bit SMIE4, o reads ODR4

SMI
(IEDIR2 = 1,
IEDIR3 = 1, or
IEDIR4 = 1)
	Internal CPU
	• reads 0 from bit SMIE2, then writes 1
	• reads 0 from bit SMIE3A, then writes 1
	• reads 0 from bit SMIE3B, then writes 1
	• reads 0 from bit SMIE4, then writes 1
		Internal CPU
		• writes 0 to bit SMIE2
		• writes 0 to bit SMIE3A
		• writes 0 to bit SMIE3B
		• writes 0 to bit SMIE4

HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 0, IEDIR3 = 0, or IEDIR4 = 0)	Internal CPU • writes to ODR2, then reads 0 from bit IRQiE2 and writes 1 • writes to ODR3, then reads 0 from bit IRQiE3 and writes 1 • writes to ODR4, then reads 0 from bit IRQiE4 and writes 1	Internal CPU • writes 0 to bit IRQiE2, o reads ODR2 • CPU writes 0 to bit IRQi host reads ODR3 • CPU writes 0 to bit IRQi host reads ODR4
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 1, IEDIR3 = 1, or IEDIR4 = 1)	Internal CPU • reads 0 from bit IRQiE2, then writes 1 • reads 0 from bit IRQiE3, then writes 1 • reads 0 from bit IRQiE4, then writes 1	Internal CPU • writes 0 to bit IRQiE2 • writes 0 to bit IRQiE3 • writes 0 to bit IRQiE4

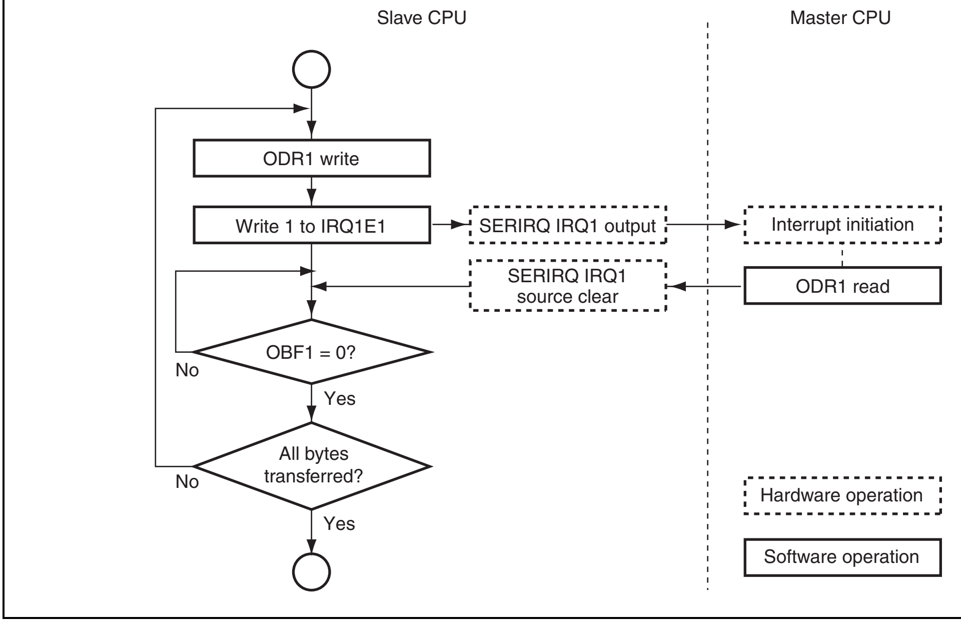


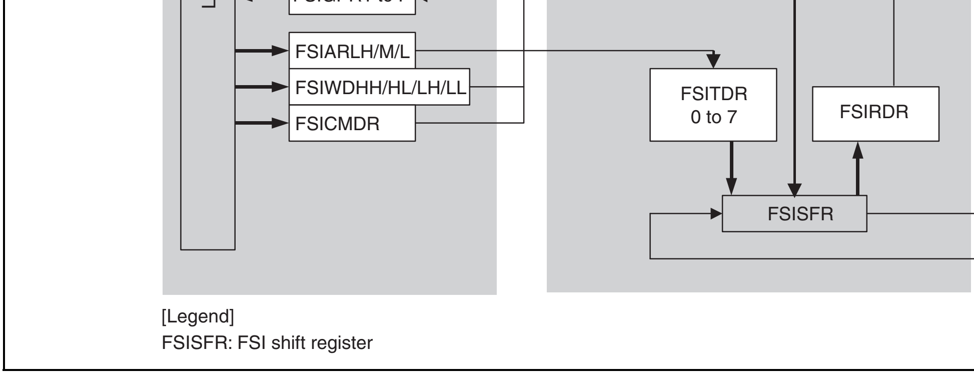
Figure 20.8 HIRQ Flowchart (Example of Channel 1)

Unlike the IDR and ODR registers, the transfer direction is not fixed for the bidirectional registers (TWR). MWMF and SWMF are provided in STR to handle this situation. After to TWR0, MWMF and SWMF must be used to confirm that the write authority for TWR TWR15 has been obtained.

Table 20.12 shows host address examples for LADR3 and registers, IDR3, ODR3, STR3, TWR0MW, TWR0SW, and TWR1 to TWR15.

TWR3	H'A253	H'3FC3
TWR4	H'A254	H'3FC4
TWR5	H'A255	H'3FC5
TWR6	H'A256	H'3FC6
TWR7	H'A257	H'3FC7
TWR8	H'A258	H'3FC8
TWR9	H'A259	H'3FC9
TWR10	H'A25A	H'3FCA
TWR11	H'A25B	H'3FCB
TWR12	H'A25C	H'3FCC
TWR13	H'A25D	H'3FCD
TWR14	H'A25E	H'3FCE
TWR15	H'A25F	H'3FCF

- Supports communications between this LSI and SPI flash memory.
- Can operate as a master.
- Transfer clock selectable from system clock ϕ or LCLK.
- Four interrupt sources: Transmit end, receive data full, and command and write rece interrupts
- Direct transfer between LPC and SPI: Supports Read instruction, and Byte/Page-Pro AAI-Program instructions.
- LPC-SPI command transfer: Supports instructions other than above.
- Supports LPC/FW memory cycles of the LPC interface.
- Supports byte, word, and longword transfers of FW memory cycles.
- Provides independent LPC communication enable bits
- Supports LPC reset and LPC shut-down.



[Legend]
FSISFR: FSI shift register

Figure 21.1 FSI Block Diagram

FSI master data output	FSIDO	Output	FSI address/direction/data output si

For details on the input/output pins of the LPC interface, see section 20.2, Input/Output

Table 21.2 shows the initial state of the FSI input/output pins when the FSIE bit in the F register is set to 1.

Table 21.2 Initial State of FSI Pins (when FSIE = 1)

Pin Name	Symbol	Pin State When FSIE is Set to 1
FSI slave select	FSISS	Outputs high level.
FSI clock	FSICK	Outputs high level or low level depending on CP CPOS.
FSI master data input	FSIDI	Inputs data.
FSI master data output	FSIDO	Outputs high level.

FSI byte count register	FSIBNR	R/W	—	H'00	H
FSI instruction register	FSIINS	R/W	—	H'00	H
FSI read instruction register	FSIRDINS	R/W	—	H'00	H
FSI program instruction register	FSIPPINS	R/W	—	H'00	H
FSI status register	FSISTR	R/W	—	H'00	H
FSI transmit data register 0	FSITDR0	R/W	—	H'00	H
FSI transmit data register 1	FSITDR1	R/W	—	H'00	H
FSI transmit data register 2	FSITDR2	R/W	—	H'00	H
FSI transmit data register 3	FSITDR3	R/W	—	H'00	H
FSI transmit data register 4	FSITDR4	R/W	—	H'00	H
FSI transmit data register 5	FSITDR5	R/W	—	H'00	H
FSI transmit data register 6	FSITDR6	R/W	—	H'00	H
FSI transmit data register 7	FSITDR7	R/W	—	H'00	H
FSI receive data register	FSIRDR	R	—	H'00	H
FSI access host base address register H	FSIHBARH	R/W	—	H'00	H
FSI access host base address register L	FSIHBARL	R/W	—	H'00	H
FSI flash memory size register	FSISR	R/W	—	H'00	H
FSI command host base address register H	CMDHBARH	R/W	—	H'00	H
FSI command host base address register L	CMDHBARL	R/W	—	H'00	H
FSI command register	FSICMDR	R	—	H'00	H
FSILPC command status register 1	FSILSTR1	R/W	R	H'00	H

FSI general-purpose register 8	FSIGPR8	R/W	R	H'00
FSI general-purpose register 9	FSIGPR9	R/W	R	H'00
FSI general-purpose register A	FSIGPRA	R/W	R	H'00
FSI general-purpose register B	FSIGPRB	R/W	R	H'00
FSI general-purpose register C	FSIGPRC	R/W	R	H'00
FSI general-purpose register D	FSIGPRD	R/W	R	H'00
FSI general-purpose register E	FSIGPRE	R/W	R	H'00
FSI general-purpose register F	FSIGPRF	R/W	R	H'00
FSILPC control register	SLCR	R/W	—	H'00
FSI address register H	FSIARH	R	—	H'00
FSI address register M	FSIARM	R	—	H'00
FSI address register L	FSIARL	R	—	H'00
FSI write data register HH	FSIWDRHH	R	—	H'00
FSI write data register HL	FSIWDRHL	R	—	H'00
FSI write data register LH	FSIWDRLH	R	—	H'00
FSI write data register LL	FSIWDRLL	R	—	H'00
FSI LPC command status register 2	FSILSTR2	R/W	—	H'01

Note: 1. Before accessing these registers, clear bit 0 in MSTPCRL (MSTP0) and bit 2 MSTPCRA (MSTPA2) to 0.

2. "R/W" in table 21.3 has the following meanings.

a) "R/W EC" indicates the access from the EC (Embedded Controller = this L

b) "R/W Host" indicates the access from the host.

					1: Clears the internal sequencer. Writing 1 to this bit generates a clear signal f sequencer in the corresponding module, resu the initialization of the FSI's internal state.
6	FSIE	0	R/W	—	FSI Enable 0: Disables FSI operation. 1: Enables FSI operation. The following shows the initial state of the FS when FSIE is set to 1: FSISS: Outputs high level. FSICK: Outputs high level or low level depen DPHS and CPOS. FSIDO: Outputs high level. FSIDI: Inputs data.
5	FRDE	0	R/W	—	Fast-Read Enable 0: The FSI is in normal read operation mode. 1: The FSI is in fast-read operation mode.
4	AAIE	0	R/W	—	AAI (Auto Address Increment) Program Enab 0: The FSI performs byte-program operation. 1: The FSI performs AAI program operation.

					falling edge. 0 1 Setting prohibited 1 0 Setting prohibited
1	—	0	R/W	—	Reserved The initial value should not be modified.
0	CKSEL	0	R/W	—	Clock select 0: Selects the system clock for FSICK 1: Selects LCLK for FSICK Note: Before selecting LCLK for FSICK, cle CPHS and CPOS bits of FSICR1 to 0.

					0: FSI transmission wait state [Clearing condition] When FSI data transmission is completed. 1: When LFBUSY = 0: Starts transmission. When LFBUSY = 1: FSI transmission is in p (automatically set).
6	RE	0	R/W	—	FSI Reception Enable Controls FSI reception and indicates receptio in combination with the LFBUSY bit. 0: FSI reception wait state [Clearing condition] When FSI data reception is completed. 1: When LFBUSY = 0: Starts reception. When LFBUSY = 1: FSI reception is in prog (automatically set).
5	FSITEIE	0	R/W	—	FSI Transmit End Interrupt Enable 0: Disables the FSITEI interrupt request. 1: Enables the FSITEI interrupt request.
4	FSIRXIE	0	R/W	—	FSI Receive Interrupt Enable 0: Disables the FSIRXI interrupt request. 1: Enables the FSIRXI interrupt request.
3 to 0	—	All 0	R/W	—	Reserved The initial value should not be modified.

	TBN0	0			one byte of FSI data transmission is comple the FSI transmission ends, TBN is cleared t 0000: Transmits no data 0001: Transmits one byte of data 0010: Transmits two bytes of data 0011: Transmits three bytes of data 0100: Transmits four bytes of data 0101: Transmits five bytes of data 0110: Transmits six bytes of data 0111: Transmits seven bytes of data 1000: Transmits eight bytes of data 1001 to 1111: Setting prohibited If transmission of nine bytes or more is spec in FSITDR7 is transmitted.
3	—	0	R/W	—	Reserved The initial value should not be modified.

010: Receives two bytes of data

011: Receives three bytes of data

100: Receives four bytes of data

101 to 111: Setting prohibited

If reception of five bytes or more is specified,
is overwritten.

21.3.4 FSI Instruction Register (FSIINS)

FSIINS sets an instruction to be sent to the SPI flash memory during command transfer. V
LFBUSY is 1, a write to this register by the EC (this LSI) is invalid. This register should
in the processing other than FSICMDI and FSIWI interrupt processing.

			R/W		
Bit	Bit Name	Initial Value	EC	Host	Description
7 to 0	bit 7 to bit 0	All 0	R/W	—	These bits store an instruction to be transmit SPI flash memory.

21.3.6 FSI Program Instruction Register (FSIPPINS)

FSIPPINS sets a program operation instruction to be sent to FSITDR during program op When LFBUSY is set to 1, a write to this register by the EC (this LSI) is invalid. This re should be modified during initialization.

		Initial	R/W		
Bit	**Bit Name**	**Value**	**EC**	**Host**	**Description**
7 to 0	bit 7 to bit 0	All 0	R/W	—	These bits store a program operation instru

21.3.7 FSI Status Register (FSISTR)

FSISTR indicates the processing status of the EC (this LSI) and the SPI flash memory tr

		Initial	R/W		
Bit	**Bit Name**	**Value**	**EC**	**Host**	**Description**
7	FSITEI	0	R/(W)*	—	FSI Transmit End Interrupt Flag [Setting condition] When write data has been transmitted to th memory. [Clearing condition] When this bit is read as 1 and then written

					1: There is write data. [Setting condition] When the TE bit is set to 1.
5	FSIRXI	0	R	—	FSI Receive End Interrupt Flag Indicates whether or not there is data to be re the EC (this LSI). 0: There is no read data. [Clearing condition] • LFBUSY = 0: When all receive data has b by the EC (when RBN is cleared to 0). • LFBUSY = 1: When all receive data has b read by the host (automatically cleared). 1: There is read data. [Setting condition] When receive data has been transferred to F
4 to 0	—	All 0	R/W	—	Reserved The initial value should not be modified.

Note: * Only 0 can be written to bit 7 to clear it.

7 to 0	bit 7 to bit 0	All 0	R/W	—	These bits store transmit data.

21.3.9 FSI Receive Data Register (FSIRDR)

FSIRDR stores a total of 4 bytes of receive data items continuously sent from the SPI fla memory. This register should not be read in the processing other than FSICMDI interrup processing. Note that four bytes of receive registers share a single register address. A reg read will be determined according to the RBN bits in FSIBNR. When RBN = B'000, H'0 out.

		Initial	R/W		
Bit	Bit Name	Value	EC	Host	Description
7 to 0	bit 7 to bit 0	All 0	R	—	These bits store receive data.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 31 to bit 24	All 0	R/W	—	These bits specify bits [31:24] of the host sta address.

- FSIHBARL

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 23 to bit 16	All 0	R/W	—	These bits specify bits [23:16] of the host sta address. The settings by bit 19 to bit 16 do not affect t operation.

21.3.11 FSI Flash Memory Size Register (FSISR)

FSISR sets the size of SPI flash memory. The host input address range will be determined on the size set in this register. Note that the host input address should not be greater than flash memory capacity. During FSI operation (in the state where FSIE or FSILIE is set), change the setting in this register.

21.3.12 FSI Command Host Base Address Registers H and L (CMDHBARH and CMDHBARL)

CMDHBARH and CMDHBARL set the upper 16 bits of the host start address which is to set a command address. The lower 16 bits of the host start address range from H'F00 H'F00F. If a host address to be input to CMDHBARH and CMDHBARL is out of the de range, Sync will not be returned. If FW memory cycle is used, bit 31 to bit 28 in CMDH set as IDSEL. During FSI operation (in the state where FSIE or FSILIE is set), do not ch setting in this register.

- CMDHBARH

		Initial	R/W		
Bit	Bit Name	Value	EC	Host	Description
7 to 0	bit 31 to bit 24	All 0	R/W	—	These bits specify bits [31:24] of the host st address.

- CMDHBARL

		Initial	R/W		
Bit	Bit Name	Value	EC	Host	Description
7 to 0	bit 23 to bit 16	All 0	R/W	—	These bits specify bits [23:16] of the host st address.

21.3.14 FSI LPC Command Status Register 1 (FSILSTR1)

FSILSTR1 indicates the LPC internal status.

			R/W		
Bit	**Bit Name**	**Initial Value**	**EC**	**Host**	**Description**
7	CMDBUSY	0	R/W*	R	FSI Command Busy Flag 0: The FSI command execution is complete [Clearing condition] • When this bit is read as 1 and then writte 1: The FSI command execution is in progres [Setting condition] • When an FSI command is received.
6	FSICMDI	0	R/W*	R	FSI Command Interrupt Flag 0:The FSI command interrupt processing is completed. [Clearing condition] • When this bit is read as 1 and then writte 1: The FSI command interrupt processing is progress. [Setting condition] • When an FSI command is received.

					1: FSI write in transferring [Setting condition] • SPI flash memory write is received whe FLDCT=0.
3	FSIWI	0	R/W*	R	FSI Write Interrupt Flag 0: FSI write interrupt is completed. [Clearing condition] • Read FSIWI=1 and then write 0. 1: FSI write interrupt is in progress. [Setting condition] • SPI flash memory write is received whe FLDCT=0.
2	FLBUSY	0	R	R	LPC-SPI Direct Transfer Busy Flag Indicates an LPC-SPI direct transfer status. 0: Direct transfer is completed. 1: During direct transfer
1, 0	—	All 0	R/W	R	Reserved The initial value should not be modified.

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

					Indicates a FSI write transfer status duri SPI direct transfer. 0: FSI write transfer is completed. 1: During FSI write transfer
3	FSIDRBUSY	0	R	—	FSI Direct Read Busy Flag Indicates a FSI read transfer status durin SPI direct transfer. 0: FSI read transfer is completed. 1: During FSI read transfer
2 to 0	SIZE2 SIZE1 SIZE0	0 0 1	R R R	— — —	Transfer Byte Count Monitor Indicates the number of transferred byte data is received in the LPC/FW memory When the Byte/Page-Program or AAI-Pr instruction is executed from the EC CPU number of transferred bytes can be conf these bits. 001: LPC/FW memory cycle (byte transf 010: FW memory cycle (word transfer) 100: FW memory cycle transfer (longwor transfer) When a transfer is made in units other th byte/word/longword, the previous value i retained. Note: This bit is not set to the value othe above.

21.3.17 FSI LPC Control Register (SLCR)

SLCR enables or disables the LPC host interface function of the FSI, FSI interrupt enab
FSI operation mode control bit.

		Initial	R/W		
Bit	**Bit Name**	**Value**	**EC**	**Host**	**Description**
7	FSILIE	0	R/W	—	FSI LPC Interface Enable Enables or disables the LPC host interface f the FSI. When disabled, address-matching i performed and Sync is not returned. 0: Disables the LPC host interface function. 1: Enables the LPC host interface function.
6	FSICMDIE	0	R/W	—	FSI Command Interrupt Enable 0: Disables the FSI command interrupt. 1: Enables the FSI command interrupt.
5	FSIWIE	0	R/W	—	FSI Write Interrupt Enable 0: Disables the FSI write interrupt. 1: Enables the FSI write interrupt.

					Selects access mode in SPI Flash memory w details, see section 21.4.6, SPI Flash Memory Operation Mode. 0: No wait cycle is inserted. 1: Wait cycles can be inserted.
2 to 0	—	All 0	R/W	—	Reserved The initial value should not be modified.

21.3.18 FSI Address Registers H, M, and L (FSIARH, FSIARM, and FSIARL)

FSIAR stores an SPI flash memory address. If the host address matches FSIHBAR, the F value is updated. FSIAR value is not updated by command access.

- FSIARH

			R/W		
Bit	**Bit Name**	**Initial Value**	**EC**	**Host**	**Description**
7 to 0	bit 23 to bit 16	All 0	R	—	These bits store bits [23:16] of the SPI flash r address.

- FSIARM

			R/W		
Bit	**Bit Name**	**Initial Value**	**EC**	**Host**	**Description**
7 to 0	bit 15 to bit 8	All 0	R	—	These bits store bits [15:8] of the SPI flash me address.

FSIWDR stores data to be written to the SPI flash memory. If the host address matches during LPC/FW memory write cycle, the FSIWDR value will be updated. FSIHBAR va updated by command access.

- FSIWDRHH

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 31 to bit 24	All 0	R	—	These bits store bits [31:24] of the SPI flash write data

- FSIWDRHL

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 23 to bit 16	All 0	R	—	These bits store bits [23:16] of the SPI flash write data.

Bit	Bit Name	Initial Value	EC	Host	Description
7 to 0	bit 7 to bit 0	All 0	R	—	These bits store bits [7:0] of the SPI flash me write data.

in the registers (FSIHBARH, FSIHBARL, FSISR, and CMDHBAR), and outputs a turn
signal to return to the idle state.

Table 21.4 LPC Memory Read/Write Cycles

State Counts	LPC Memory Read Cycles			LPC Memory Write Cyc		
	Content	**Driven by**	**Value (3 to 0)**	**Content**	**Driven by**	**Valu**
1	Start	Host	0000	Start	Host	000
2	Cycle type/ direction	Host	0100	Cycle type/ direction	Host	011
3	Address 1	Host	bit 31 to bit 28	Address 1	Host	bit 3
4	Address 2	Host	bit 27 to bit 24	Address 2	Host	bit 2
5	Address 3	Host	bit 23 to bit 20	Address 3	Host	bit 2
6	Address 4	Host	bit 19 to bit 16	Address 4	Host	bit 1
7	Address 5	Host	bit 15 to bit 12	Address 5	Host	bit 1
8	Address 6	Host	bit 11 to bit 8	Address 6	Host	bit 1
9	Address 7	Host	bit 7 to bit 4	Address 7	Host	bit 7
10	Address 8	Host	bit 3 to bit 0	Address 8	Host	bit 3
11	Turn-around (recovery)	Host	1111	Data 1	Host	bit 3
12	Turn-around	None	ZZZZ	Data 2	Host	bit 7
13	Wait*	Slave	0110	Turn-around (recovery)	Host	111

Table 21.5 FW Memory Read/Write Cycles (Byte Transfer)

State	FW Memory Read Cycles			FW Memory Write Cycle		
Counts	Content	Driven by	Value (3 to 0)	Content	Driven by	Value
1	Start	Host	1101	Start	Host	1110
2	Device select	Host	ID3 to ID0	Device select	Host	ID3 to
3	Address 1	Host	bit 27 to bit 24	Address 1	Host	bit 27
4	Address 2	Host	bit 23 to bit 20	Address 2	Host	bit 23
5	Address 3	Host	bit 19 to bit 16	Address 3	Host	bit 19
6	Address 4	Host	bit 15 to bit 12	Address 4	Host	bit 15
7	Address 5	Host	bit 11 to bit 8	Address 5	Host	bit 11
8	Address 6	Host	bit 7 to bit 4	Address 6	Host	bit 7 t
9	Address 7	Host	bit 3 to bit 0	Address 7	Host	bit 3 t
10	Size	Host	0000	Size	Host	0000
11	Turn-around (recovery)	Host	1111	Data 1	Host	bit 3 t
12	Turn-around	None	ZZZZ	Data 2	Host	bit 7 t
13	Wait*	Slave	0110	Turn-around (recovery)	Host	1111
14	Synchronization	Slave	0000	Turn-around	None	ZZZZ
15	Data 1	Slave	bit 3 to bit 0	Wait*	Slave	0110
16	Data 2	Slave	bit 7 to bit 4	Synchronization	Slave	0000

operating frequency of the system clock is 10 MHz.

21.4.2 SPI Flash Memory Transfer

The SPI flash memory transfer is performed using FSIDO and FSIDI synchronously wit
The initial value of FSICK can be either fixed to high or low through programming.

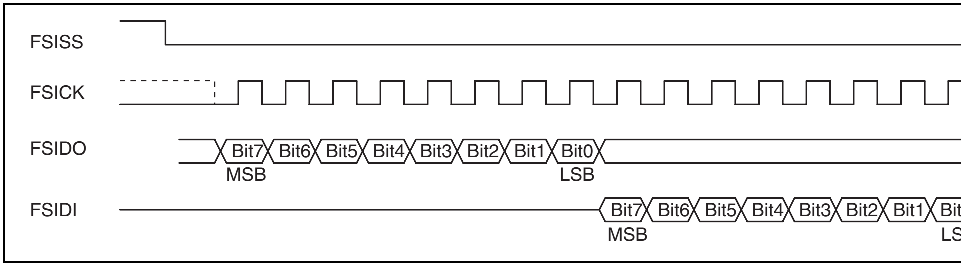
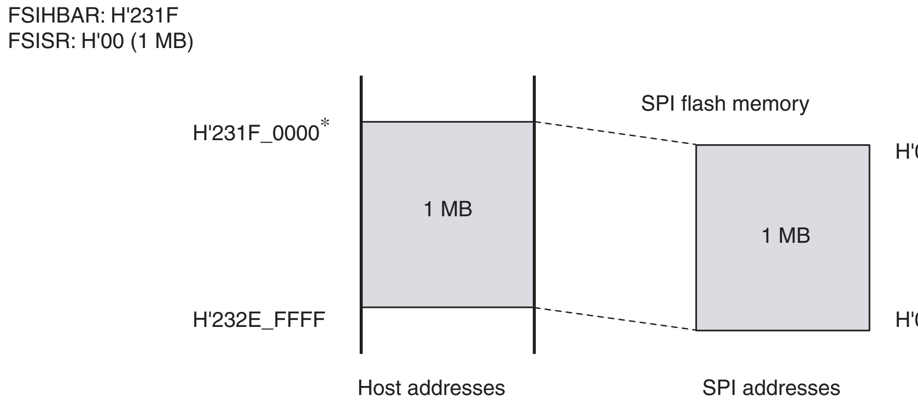


Figure 21.2 Example of SPI Flash Memory Transfer

WRSR	Writes status register
READ	Reads SPI flash memory
Fast-Read	Fast-reads SPI flash memory
Byte-Program	Byte-programs SPI flash memory
Page-Program	Page-programs SPI flash memory
AAI-Program	Address auto increment program
Sector-Erase	Sector erasure
Block-Erase	Block erasure
Chip/Bulk-Erase	Chip/bulk erasure
RDID	Reads manufacturing ID and product ID
EWSR	Enables status register write
DP (DEEP POWER DOWN)	Deep power-down
RES	Releases deep power-down

(1) FSI Address Conversion

The host address can be converted into the SPI flash memory address by setting FSIHBA FSIHBARL, and FSISR. The host address space ranges from H'0000_0000 to H'FFFF_F SPI flash memory address space ranges from H'00_0000 to H'FF_FFFF. Figure 21.3 sho example of the FSI memory address conversion.



Note: * The upper 16 bits of the host address are set to the value in the FSIHBAR register.

Figure 21.3 FSI Address Conversion Example

As shown in figure 21.3, if an address ranging from H'231F_0000 to H'232E_FFFF is ac LPC/FW memory write cycle, the SPI flash memory is accessed. If a host address to be out of the determined range, Sync will not be returned. During an SPI flash memory acc wait cycle will be inserted to the LPC bus cycle. In an LPC memory cycle, one-byte tran enabled. In an FW memory cycle, a byte, word, and a longword transfer are enabled.

FSITDR. Figure 21.5 shows the Page-Program instruction execution timing.

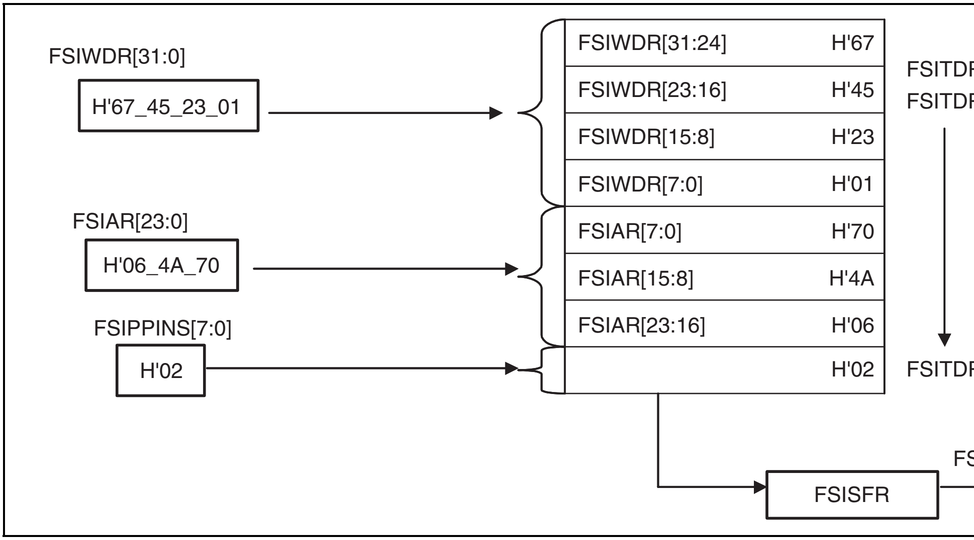


Figure 21.4 Data Transfer to FSITDR (Example)

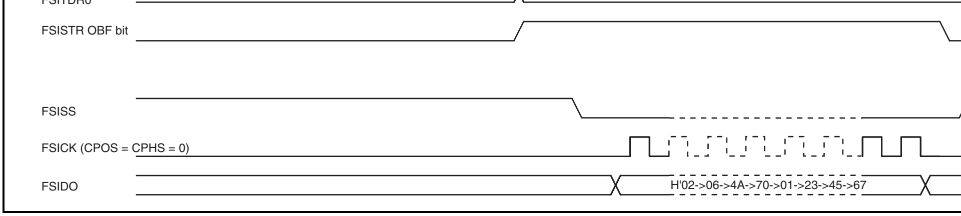


Figure 21.5 Page-Program Instruction Execution Timing

returned to the host. To execute the AAI-Program instruction, byte transfer access in LPC write cycle or FW memory write cycle should be performed. To return to the AAI-Progra instruction (first byte), clear the AAIE bit once or perform initialization of the FSI interna sequencer in SRES of FSICR1. After the Read instruction or the LPC-SPI command is tr during the AAI-Program instruction execution, the FSI internal sequencer is initialized to the AAI-Program Instruction (first byte). Figures 21.6 and 21.7 show AAI-Program execu timings.

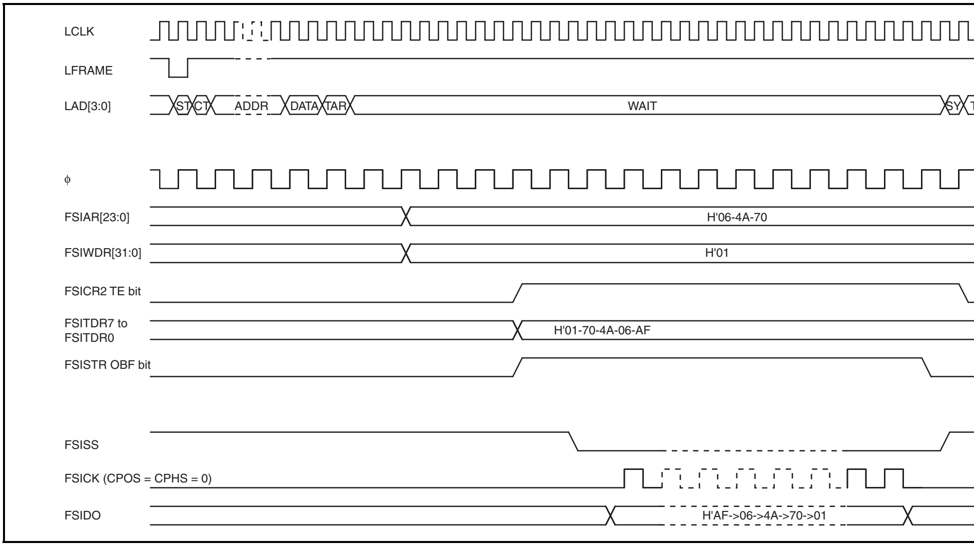


Figure 21.6 AAI-Program Instruction Execution Timing (First Byte)

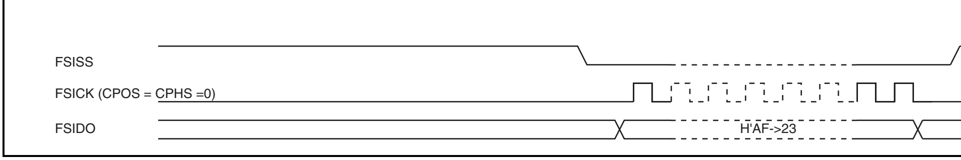


Figure 21.7 AAI-Program Instruction Execution Timing (Second and Following Bytes)

(4) Read Instructions

If an LPC/FW memory read cycle occurs while the FRDE bit in FSICR1 is cleared to 0, flash memory address is stored in FSIAR. Then, the SPI flash memory address and the i which is stored in FSIRDINS in advance are transferred to FSITDR. After SYNC (long been returned, the RE bit in FSICR2 is set, and Read instruction execution starts. The re then received and stored in FSIRDR. When the reception has been completed, SYNC (R read data, and TAR are returned to the host. Figure 21.8 shows an example of data trans FSIRDR. Figure 21.9 shows the Read instruction execution timing.

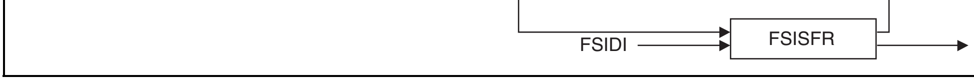


Figure 21.8 Data Transfer to FSIRDR (Example)

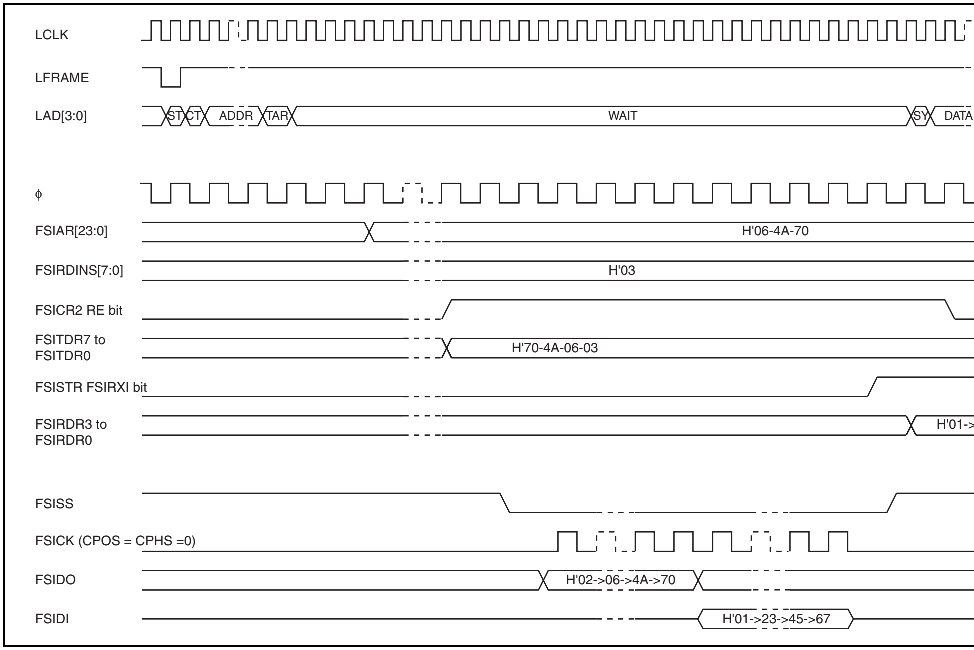


Figure 21.9 Read Instruction Execution Timing

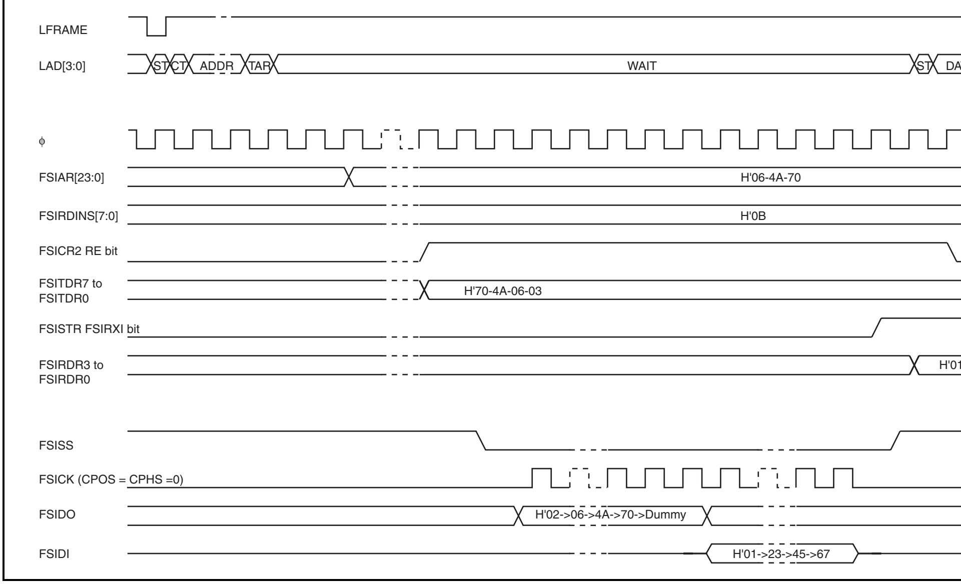
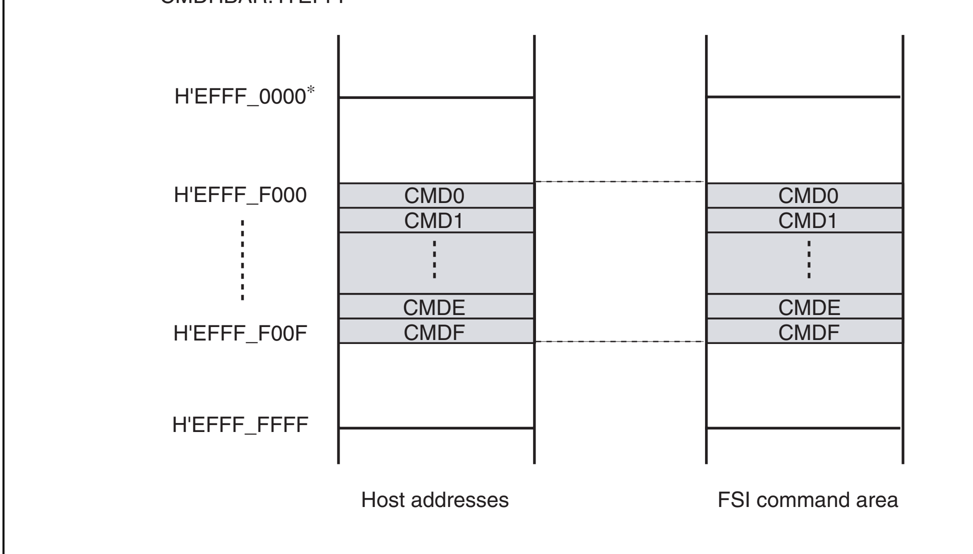


Figure 21.10 Fast-Read Instruction Execution Timing



Note: * The upper 16 bits of the host address are set to the value in the CMDHBAR register.

Figure 21.11 FSI Command Space Settings (Example)

As shown in figure 21.11, a host address ranging from H'EFFF_F000 to H'EFFF_F00F is the FSI command space while the CMDHBAR register is set to H'EFFF.

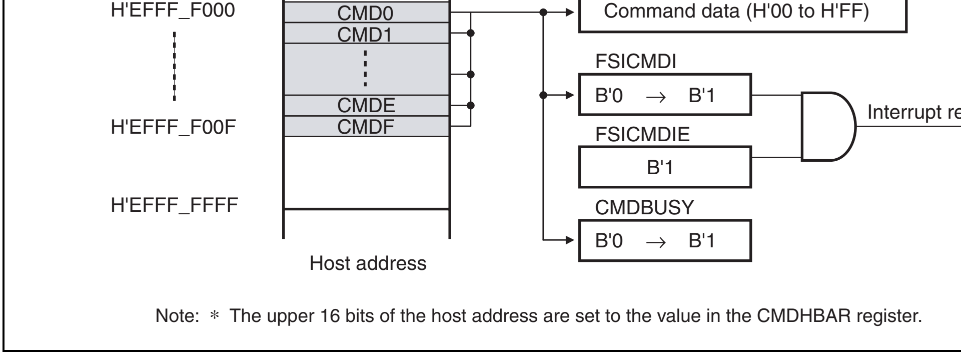
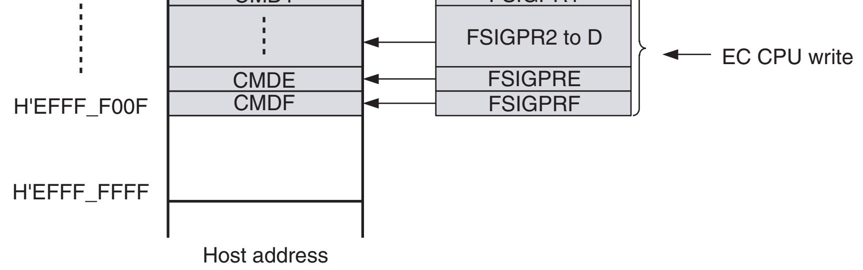

H'EFFF_F000
H'EFFF_F00F
H'EFFF_FFFF
CMD0
CMD1
CMDE
CMDF
Host address
Command data (H'00 to H'FF)
FSICMDI
B'0 → B'1
FSICMDIE
B'1
CMDBUSY
B'0 → B'1
Interrupt re
Note: * The upper 16 bits of the host address are set to the value in the CMDHBAR register.

Figure 21.12 FSI Command Write Operation (Example)

As shown in figure 21.12, if a host address ranging from H'EFFF_F000 to H'EFFF_F00
accessed in LPC/FW memory write cycle while the CMDHBAR register is set to H'EFF
write data is stored in FSICMDR, and then the CMDBUSY and FSICMDI flags in FSIL
set to 1. In this case, an interrupt is requested according to the FSICMDIE state. Sync is
returned if the host address to be input is out of the determined range. In FSI command
wait cycle will be inserted to the LPC bus cycle. If the CMDBUSY flag is set to 1, Sync
returned during the operations other than FSI command read.



Note: * The upper 16 bits of the host address are set to the value in the CMDHBAR register.

Figure 21.13 FSI Command Read (Example)

As shown in figure 21.13, if a host address ranging from H'EFFF_F000 to H'EFFF_F00F accessed in LPC/FW memory read cycle while the CMDHBAR register is set to H'EFFF, FSILSTR1 or data in FSIGPR1 to FSIGPRF is returned. Sync is not returned if the host a be input is out of the determined range. In FSI command read, no wait cycle will be inser LPC bus cycle. Before reading the FSIGPR, ensure that the CMDBUSY bit in FSILSTR1 been cleared to 0.

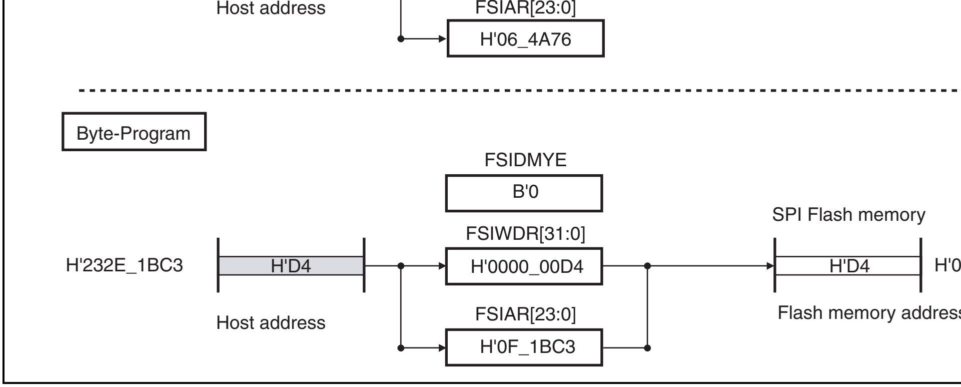


Figure 21.14   FSI Dummy Write (Example)

As shown in figure 21.14, if an LPC/FW memory write cycle occurs while the FSIDMY
FSILSTR1 is 1, the FSI does not access the SPI flash memory but stores the SPI flash m
address and write data in FSIAR and FSIWDR, respectively.

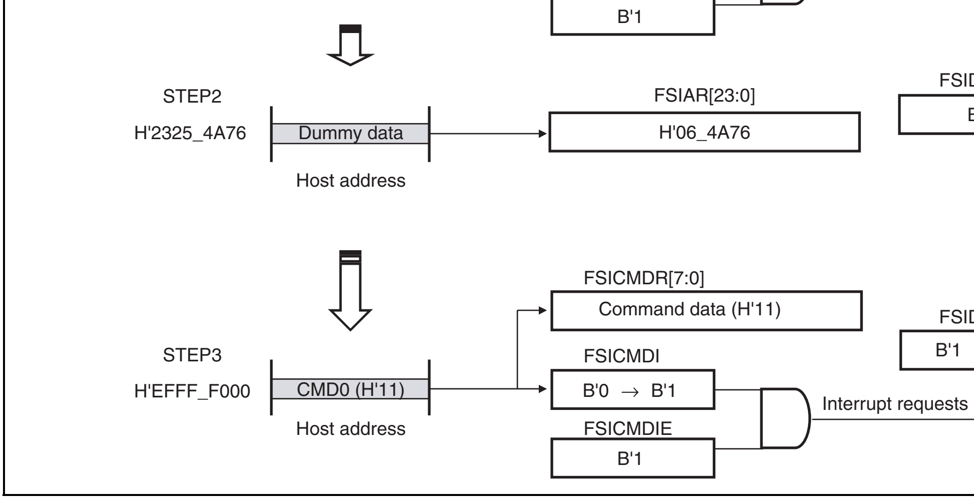


Figure 21.15 SPI Flash Memory Erasure (Example)

In flash memory erasure, the SPI flash memory address is stored in FSIAR and an erasure instruction for the SPI flash memory is executed by an SPI command. The flash memory storage in FSIAR is performed by writing data to the sector or block address to be erased host. To distinguish the SPI flash memory erasure from the SPI flash memory programmi erasure is performed in the following sequence using the FSIDMYE.

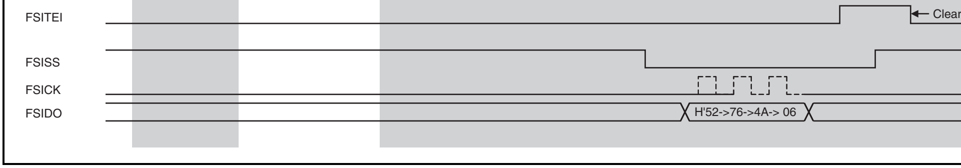


Figure 21.16 Execution Timing of SPI Flash Memory

Step 1:

1. Write an erasure setting command (Host).
2. Generate an FSICMDI interrupt request.
3. Set the FSIDMYE bit in FSILSTR1 to 1 and clear the FSICMDI and CMDBUSY bi FSILSTR1 to 0.
4. Complete the interrupt processing.
5. Check that the FSIDMYE bit in FSILSTR1 is set to 1 and that the CMDBUSY and F bits in FSILSTR1 are cleared to 0 (Host).

Step 2:

1. Perform a dummy write to the sector or block address to be erased (Host).
2. Store the SPI flash memory address and write data in the FSIAR register and FSIWI register, respectively*.

Note: * Use the data stored in FSIWDR if necessary on the user side.

Step 3:

1. Write an erasure setting command (Host).
2. Generate an FSICMDI interrupt request.
3. Clear the FSICMDI bit in FSILSTR1 to 0.

8. Complete the interrupt processing.

9. Check that the FSIDMYE, CMDBUSY, and FSICMDI bits in FSILSTR1 are cleared (Host).

(6) FSI Command Usage Example 2 (SPI Flash Memory Status Read)

Figure 21.17 shows an example of the execution timing of the SPI flash memory status re instruction.

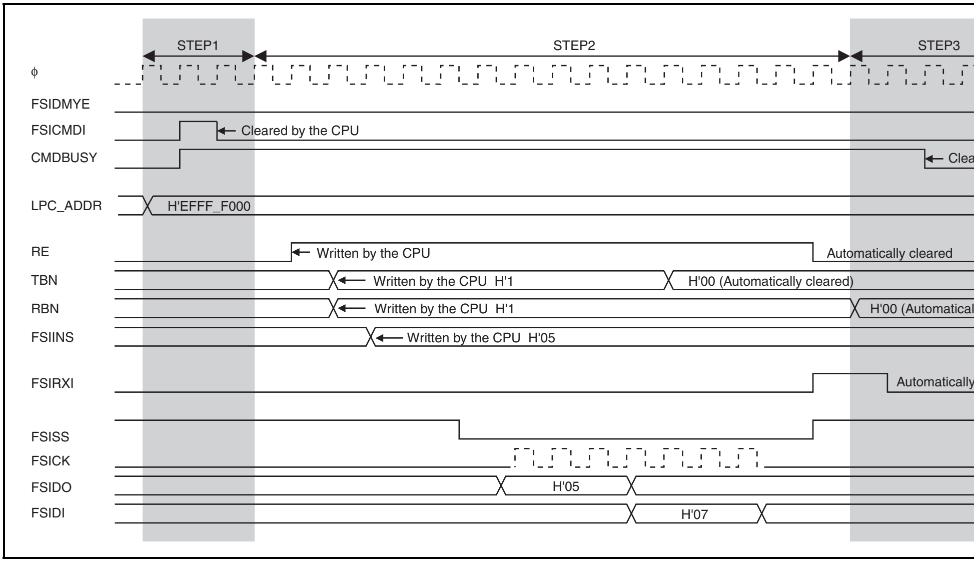


Figure 21.17 Execution Timing of SPI Flash Memory Status Read Instructi

The SPI flash memory status read instruction is executed in the following sequence.

— Set the RE bit in FSICR2 to 1.

— Set the TBN bit in FSIBNR to 1-byte transfer and set the RBN bit in FSIBNR to reception.

— Write the status read instruction to FSIINS (start the SPI flash memory status rea instruction execution).

2. Complete the interrupt processing.

Step 3:

1. Generate an FSIRXI interrupt request.
2. Write read data stored in FSIRDR to SPIGPR.
3. Clear the CMDBUSY bit in FSILSTR1 to 0.
4. Complete the interrupt processing
5. Check that the CMDBUSY and FSICMDI bits in FSILSTR1 are cleared to 0 (Host).
6. Read the SPI flash memory status from FSIGPR (Host).

			FSIWI ← 1	flash memory by the EC CPU. cycle is inserted to the LPC bu by FSIWBUSY whether or not transfer has been completed.

Mode 2	0	1	FSIWBUSY ← 1
			FSIWI ← 1
				Control the write operation to t flash memory by the EC CPU. cycles are inserted to the LPC Provision of wait cycles can be by clearing FSIWBUSY.

Mode 3	1	0	LFBUSY ← 1 (Automatically cleared)	Control the write operation to t flash memory by the FSI. No w inserted to the LPC bus. Confi LFBUSY whether or not a writ has been completed.
Mode 4	1	1	LFBUSY ← 1 (Automatically cleared)	Control the write operation to t flash memory by the FSI. Wait inserted to the LPC bus. Provi cycles can be canceled by cle LFBUSY.

FSISR	Bits 7 to 0	Initialized	Retained	Retained	Retained	R
CMDHBARH/ CMDHBARL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Re
FSICMDR	Bits 7 to 0	Initialized	Retained	Retained	Retained	Re
FSILSTR1	Bits 7, 6, 4, and 3	Initialized	Initialized	Retained	Retained	Re
	Bit 2	Initialized	Initialized	Retained	Retained	In
	Bits 5, 1 and 0	Initialized	Retained	Retained	Retained	Re
FSILSTR2	Bits 7 to 5	Initialized	Retained	Retained	Retained	Re
	Bits 4 and 3	Initialized	Initialized	Retained	Retained	In
	Bits 2 to 0	Initialized	Retained	Retained	Retained	Re
SPIGPR1 to SPIGPRF	Bits 7 to 0	Initialized	Retained	Retained	Retained	Re
SLCR	Bits 7 to 0	Initialized	Retained	Retained	Retained	Re
FSIARH/ FSIARM/ FSIARL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Re
FSIWDRHH/ FSIWDRHL/ FSIWDRLH/ FSIWDRLL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Re
LPC internal sequencer		Initialized	Initialized	Initialized	Initialized	Re

FSIRDINS	Bits 7 to 0	Initialized	Retained	Retained	Retained	Re
FSIPPINS	Bits 7 to 0	Initialized	Retained	Retained	Retained	Ret
FSISTR	Bit 7	Initialized	Retained	Retained	Retained	Init
	Bits 6 and 5	Initialized	Retained	Retained	Retained	Init
	Bits 4 to 0	Initialized	Retained	Retained	Retained	Ret
FSITDR7 to FSITDR0	Bits 7 to 0	Initialized	Retained	Retained	Retained	Ret
FSIRDR	Bits 7 to 0	Initialized	Retained	Retained	Retained	Ret
FSI internal sequencer		Initialized	Retained	Retained	Retained	Init

Interrupt Name	Interrupt Source		Interrupt Enable Bit
FSII	FSITEI	Transmit end	FSITEIE
	FSIRXI	Receive data full	FSIRXIE
LFSII	FSICMDI	FSI command reception	FSICMDIE
	FSIWI	FSI write reception	FSIWIE

21.7 Usage Note

21.7.1 Longword Transfer in FW Memory Write Cycles

When longword transfers of FW memory write cycles are used, the maximum operating of the system clock is 10 MHz.

- Conversion cycle: 40 cycles (A/D conversion clock)
- Two kinds of operating modes

  Single mode: Single-channel A/D conversion

  Scan mode: Continuous A/D conversion on one to four channels or continuous A/D on one to eight channels
- A/D conversion clocks specifiable (ϕ, ϕ/2, ϕ/4, or ϕ/8)
- Eight data registers

  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of A/D conversion start

  Software

  Conversion start trigger from 16-bit timer pulse unit (TPU) or 8-bit timer (TMR)
- Interrupt source

  A/D conversion end interrupt (ADI) request can be generated

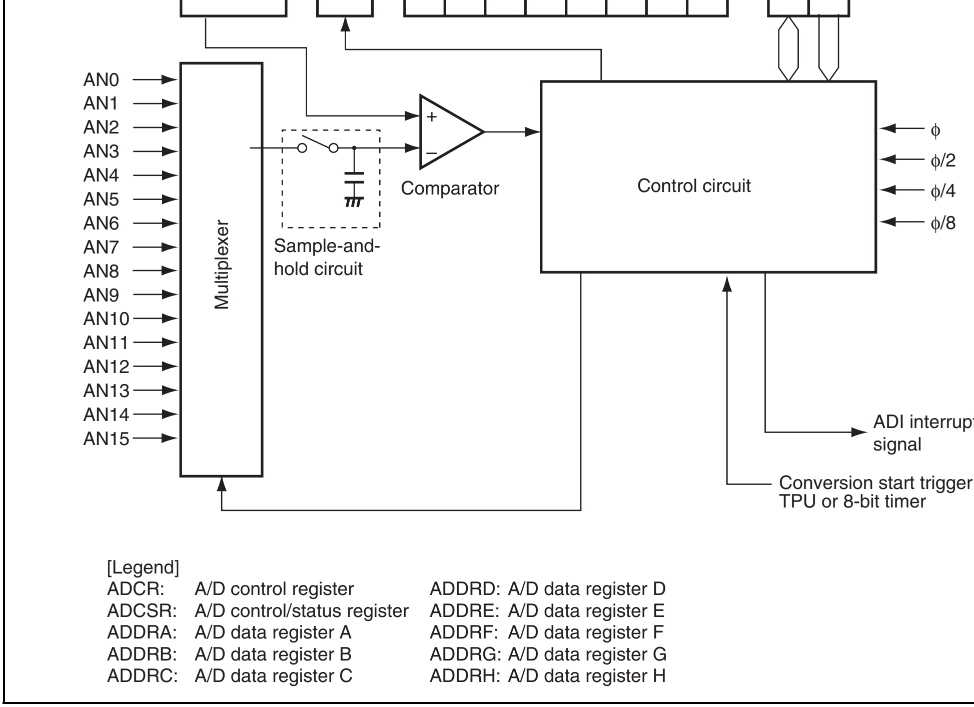

AN0
AN1
AN2
AN3
AN4
AN5
AN6
AN7
AN8
AN9
AN10
AN11
AN12
AN13
AN14
AN15
Multiplexer
Sample-and-
hold circuit
+
–
Comparator
Control circuit
ϕ
ϕ/2
ϕ/4
ϕ/8
ADI interrupt
signal
Conversion start trigger
TPU or 8-bit timer
[Legend]
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D
ADDRE: A/D data register E
ADDRF: A/D data register F
ADDRG: A/D data register G
ADDRH: A/D data register H

Figure 22.1 Block Diagram of A/D Converter

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground
Reference power supply pin	AVref	Input	Reference voltage for A/D converter
Analog input pin 0	AN0	Input	Channel set 0 analog input
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 8	AN8	Input	Channel set 1 analog input
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	

A/D data register C	ADDRC	R	H'0000	H'FC04	16
A/D data register D	ADDRD	R	H'0000	H'FC06	16
A/D data register E	ADDRE	R	H'0000	H'FC08	16
A/D data register F	ADDRF	R	H'0000	H'FC0A	16
A/D data register G	ADDRG	R	H'0000	H'FC0C	16
A/D data register H	ADDRH	R	H'0000	H'FC0E	16
A/D control/status register	ADCSR	R/W	H'00	H'FC10	8
A/D control register	ADCR	R/W	H'00	H'FC11	8

Table 22.3 Analog Input Channels and Corresponding ADDR

Analog Input Channel		A/D Data Register to Store
Channel Set 0 (CH3 = 0)	Channel Set 1 (CH3 = 1)	Conversion Results
AN0	AN8	ADDRA
AN1	AN9	ADDRB
AN2	AN10	ADDRC
AN3	AN11	ADDRD
AN4	AN12	ADDRE
AN5	AN13	ADDRF
AN6	AN14	ADDRG
AN7	AN15	ADDRH

				in scan mode [Clearing condition] When 0 is written after reading ADF = 1
6	ADIE	0	R/W	A/D Interrupt Enable Enables ADI interrupt by ADF when this bit is se
5	ADST	0	R/W	A/D Start When this bit is cleared to 0, A/D conversion sto enters wait state. When this bit is set to 1 by a c start trigger from software, TPU, or TMR, A/D co starts. This bit remains set to 1 during A/D conve single mode, this bit is automatically cleared to 0 conversion on the specified channel ends. In sca conversion continues sequentially on the specifie channels until this bit is cleared to 0 by a reset, c software.
4	—	0	—	Reserved This bit is always read as 0 and cannot be modif

0010: AN2	0010: AN0 to AN2	0010: AN
0011: AN3	0011: AN0 to AN3	0011: AN
0100: AN4	0100: AN4	0100: AN
0101: AN5	0101: AN4, AN5	0101: AN
0110: AN6	0110: AN4 to AN6	0110: AN
0111: AN7	0111: AN4 to AN7	0111: AN
1000: AN8	1000: AN8	1000: AN
1001: AN9	1001: AN8, AN9	1001: AN
1010: AN10	1010: AN8 to AN10	1010: AN
1011: AN11	1011: AN8 to AN11	1011: AN
1100: AN12	1100: AN12	1100: AN
1101: AN13	1101: AN12, AN13	1101: AN
1110: AN14	1110: AN12 to AN14	1110: AN
1111: AN15	1111: AN12 to AN15	1111: AN

[Legend]

X: Don't care

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

				11: Setting prohibited
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Select the A/D conversion operating mode. 0X: Single mode 10: Scan mode Continuous A/D conversion on 1 to 4 channels 11: Scan mode Continuous A/D conversion on 1 to 8 channels
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	These bits select the clock (ADCLK)* used in A/D conversion. Set these bits while the ADST bit in AD then set the conversion mode. 00: φ 01: φ/2 10: φ/4 00: φ/8
1	ADSTCLR	0	R/W	A/D Start Clear Sets the automatic clearing of the ADST bit in scan 0: Disables the automatic clearing of the ADST bit i mode 1: Automatically clears the bit when A/D conversion the selected channels are completed
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

[Legend]

X: Don't care

Note: * Set the clock so that ADCLK ≤ 10 MHz.

Operations are as follows.

1. A/D conversion on the specified channel is started when the ADST bit in ADCSR is software, the TMR, or the TPU.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of A/D conversion, the ADF bit in ADCSR is set to 1. If the ADIE bi 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When conversion ends, the A is automatically cleared to 0 and the A/D converter enters wait state. When the ADS cleared to 0 during A/D conversion, the conversion stops and the A/D converter ente state.

from the following channels: AN0 when CH3 = 0 and CH2 = 0, AN4 when CH3 = 0
= 1, AN8 when CH3 = 1 and CH2 = 0, and AN12 when CH3 = 1 and CH2 = 1.

When continuous A/D conversion on eight channels is selected, A/D conversion starts
the following channels: AN0 when CH3 = 0 and CH2 = 0 and AN8 when CH3 = 1 an
0.

3. When A/D conversion for each channel is completed, the result is sequentially transfe
the A/D data register corresponding to each channel.
4. When conversion of all the selected channels is completed, the ADF bit in ADCSR is
If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. Conversion from
channel in the channel set starts again.
5. The ADST bit is not automatically cleared to 0 so steps [2] and [3] are repeated as lon
ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops
A/D converter enters wait state. After this, setting the ADST bit to 1 starts A/D conve
from the first channel again.
6. When the ADST bit is automatically cleared on completion of the A/D conversion of
selected channels with the ADSTCLR bit in ADCR set to 1, A/D conversion stops an
the wait state.

In scan mode, the values shown in table 22.4 become those for the first conversion time. second and subsequent conversion times are listed in table 22.5. In either case, bits CKS CKS0 in ADCR should be set so that the conversion time is within the ranges indicated A/D conversion characteristics.

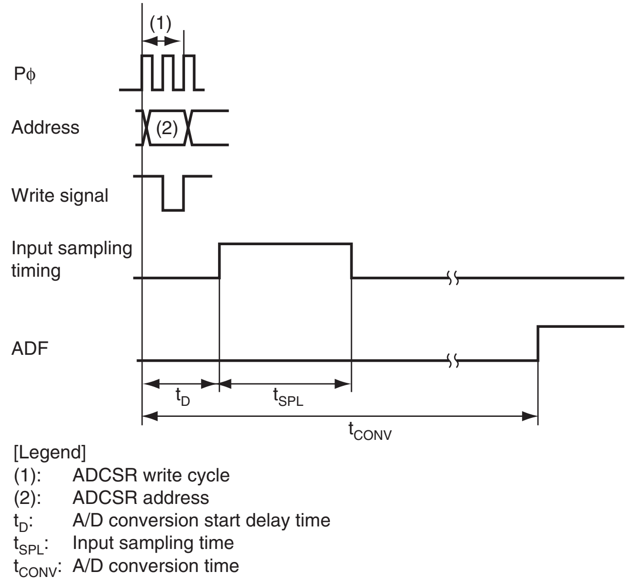


Figure 22.2 A/D Conversion Timing

Table 22.5 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	40 (fixed)
0	1	80 (fixed)
1	0	160 (fixed)
1	1	320 (fixed)

22.6 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 22.3).

- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion chara when the digital output changes from the minimum voltage value B'00 0000 0000 (H B'00 0000 0001 (H'001) (see figure 22.4).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion chara when the digital output changes from B'11 1111 1110 (H'3FE) to B'11 1111 1111 (H figure 22.4).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between the zero v the full-scale voltage. Does not include the offset error, full-scale error, or quantizati (see figure 22.4).

- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offs full-scale error, quantization error, and nonlinearity error.

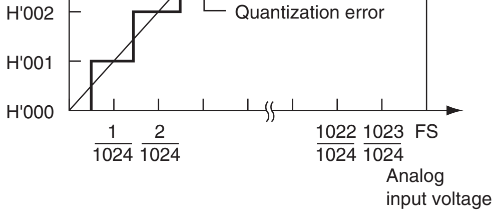


Figure 22.3   A/D Conversion Accuracy Definitions

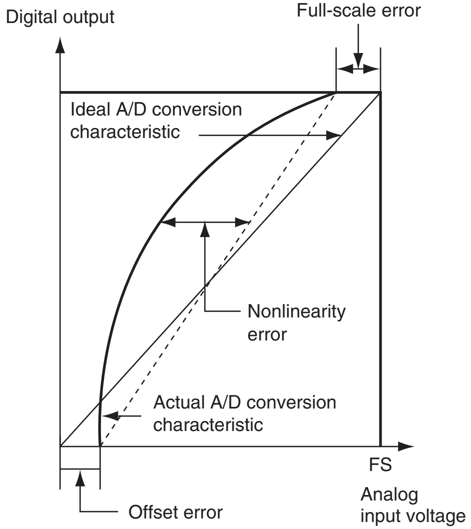


Figure 22.4   A/D Conversion Accuracy Definitions

This LSI’s analog input is designed so that the conversion accuracy is guaranteed for an signal for which the signal source impedance is 5 kΩ or less. This specification is provic enable the A/D converter’s sample-and-hold circuit input capacitance to be charged with sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficie may not be possible to guarantee the A/D conversion accuracy. However, if a large capa provided externally in single mode, the input load will essentially comprise only the inte resistance of 10 kΩ, and the signal source impedance is ignored. However, since a low-p effect is obtained in this case, it may not be possible to follow an analog signal with a la differential coefficient (e.g., voltage fluctuation ratio of 5 mV/µs or greater) (see figure When converting a high-speed analog signal or converting in scan mode, a low-impedar should be inserted.

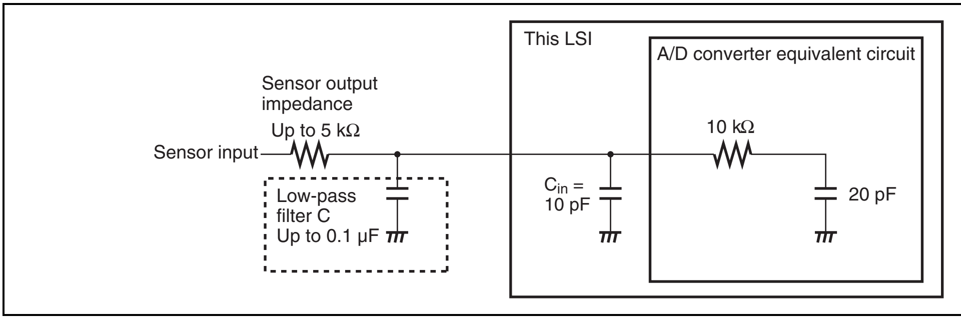


Figure 22.5 Example of Analog Input Circuit

If conditions shown below are not met, the reliability of this LSI may be adversely affect

- Analog input voltage range

  The voltage applied to analog input pins (AN0 to AN15) during A/D conversion shou the range AVss ≤ ANn ≤ AVref (n = 0 to 15).

- Relation between AVcc, AVss and Vcc, Vss

  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc = Vcc ±0.3 V and A Vss. If the A/D converter is not used, set AVcc = Vcc and AVss = Vss.

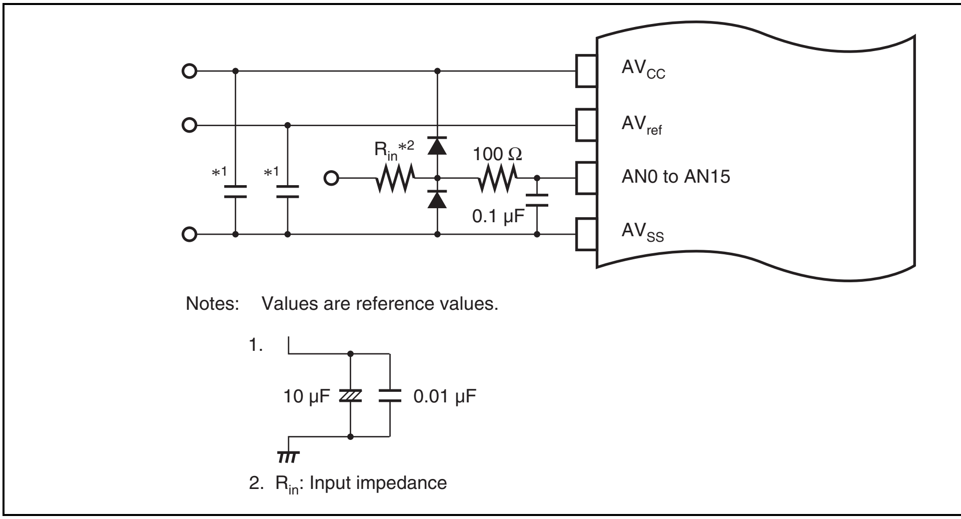
- AVref pin range

  The reference voltage of the AVref pin should be in the range AVref ≤ AVcc.

22.7.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as po and layout in which digital circuit signal lines and analog circuit signal lines cross or are proximity should be avoided as far as possible. Failure to do so may result in incorrect op of the analog circuitry due to inductance, adversely affecting A/D conversion values. Als circuitry must be isolated from the analog input pins (AN0 to AN15), analog reference vo (AVref), and analog power supply voltage (AVcc) by the analog ground (AVss). Also, th ground (AVss) should be connected at one point to a stable ground (Vss) on the board.

in the A/D converter exceeds the current input via the input impedance (R_{in}), an error wi the analog input pin voltage. Careful consideration is therefore required when deciding t constants.



Notes: Values are reference values.

1.

2. R_{in}: Input impedance

Figure 22.6 Example of Analog Input Protection Circuit

Note: Values are reference values.

Figure 22.7   Analog Input Pin Equivalent Circuit

22.7.7 Module Stop Mode Setting

When this LSI enters software standby mode with A/D conversion enabled, the analog in
retained, and the analog power supply current is equal to the current as during A/D conve
the analog power supply current needs to be reduced in software standby mode, clear the
TRGS1, and TRGS0 bits all to 0 to disable A/D conversion.

switch from continuous scan mode to single mode or one-cycle scan mode)

If any of the above points is applicable, please make settings in accord with the instructi

If 1. is applicable:

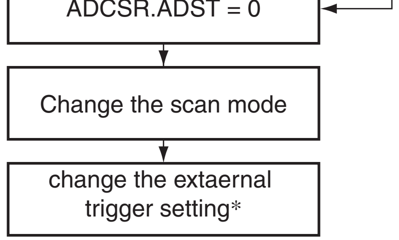
Do not set the ADST bit in ADCSR to 1.

If 2. or 3. is applicable:

Be sure to invalidate the external trigger input before changing the setting from activatio external trigger to disabling of the external trigger or changing the scan-mode setting (cl the setting of the SCANE and ADSTCLR bits) while activation by the external trigger is

Setting the TRGS1 and TRGS0 bits in ADCR according to the procedure overleaf invali external trigger input.

See figure 22.8 for details of the procedure in cases where 2. or 3. is applicable.



Note * Overwrite the TRGS1 and TRGS0 bits settings at the same time (in a byte unit).

Figure 22.8 Procedure for Changing Modes when Starting of the A/D Converter External Trigger has been Selected

- Two flash-memory MATs according to LSI initiation mode.

  The on-chip flash memory has two memory spaces in the same address space (herea referred to as memory MATs). The mode setting at initiation determines which mem is initiated first.

  The MAT can be switched by using the bank-switching method after initiation.

  — The user memory MAT is initiated at a power-on reset in user mode: 160K bytes

  — The user boot memory MAT is initiated at a power-on reset in user boot mode: 8

- Programming/erasing interface by the download of on-chip program

  This LSI has a programming/erasing program. After downloading this program to th RAM, programming/erasing can be performed by setting the parameters.

- Programming/erasing time

  Programming time: 1 ms (typ) for 128-byte simultaneous programming, 7.8 µs per b

  Erasing time: 600 ms (typ) per 1 block (64 kbytes)

- Number of programming

  The number of programming can be up to 100 times at the minimum. (1 to 100 times guaranteed.)

- Three on-board programming modes

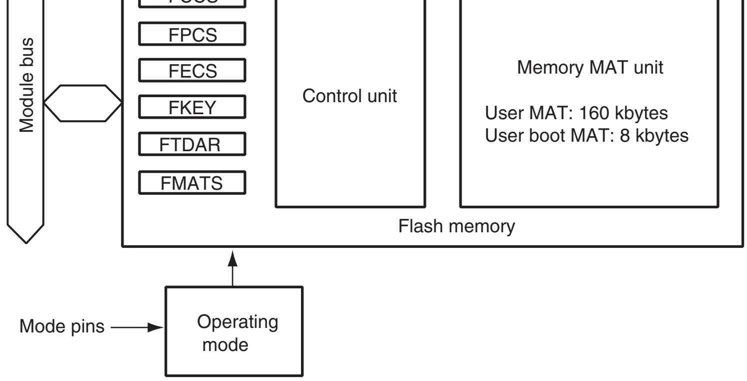
  Boot mode: Using the on-chip SCI-1, the user MAT can be programmed/erased. In b the bit rate between the host and this LSI can be adjusted automatically.

  User program mode: Using a desired interface, the user MAT can be programmed/er

  User boot mode: The User boot program of The optional interface can be made and MAT can be programmed.

- Off-board programming mode

  Programmer mode: Using a PROM programmer, the user MAT can be programmed/



[Legend]

FCCS:	Flash code control/status register
FPCS:	Flash program code select register
FECS:	Flash erase code select register
FKEY:	Flash key code register
FMATS:	Flash MAT select register
RAMER:	Flash transfer destination address register

Note: To read/write the registers above, the FLSHE bit in the serial timer control register (STCR) must be set to 1.

Figure 24.1 Block Diagram of Flash Memory

24.2 Mode Transition Diagram

When the mode pins are set in the reset state and reset start is performed, this LSI enters operating mode as shown in figure 24.2. Although the flash memory can be read in user cannot be programmed or erased. The flash memory can be programmed or erased in boo user program mode, user boot mode, and programmer mode. The differences between bo user program mode, user boot mode, and programmer mode are shown in table 24.1.

Figure 24.2 Mode Transition of Flash Memory

Table 24.1 Differences between Boot Mode, User Program Mode, and Programm

Item	Boot Mode	User Program Mode	User Boot Mode	Progr Mode
Programming/ erasing environment	On-board programming	On-board programming	On-board programming	PROM progra
Programming/ erasing enable MAT	• User MAT • User boot MAT	• User MAT	• User MAT	• Us • Us M
All erasure	O (Automatic)	O	O	O (Au
Block division erasure	O*[1]	O	O	×
Program data transfer	From host via SCI	Via any device	Via any device	Via progra
Reset initiation MAT	Embedded program storage area	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode and reset	Changing FLSHE bit setting	Changing mode and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

2. First, the reset vector is fetched from the embedded program storage MAT. A flash memory related registers are checked, the reset vector is fetched from t boot MAT.

24.3 Flash Memory MAT Configuration

This LSI's flash memory is configured by the 160-Kbyte user MAT and 8-Kbyte user boo The start address is allocated to the same address in the user MAT and user boot MAT. T when program execution or data access is performed between two MATs, the MAT must switched by using FMATS.

The user MAT or user boot MAT can be read in all modes. However, the user boot MAT programmed only in boot mode and programmer mode.

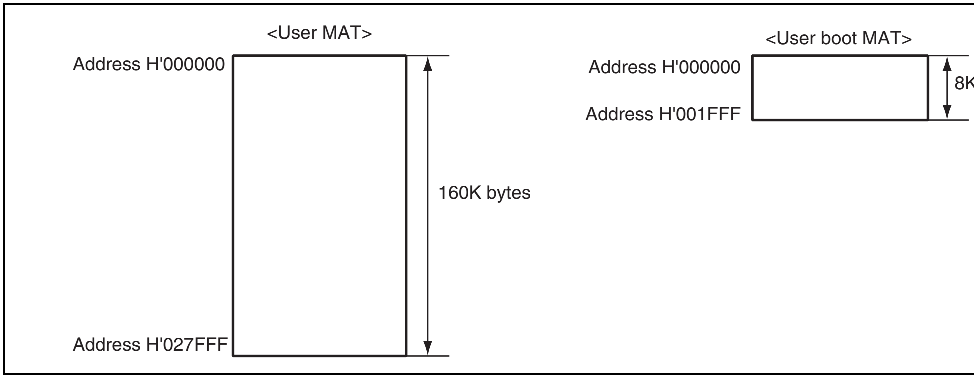


Figure 24.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address that ex size of the 8-Kbyte user boot MAT should not be accessed. If the attempt is made, data is an undefined value.

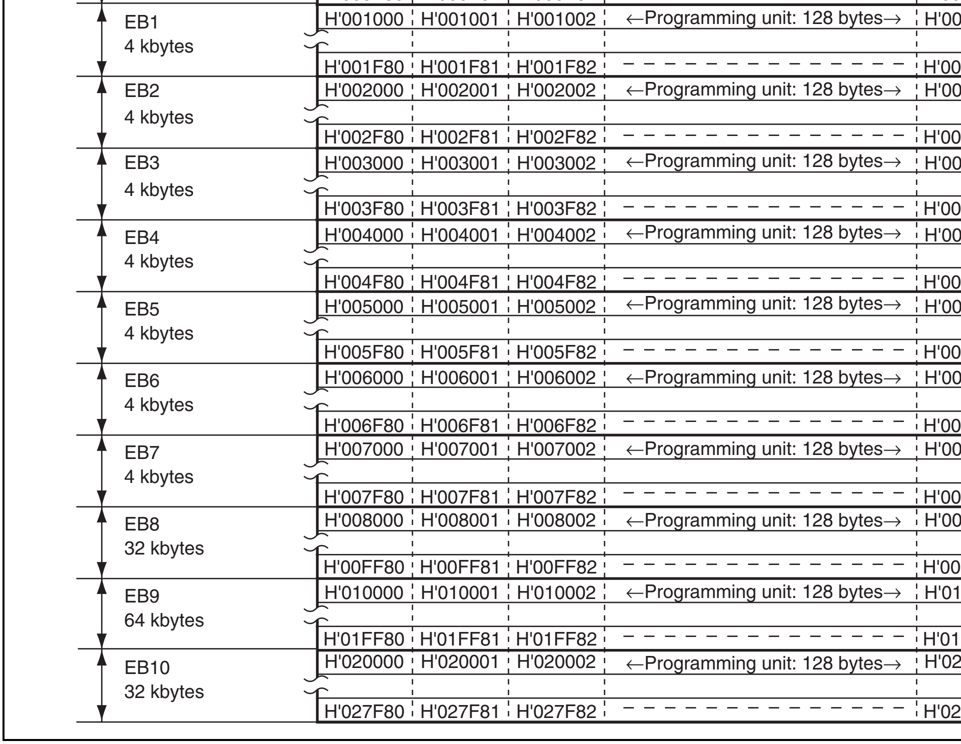


Figure 24.4 Block Structure of the User MAT

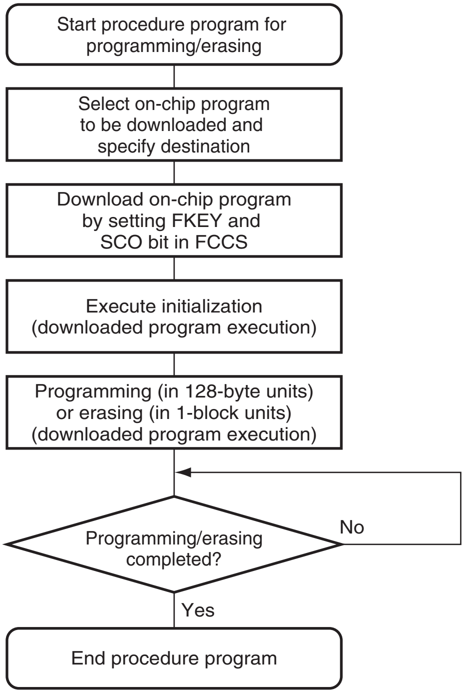

Start procedure program for programming/erasing
Select on-chip program to be downloaded and specify destination
Download on-chip program by setting FKEY and SCO bit in FCCS
Execute initialization (downloaded program execution)
Programming (in 128-byte units) or erasing (in 1-block units) (downloaded program execution)
Programming/erasing completed?
No
Yes
End procedure program

Figure 24.5 Procedure for Creating Procedure Program

with the embedded program storage area during download. Since the memory MAT can
read during programming/erasing, the procedure program must be executed in a space o
the flash memory (for example, on-chip RAM). Since the download result is returned to
programming/erasing interface parameter, whether download is normally executed or no
confirmed.

(3) Initialization of Programming/Erasing

A pulse with the specified period must be applied when programming or erasing. The sp
pulse width is made by the method in which wait loop is configured by the CPU instruct
Accordingly, the operating frequency of the CPU needs to be set before programming/er
operating frequency of the CPU is set by the programming/erasing interface parameter.

(4) Execution of Programming/Erasing

The start address of the programming destination and the program data are specified in 1
units when programming. The block to be erased is specified with the erase block numb
erase-block units when erasing. Specifications of the start address of the programming d
program data, and erase block number are performed by the programming/erasing interf
parameters, and the on-chip program is initiated. The on-chip program is executed by us
JSR or BSR instruction and executing the subroutine call of the specified address in the
RAM. The execution result is returned to the programming/erasing interface parameter.

The area to be programmed must be erased in advance when programming flash memor
interrupts are disabled during programming/erasing.

Table 24.2 Pin Configuration

Abbreviation	I/O	Function
$\overline{\text{RES}}$	Input	Reset
MD2, MD1	Input	Set operating mode of this LSI
TxD1	Output	Serial transmit data output (used in boot r
RxD1	Input	Serial receive data input (used in boot mo

Flash key code register	FKEY	R/W	H'00	H'FEAC
Flash MAT select register	FMATS	R/W	H'00	H'FEAD
Flash transfer destination address register	FTDAR	R/W	H'00	H'FEAE

Note: * Bits other than the SCO bit are read-only bits. The SCO bit is a write-only bit always read as 0.

Table 24.4 Parameter Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	D W
Download path fail result parameter	DPFR	R/W*	Undefined	On-chip RAM*	8
Flash path/fail parameter	FPFR	R/W	Undefined	R0L of CPU	8
Flash program/erase frequency parameter	FPEFEQ	R/W	Undefined	ER0 of CPU	8
Flash multipurpose address area parameter	FMPAR	R/W	Undefined	ER1 of CPU	8
Flash multipurpose data destination parameter	FMPDR	R/W	Undefined	ER0 of CPU	8
Flash erase block select parameter	FEBS	R/W	Undefined	ER0 of CPU	8

Note: * One byte of the start address on the on-chip RAM specified by FTDAR

	FKEY	O	—	O	O	—
	FMATS	—	—	O*[1]	O*[1]	O*
	FTDAR	O	—	—	—	—
Programming/ erasing interface parameters	DPFR	O	—	—	—	—
	FPFR	—	O	O	O	—
	FPEFEQ	—	O	—	—	—
	FMPAR	—	—	O	—	—
	FMPDR	—	—	O	—	—
	FEBS	—	—	—	O	—

Notes: 1. Programming and erasure of the user MAT in user boot mode require settings.

2. A setting may be required depending on the combination of the startup mode a MAT to be read.

Bit	Bit Name	Value	R/W	Description
7	—	1	R	Reserved
6	—	0	R	These are read-only bits and cannot be modifi
5	—	0	R	
4	FLER	0	R	Flash Memory Error Indicates that an error has occurred during pro or erasing the flash memory. When this bit is s the flash memory enters the error protection s When this bit is set to 1, high voltage is applie internal flash memory. To reduce the damage flash memory, the reset must be released afte input period (period of $\overline{\text{RES}}$ = 0) of at least 10 0: Flash memory operates normally (Error pr invalid) [Clearing condition] • At a power-on reset 1: An error occurs during programming/erasi memory (Error protection is valid) [Setting conditions] • When an interrupt, such as NMI, occurs du programming/erasing. • When the flash memory is read during programming/erasing (including a vector r an instruction fetch). • When the SLEEP instruction is executed d programming/erasing (including software s mode).

FKEY, and this operation must be executed in t
chip RAM. Dummy read of FCCS must be exec
twice immediately after setting this bit to 1. All i
must be disabled during download. This bit is c
0 when download is completed.

During program download initiated with this bit,
particular processing which accompanies bank
switching of the program storage area is execu

0: Download of the programming/erasing prog
not requested.

[Clearing condition]

- When download is completed

1: Download of the programming/erasing prog
requested.

[Setting conditions] (When all of the following c
are satisfied)

- H'A5 is written to FKEY
- Setting of this bit is executed in the on-chip

Note: * This is a write-only bit. This bit is always read as 0.

				[Clearing condition]
				When transfer is completed
				1: Programming program is selected.

(3) Flash Erase Code Select Register (FECS)

FECS selects the erasing program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifi
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program to be downloade 0: Erasing program is not selected. [Clearing condition] When transfer is completed 1: Erasing program is selected.

3	K3	0	R/W	on-chip program cannot be downloaded to the RAM. Only when H'5A is written can programming/era the flash memory be executed. When a value o H'5A is written, even if the programming/erasin program is executed, programming/erasing can performed. H'A5: Writing to the SCO bit is enabled. (The S cannot be set to 1 when FKEY is a value than H'A5.) H'5A: Programming/erasing of the flash memor enabled. (When FKEY is a value other th the software protection state is entered.) H'00: Initial value
2	K2	0	R/W	
1	K1	0	R/W	
0	K0	0	R/W	

2	MS2	0	R/W*[2]	User Boot MAT. (The user boot MAT cannot programmed in user program mode even if th boot MAT is selected by FMATS. The user bo must be programmed in boot mode or progra mode.) H'AA: User boot MAT is selected (user MAT i when the value of these bits is other th Initial value when initiated in user boot H'00: Initial value when initiated in a mode ot user boot mode (user MAT is selected) [Programmable condition] Execution state in the on-chip RAM
1	MS1	0/1*[1]	R/W*[2]	
0	MS0	0	R/W*[2]	

Note: *1 The value is 1 in user boot mode and 0 otherwise.

*2 The initial value is 1 in user boot mode and 0 in a mode other than user boot

				set in bits TDA6 to TDA0 is within the range of H'01 when download is executed by setting the in FCCS to 1. Make sure that this bit is cleared before setting the SCO bit to 1 and the value sp by bits TDA6 to TDA0 should be within the rang H'00 to H'01. 0: The value specified by bits TDA6 to TDA0 is the range. 1: The value specified by bits TDA6 to TDA0 is H'02 and H'FF and download has stopped.
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the on-chip RAM start address of the download destination. A value between H'00 ar and up to 3 kbytes can be specified as the start of the on-chip RAM.
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	H'00: H'FFD080 is specified as the star address.
1	TDA1	0	R/W	
0	TDA0	0	R/W	H'01: H'FFD880 is specified as the star address. H'02 to H'7F: Setting prohibited. (Specifying a value from H'02 to the TDER bit to 1 and stops dow the on-chip program.)

is written in R0L. The programming/erasing interface parameters are used in download
initialization before programming or erasing, programming, and erasing. Table 24.6 sho
usable parameters and target modes. The meaning of the bits in the flash pass and fail re
parameter (FPFR) varies in initialization, programming, and erasure.

Table 24.6 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	All
DPFR	O	—	—	—	R/W	Undefined	On
FPFR	—	O	O	O	R/W	Undefined	R0
FPEFEQ	—	O	—	—	R/W	Undefined	ER
FMPAR	—	—	O	—	R/W	Undefined	ER
FMPDR	—	—	O	—	R/W	Undefined	ER
FEBS	—	—	—	O	R/W	Undefined	ER

Note: * A single byte of the start address of the on-chip RAM specified by FTDAR

(a) Download Control

The on-chip program is automatically downloaded by setting the SCO bit in FCCS to 1.
chip RAM area to download the on-chip program is the 4-kbyte area starting from the st
specified by FTDAR. Download is set by the programming/erasing interface registers, a
download pass and fail result parameter (DPFR) indicates the return value.

The start address of the programming destination on the user MAT must be stored in gen
register ER1. This parameter is called the flash multipurpose address area parameter (FM

The program data is always in 128-byte units. When the program data does not satisfy 12
128-byte program data is prepared by filling the dummy code (H'FF). The boundary of th
address of the programming destination on the user MAT is aligned at an address where t
eight bits (A7 to A0) are H'00 or H'80.

The program data for the user MAT must be prepared in consecutive areas. The program
must be in a consecutive space which can be accessed using the MOV.B instruction of th
and is not in the flash memory space.

The start address of the area that stores the data to be written in the user MAT must be se
general register ER0. This parameter is called the flash multipurpose data destination are
parameter (FMPDR).

For details on the programming procedure, see section 24.8.2, User Program Mode.

(d) Erasure

When the flash memory is erased, the erase block number on the user MAT must be pass
erasing program which is downloaded.

The erase block number on the user MAT must be set in general register ER0. This param
called the flash erase block select parameter (FEBS).

One block is selected from the block numbers of 0 to 10 as the erase block number.

For details on the erasing procedure, see section 24.8.2, User Program Mode.

				Only one type can be specified for the on-chip which can be downloaded. When the program downloaded is not selected, more than two typ programs are selected, or a program which is mapped is selected, an error occurs. 0: Download program selection is normal 1: Download program selection is abnormal
1	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) and returns the 0: FKEY setting is normal (H'A5) 1: FKEY setting is abnormal (value other than
0	SF	—	R/W	Success/Fail Returns the download result. Reads back the downloaded to the on-chip RAM and determin whether it has been transferred to the on-chip 0: Download of the program has ended norma error) 1: Download of the program has ended abnor (error occurs)

				These bits return 0.
1	FQ	—	R/W	Frequency Error Detect Compares the specified CPU operating frequer the operating frequencies supported by this LS returns the result. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Returns the initialization result. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occ

				to 1. Whether the error protection state is ente can be confirmed with the FLER bit in FCCS. conditions to enter the error protection state se 24.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming ca performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data cou written because the user MAT was not erased is set to 1, there is a high possibility that the u has been written to partially. In this case, after the error factor, erase the user MAT. Also an a write the user MAT when the FMATS value is the user boot MAT is selected leads to a progr execution error. In that case, both the user MA user boot MAT are not rewritten. Writing to the MAT must be performed in boot mode or prog mode. 0: Programming has ended normally 1: Programming has ended abnormally (progr result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'5A) before progran starts, and returns the result. 0: FKEY setting is normal (H'5A) 1: FKEY setting is abnormal (value other than
3	—	—	—	Unused Returns 0.

				When the following items are specified as the s address of the programming destination, an err occurs. • An area other than flash memory • The specified address is not aligned with th byte boundary (lower eight bits of the addres other than H'00 and H'80) 0: Setting of the start address of the programmi destination is normal 1: Setting of the start address of the programmi destination is abnormal
0	SF	—	R/W	Success/Fail Returns the programming result. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error o

				to 1. Whether the error protection state is ente can be confirmed with the FLER bit in FCCS. F conditions to enter the error protection state se 24.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming can erased (FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect Returns 1 when the user MAT could not be era when the flash memory related register setting partially changed. If this bit is set to 1, there is possibility that the user MAT has been erased In this case, after removing the error factor, era user MAT. Also an attempt to erase the user M the FMATS value is H'AA and the user boot M selected leads to an erasure execution error. In case, both the user MAT and user boot MAT a erased. Erasure of the user boot MAT must be performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally
4	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'5A) before erasure and returns the result. 0: FKEY setting is normal (H'5A) 1: FKEY setting is abnormal (value other than

Indicates the erasure result.

0: Erasure has ended normally (no error)

1: Erasure has ended abnormally (error occurs) setting value must be calculated as follows:

1. Round off the operating frequency express unit at the third decimal place to make it in decimal places.
2. Multiply the rounded number by 100 and c result into binary and write it to FPEFEQ (g register ER0).

For example, when the operating frequency of is 20.000 MHz, the setting value is as follows:

1. Round 20.000 off at the third decimal plac 20.00.
2. Convert 20.00 × 100 = 2000 into a binary r and set B'0000 0111 1101 0000 (H'07D0)

				programming is executed starting from the spe start address of the user MAT. Therefore, the s start address of the programming destination b 128-byte boundary, and MOA6 to MOA0 are al cleared to 0.

(5) Flash Multipurpose Data Destination Parameter (FMPDR: General Register E CPU)

FMPDR stores the start address in the area which stores the data to be programmed in the MAT.

When the storage destination for the program data is in flash memory, an error occurs. Th occurrence is indicated by the WD bit in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	These bits store the start address of the area w stores the program data for the user MAT. Con 128-byte data is programmed to the user MAT from the specified start address.

EBS0	A value of 0 corresponds to block EB0 and 10 corresponds to block EB10. Do not set a value the range from 0 to 10 (from H'00 to H'0A).

Mode Setting	MD2	MD1	NMI
Boot mode	1	0	1
User program mode	0	1	0/1
User boot mode	1	0	0

24.8.1 Boot Mode

Boot mode executes programming/erasing of the user MAT and the user boot MAT by m the control command and program data transmitted from the externally connected host vi chip SCI_1.

In boot mode, the tool for transmitting the control command and program data, and the pr data must be prepared in the host. The serial communication mode is set to asynchronous The system configuration in boot mode is shown in figure 24.6. Interrupts are ignored in mode. Configure the user system so that interrupts do not occur.

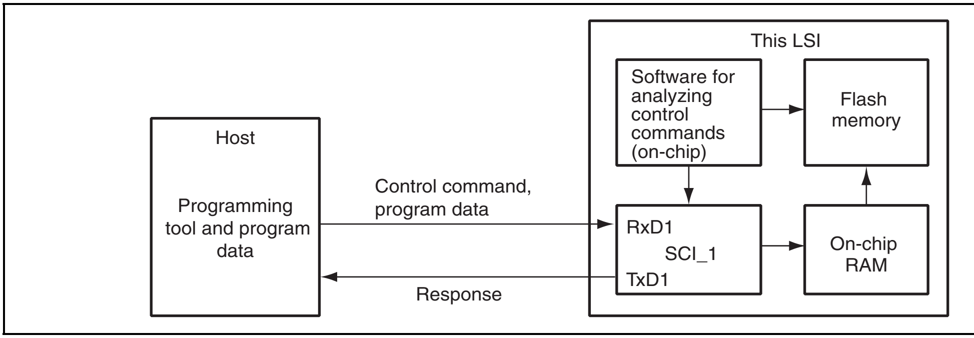

This LSI
Software for analyzing control commands (on-chip)
Flash memory
Host
Programming tool and program data
Control command, program data
RxD1
SCI_1
TxD1
On-chip RAM
Response

Figure 24.6 System Configuration in Boot Mode

adjustment end sign. When the host receives this bit adjustment end sign normally, it tra byte of H'55 to this LSI. When reception is not executed normally, initiate boot mode ag bit rate may not be adjusted within the allowable range depending on the combination o rate of the host and the system clock frequency of this LSI. Therefore, the transfer bit ra host and the system clock frequency of this LSI must be as shown in table 24.8.

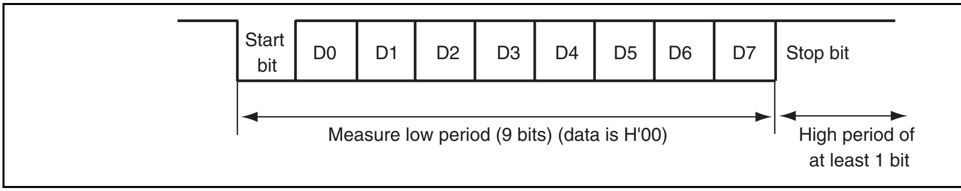


Figure 24.7 Automatic-Bit-Rate Adjustment Operation

Table 24.8 System Clock Frequency for Automatic-Bit-Rate Adjustment

Bit Rate of Host	System Clock Frequency of This LS
9,600 bps	8 to 20 MHz
19,200 bps	8 to 20 MHz

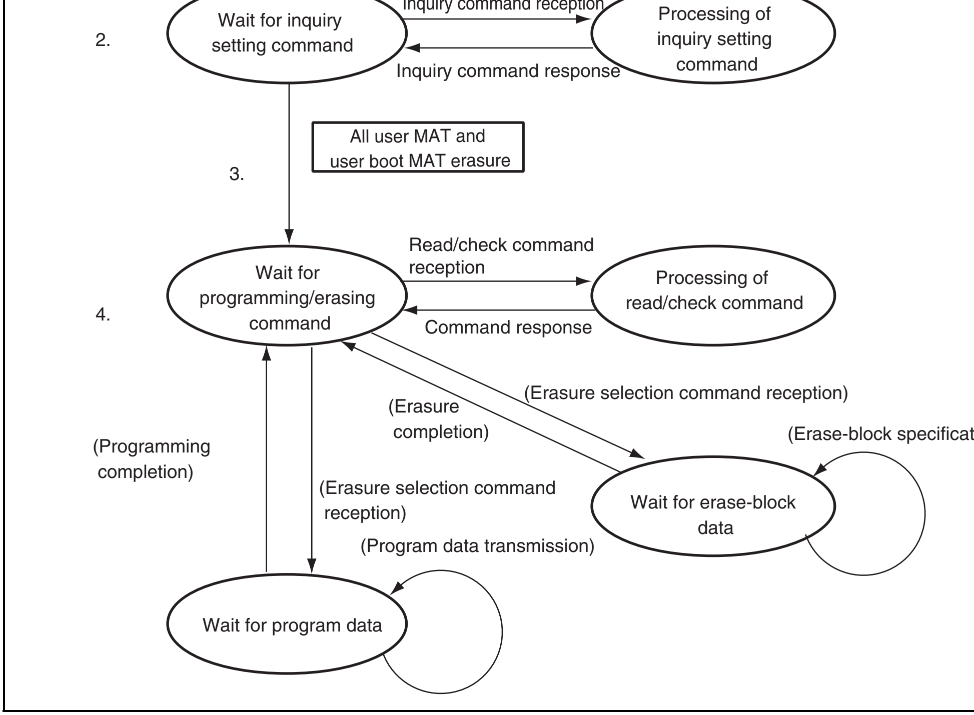
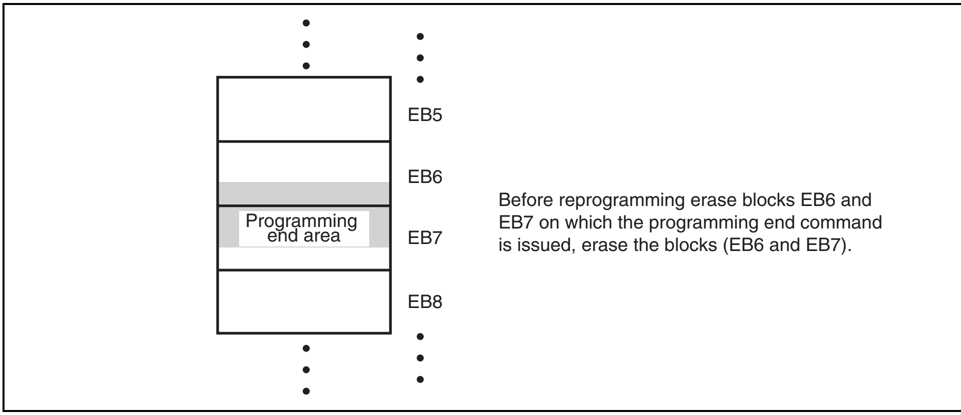

2.
Wait for inquiry setting command
Inquiry command reception
Processing of inquiry setting command
Inquiry command response
All user MAT and user boot MAT erasure
3.
4.
Wait for programming/erasing command
Read/check command reception
Processing of read/check command
Command response
(Erasure selection command reception)
(Erasure completion)
(Erase-block specificat
(Programming completion)
(Erasure selection command reception)
Wait for erase-block data
(Program data transmission)
Wait for program data

Figure 24.8 Boot Mode State Transition Diagram

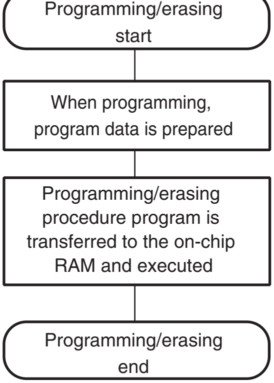
which the programming end command is issued, erase the erase block. An example o
erase block is shown in figure 24.9. When the erasure preparation notice is received,
of waiting for erase block data is entered. The erase block number must be transmitte
erasing command is transmitted. When the erasure is finished, the erase block numbe
set to H'FF and transmitted. Then the state of waiting for erase block data is returned
state of waiting for programming/erasing command. Erasure must be executed when
specified block is programmed without a reset start after programming is executed in
mode. When programming can be executed by only one operation, all blocks are era
entering the state of waiting for programming/erasing command or another command
this case, the erasing operation is not required. The commands other than the programming/erasing command perform sum check, blank check (erasure check), an
read of the user MAT and user boot MAT, and acquisition of current status informat

Memory read of the user MAT and user boot MAT can only read the data programmed a
user MAT and user boot MAT have automatically been erased. No other data can be rea



Before reprogramming erase blocks EB6 and EB7 on which the programming end command is issued, erase the blocks (EB6 and EB7).

Figure 24.9 Example of Erase Block Including Programmed Area


Programming/erasing start
When programming, program data is prepared
Programming/erasing procedure program is transferred to the on-chip RAM and executed
Programming/erasing end

1. Programming/erasing is executed only in the on-chip RAM.

2. After programming/erasing is finished, protect the flash memory by the hardware protection.

3. Make sure that the program data do not overlap with the download destination specified by FTDAR.

Figure 24.10 Programming/Erasing Flow

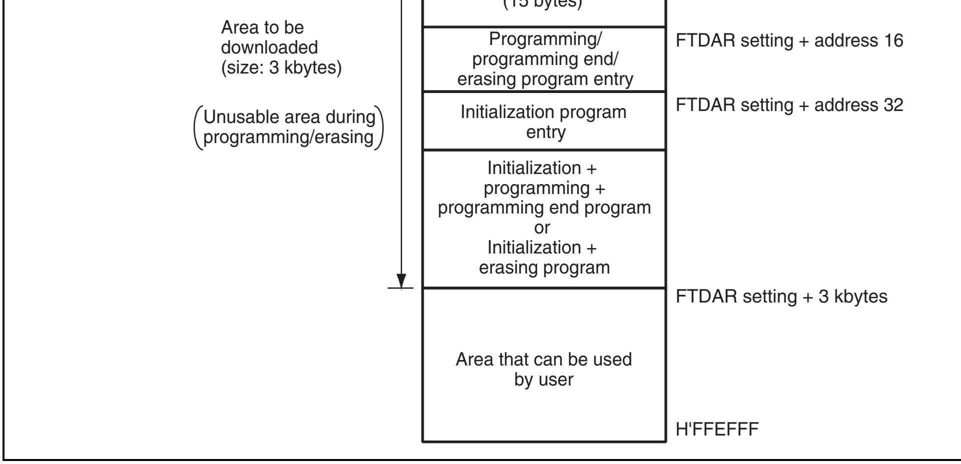
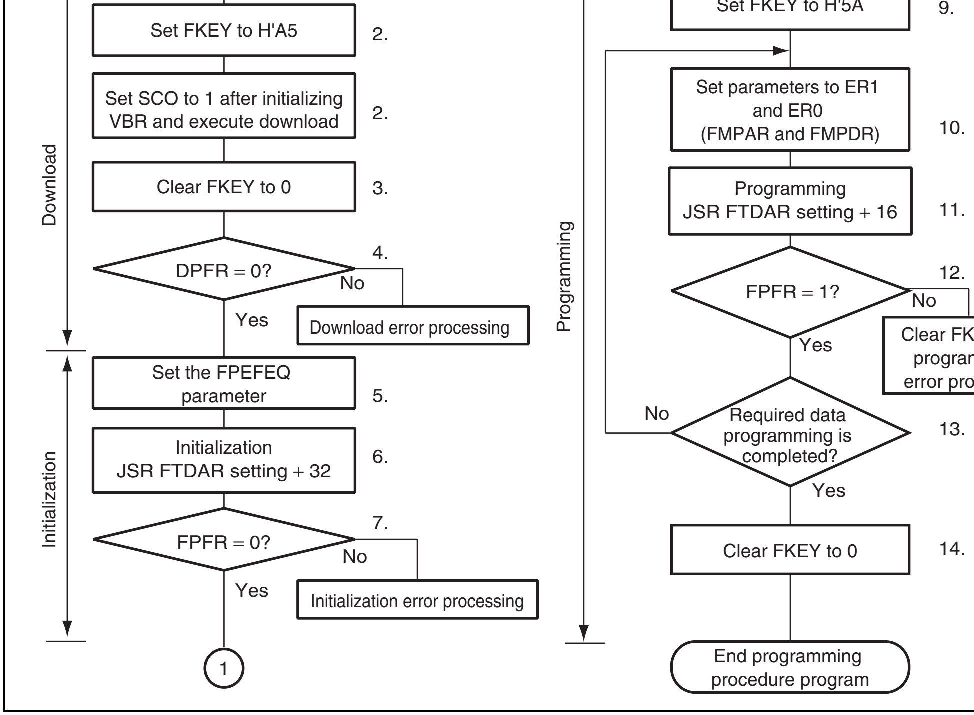

(15 bytes)
Area to be downloaded (size: 3 kbytes)
(Unusable area during programming/erasing)
Programming/ programming end/ erasing program entry
FTDAR setting + address 16
Initialization program entry
FTDAR setting + address 32
Initialization + programming + programming end program or Initialization + erasing program
FTDAR setting + 3 kbytes
Area that can be used by user
H'FFEFFF

Figure 24.11 RAM Map when Programming/Erasing is Executed



Download
Set FKEY to H'A5
2.
Set SCO to 1 after initializing VBR and execute download
2.
Clear FKEY to 0
3.
DPFR = 0?
4.
No
Download error processing
Yes
Initialization
Set the FPEFEQ parameter
5.
Initialization JSR FTDAR setting + 32
6.
FPFR = 0?
7.
No
Initialization error processing
Yes
1
Programming
Set FKEY to H'5A
9.
Set parameters to ER1 and ER0 (FMPAR and FMPDR)
10.
Programming JSR FTDAR setting + 16
11.
FPFR = 1?
12.
No
Clear FK progra error pro
Yes
No
Required data programming is completed?
13.
Yes
Clear FKEY to 0
14.
End programming procedure program

Figure 24.12 Programming Procedure in User Program Mode

H'FF, the program processing time can be shortened.

1. Select the on-chip program to be downloaded and the download destination. When th
   bit in FPCS is set to 1, the programming program is selected. Several programming/
   programs cannot be selected at one time. If several programs are selected, a downloa
   returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the
   destination is specified by FTDAR.
2. Write H'A5 in FKEY. If H'A5 is not written to FKEY, the SCO bit in FCCS cannot b
   to request download of the on-chip program.
   - H'A5 is written to FKEY.
   - Setting the SCO bit is executed in the on-chip RAM.

   When the SCO bit is set to 1, download is started automatically. Since the SCO bit is
   to 0 when the procedure program is resumed, the SCO bit cannot be confirmed to be
   procedure program. The download result can be confirmed by the return value of the
   parameter. To prevent incorrect decision, before setting the SCO bit to 1, set one byt
   on-chip RAM start address specified by FTDAR, which becomes the DPFR paramet
   value other than the return value (e.g. H'FF). Particular processing that is accompani
   switching as described below is performed when download is executed. Dummy rea
   must be performed twice immediately after the SCO bit is set to 1.
   - The user-MAT space is switched to the on-chip program storage area.
   - After the program to be downloaded and the on-chip RAM start address specifie
     FTDAR are checked, they are transferred to the on-chip RAM.
   - FPCS, FECS, and the SCO bit in FCCS are cleared to 0.
   - The return value is set in the DPFR parameter.
   - The values of general registers of the CPU are held.
   - During download, no interrupts can be accepted. However, since the interrupt re
     held, when the procedure program is resumed, the interrupts are requested.

— If the value of the DPFR parameter is the same as that before downloading, the se
the start address of the download destination in FTDAR may be abnormal. In this
confirm the setting of the TDER bit in FTDAR.

— If the value of the DPFR parameter is different from that before downloading, che
bit or FK bit in the DPFR parameter to confirm the download program selection a
setting, respectively.

5. The operating frequency of the CPU is set in the FPEFEQ parameter for initialization
settable operating frequency of the FPEFEQ parameter ranges from 8 to 32 MHz. Wh
frequency is set otherwise, an error is returned to the FPFR parameter of the initializa
program and initialization is not performed. For details on setting the frequency, see s
24.7.2 (3), Flash Program/Erase Frequency Parameter (FPEFEQ: General Register ER
CPU).

6. Initialization is executed. The initialization program is downloaded together with the
programming program to the on-chip RAM. The entry point of the initialization progr
the address which is 32 bytes after #DLTOP (start address of the download destinatio
specified by FTDAR). Call the subroutine to execute initialization by using the follow
steps.

```
MOV.L #DLTOP+32,ER2      ; Set entry address to ER2
JSR   @ER2               ; Call initialization routine
NOP
```

— The general registers other than R0L are held in the initialization program.

— R0L is a return value of the FPFR parameter.

— Since the stack area is used in the initialization program, a stack area of 128 bytes
maximum must be allocated in RAM.

— Interrupts can be accepted during execution of the initialization program. Make su
program storage area and stack area in the on-chip RAM and register values are no
overwritten.

9. FKEY must be set to H'5A and the user MAT must be prepared for programming.

10. The parameters required for programming are set. The start address of the programm destination on the user MAT (FMPAR parameter) is set in general register ER1. The address of the program data storage area (FMPDR parameter) is set in general regist

    — Example of FMPAR parameter setting: When an address other than one in the us area is specified for the start address of the programming destination, even if the programming program is executed, programming is not executed and an error is the FPFR parameter. Since the program data for one programming operation is 1 the lower eight bits of the address must be H'00 or H'80 to be aligned with the 12 boundary.

    — Example of FMPDR parameter setting: When the storage destination for the prog is flash memory, even if the programming routine is executed, programming is n executed and an error is returned to the FPFR parameter. In this case, the prograr must be transferred to the on-chip RAM and then programming must be executed

11. Programming is executed. The entry point of the programming program is at the add is 16 bytes after #DLTOP (start address of the download destination specified by FT Call the subroutine to execute programming by using the following steps.

```
MOV.L    #DLTOP+16,ER2   ; Set entry address to ER2
JSR      @ER2            ; Call programming routine
NOP
```

    — The general registers other than R0L are held in the programming program.

    — R0L is a return value of the FPFR parameter.

    — Since the stack area is used in the programming program, a stack area of 128 byt maximum must be allocated in RAM.

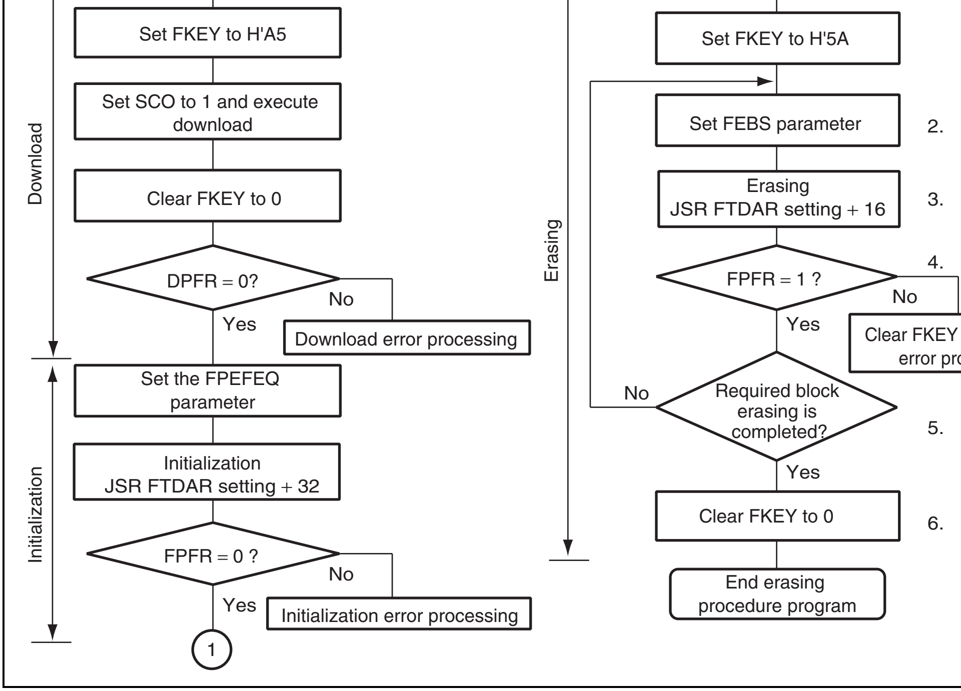

Set FKEY to H'A5
Set SCO to 1 and execute download
Clear FKEY to 0
DPFR = 0?
No
Yes
Download error processing
Download
Set the FPEFEQ parameter
Initialization JSR FTDAR setting + 32
FPFR = 0 ?
No
Yes
Initialization error processing
Initialization
1
Set FKEY to H'5A
Set FEBS parameter
2.
Erasing JSR FTDAR setting + 16
3.
FPFR = 1 ?
4.
No
Yes
Clear FKEY
error pro
Erasing
No
Required block erasing is completed?
5.
Yes
Clear FKEY to 0
6.
End erasing procedure program

Figure 24.13   Erasing Procedure in User Program Mode

The procedure program must be executed in an area other than the user MAT to be erase
the SCO bit in FCCS to 1 to request download must be executed in the on-chip RAM. T
that can be executed in the steps of the procedure program (on-chip RAM and user MAT
shown in section 24.8.4, Storable Areas for On-Chip Program and Program Data. For th
downloaded on-chip program area, see figure 24.11.

2. Set the FEBS parameter necessary for erasure. Set the erase block number (FEBS para of the user MAT in general register ER0. If a value other than an erase block number user MAT is set, no block is erased even though the erasing program is executed, and is returned to the FPFR parameter.

3. Erasure is executed. Similar to as in programming, the entry point of the erasing prog the address which is 16 bytes after #DLTOP (start address of the download destinatio specified by FTDAR). Call the subroutine to execute erasure by using the following s

```
MOV.L #DLTOP+16, ER2      ; Set entry address to ER2
JSR   @ER2                ; Call erasing routine
NOP
```

   - The general registers other than R0L are held in the erasing program.
   - R0L is a return value of the FPFR parameter.
   - Since the stack area is used in the erasing program, a stack area of 128 bytes at the maximum must be allocated in RAM.

4. The return value in the erasing program, the FPFR parameter is determined.

5. Determine whether erasure of the necessary blocks has finished. If more than one bloc be erased, update the FEBS parameter and repeat steps 2 to 5.

6. After erasure completes, clear FKEY and specify software protection. If this LSI is re a reset immediately after erasure has finished, secure the reset input period (period of of at least 100 μs.

For the mode pin settings to start up user boot mode, see table 24.7.

When the reset start is executed in user boot mode, the built-in check routine runs. The u and user boot MAT states are checked by this check routine.

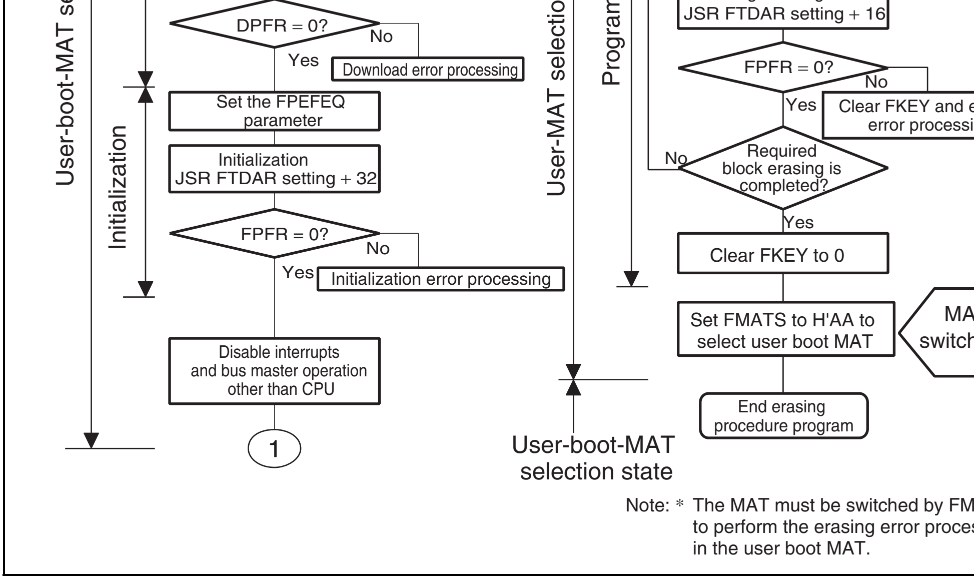
While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user bo At this point, H'AA is set to FMATS because the execution target MAT is the user boot

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by settin is required: switching from user-boot-MAT selection state to user-MAT selection state, switching back to user-boot-MAT selection state after programming completes.

Figure 24.14 shows the procedure for programming the user MAT in user boot mode.


User-boot-MAT se
DPFR = 0?
No
Yes
Download error processing
Initialization
Set the FPEFEQ parameter
Initialization JSR FTDAR setting + 32
FPFR = 0?
No
Yes
Initialization error processing
Disable interrupts and bus master operation other than CPU
1
User-MAT selectio
Program
JSR FTDAR setting + 16
FPFR = 0?
No
Yes
Clear FKEY and e error processi
No
Required block erasing is completed?
Yes
Clear FKEY to 0
Set FMATS to H'AA to select user boot MAT
MA switch
End erasing procedure program
User-boot-MAT selection state

Note: * The MAT must be switched by FM to perform the erasing error proces in the user boot MAT.

Figure 24.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boo whether the MAT is switched or not as shown in figure 24.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the us hidden in the background. The user MAT and user boot MAT are switched only while th MAT is being programmed. Because the user boot MAT is hidden while the user MAT is programmed, the procedure program must be executed in an area other than flash memor the programming procedure completes, switch the MATs again to return to the first state.

(3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processing made by setting FM
required: switching from user-boot-MAT selection state to user-MAT selection state, an
switching back to user-boot-MAT selection state after erasing completes.

Figure 24.15 shows the procedure for erasing the user MAT in user boot mode.

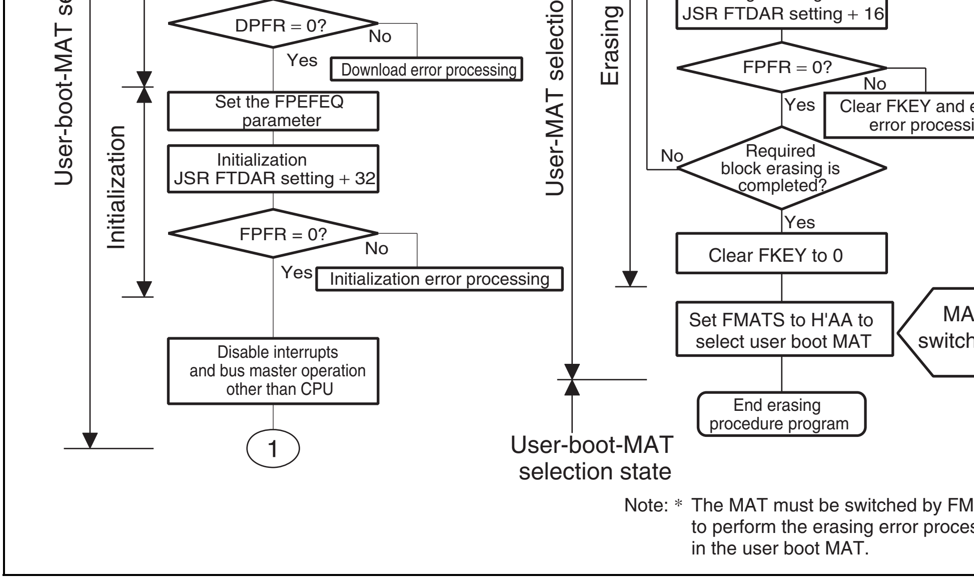

User-boot-MAT se
DPFR = 0?
No
Yes
Download error processing
Initialization
Set the FPEFEQ parameter
Initialization
JSR FTDAR setting + 32
FPFR = 0?
No
Yes
Initialization error processing
Disable interrupts and bus master operation other than CPU
1
User-MAT selectio
Erasing
JSR FTDAR setting + 16
FPFR = 0?
No
Yes
Clear FKEY and e error processi
No
Required block erasing is completed?
Yes
Clear FKEY to 0
Set FMATS to H'AA to select user boot MAT
MA switch
End erasing procedure program
User-boot-MAT selection state
Note: * The MAT must be switched by FM to perform the erasing error proces in the user boot MAT.

Figure 24.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode on whether the MAT is switched or not as shown in figure 24.15.

MAT switching is enabled by writing a specific value to FMATS. Note, however, that wh MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allo MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt v read is undetermined. Perform MAT switching in accordance with the description in secti 24.10, Switching between User MAT and User Boot MAT.

- The on-chip program is downloaded to and executed in the on-chip RAM specified b FTDAR. Therefore, this on-chip RAM area is not available for use.
- Since the on-chip program uses a stack area, allocate 128 bytes at the maximum as a area.
- Download requested by setting the SCO bit in FCCS to 1 should be executed from th RAM because it will require switching of the memory MATs.
- In an operating mode in which the external address space is not accessible, such as s mode, the required procedure programs should be transferred to the on-chip RAM be programming/erasing starts (download result is determined).
- The flash memory is not accessible during programming/erasing. Programming/erasi executed by the program downloaded to the on-chip RAM. Therefore, the procedure that initiates operation should be stored in the on-chip RAM other than the flash mer
- After programming/erasing starts, access to the flash memory should be inhibited un is cleared. The reset input state (period of $\overline{\text{RES}}$ = 0) must be set to at least 100 μs wh operating mode is changed and the reset start executed on completion of programmin Transitions to the reset state are inhibited during programming/erasing. When the res is input, a reset input state (period of $\overline{\text{RES}}$ = 0) of at least 100 μs is needed before the signal is released.
- Switching of the MATs by FMATS should be required when programming/erasong MAT is operated in user boot mode. The program that switches the MATs should be from the on-chip RAM. (For details, see section 24.10, Switching between User MA User Boot MAT.) Make sure you know which MAT is currently selected when witcl
- When the program data storage area is within the flash memory area, an error will oc when the data stored is normal program data. Therefore, the data should be transferre on-chip RAM to place the address that the FMPDR parameter indicates in an area ot the flash memory.

Table 24.10 Usable Area for Programming in User Program Mode

	Storable/Executable Area		Selected MAT	
Item	**On-Chip RAM**	**User MAT**	**User MAT**	**Embedded Storage MA**
Storage area for program data	O	×*	—	—
Operation for selecting on-chip program to be downloaded	O	O	O	
Operation for writing H'A5 to FKEY	O	O	O	
Execution of writing 1 to SCO bit in FCCS (download)	O	×		O
Operation for clearing FKEY	O	O	O	
Decision of download result	O	O	O	
Operation for download error	O	O	O	
Operation for setting initialization parameter	O	O	O	
Execution of initialization	O	×	O	
Decision of initialization result	O	O	O	
Operation for initialization error	O	O	O	
Operation for disabling interrupts	O	O	O	
Operation for writing H'5A to FKEY	O	O	O	
Operation for setting programming parameter	O	×	O	
Execution of programming	O	×	O	
Decision of programming result	O	×	O	
Operation for programming error	O	×	O	
Operation for clearing FKEY	O	×	O	

Note: * Transferring the program data to the on-chip RAM beforehand enables this are used.

Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting erasure parameter	O	×	O
Execution of erasure	O	×	O
Decision of erasure result	O	×	O
Operation for erasure error	O	×	O
Operation for clearing FKEY	O	×	O

Writing 1 to SCO in FCCS (download)	O	×			O
FKEY clearing	O	O		O	
Determination of download result	O	O		O	
Download error processing	O	O		O	
Setting initialization parameter	O	O		O	
Initialization	O	×		O	
Determination of initialization result	O	O		O	
Initialization error processing	O	O		O	
Disabling interrupts	O	O		O	
Switching MATs by FMATS	O	×	O		
Writing H'5A to FKEY	O	×	O		
Setting programming parameter	O	×	O		
Programming	O	×	O		
Determination of programming result	O	×	O		
Programming error processing	O	×*2	O		
FKEY clearing	O	×	O		
Switching MATs by FMATS	O	×		O	

Notes: 1. Transferring the data to the on-chip RAM in advance enables this area to be us

2. Switching FMATS by a program in the on-chip RAM enables this area to be us

FKEY clearing	O	O		O
Determination of download result	O	O		O
Download error processing	O	O		O
Setting initialization parameter	O	O		O
Initialization	O	×		O
Determination of initialization result	O	O		O
Initialization error processing	O	O		O
Disabling interrupts	O	O		O
Switching MATs by FMATS	O	×		O
Writing H'5A to FKEY	O	×	O	
Setting erasure parameter	O	×	O	
Erasure	O	×	O	
Determination of erasure result	O	×	O	
Erasing error processing	O	×*	O	
FKEY clearing	O	×	O	
Switching MATs by FMATS	O	×	O	

Note: * Switching FMATS by a program in the on-chip RAM enables this area to be use program is initiated, and the error in programming/erasing is indicated by the FPFR para

Table 24.14 Hardware Protection

		Function to be Pro	
Item	**Description**	**Download**	**Progra Erasin**
Reset protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and the programming/erasing protection state is entered. • The reset state will not be entered by a reset using the $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has settled after a power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width given in the AC characteristics. If a reset is input during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	O	O

by SCO bit	entered when the SCO bit in FCCS is cleared to 0 to disable download of the programming/erasing programs.		
Protection by FKEY	The programming/erasing protection state is entered because download and programming/erasing are disabled unless the required key code is written in FKEY.	O	O

24.9.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when a CPU runa occurs or operations not according to the programming/erasing procedures are detected programming/erasing of the flash memory. Aborting programming or erasure in such ca prevents damage to the flash memory due to excessive programming or erasing.

If an error occurs during programming/erasing of the flash memory, the FLER bit in FC to 1 and the error protection state is entered.

- When an interrupt request, such as NMI, occurs during programming/erasing.
- When the flash memory is read from during programming/erasing (including a vecto an instruction fetch).
- When a SLEEP instruction is executed (including software-standby mode) during programming/erasing.

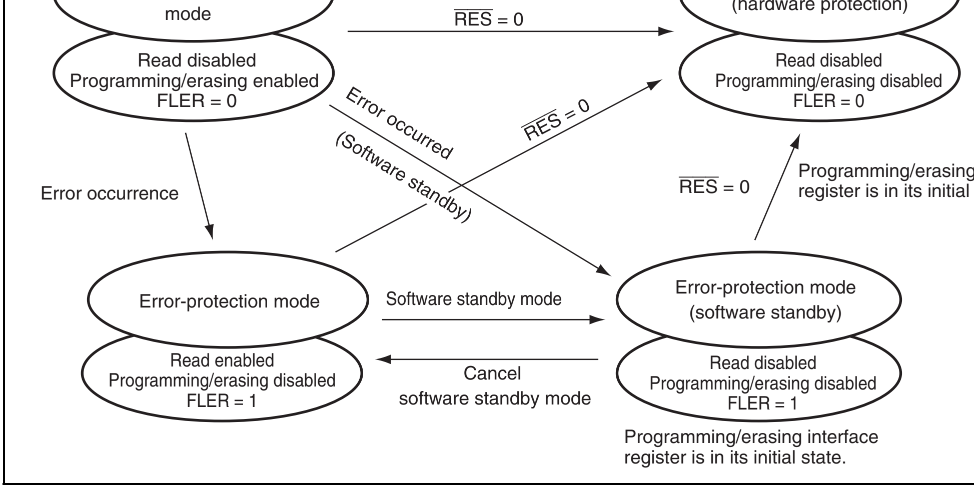


Figure 24.16 Transitions to Error Protection State

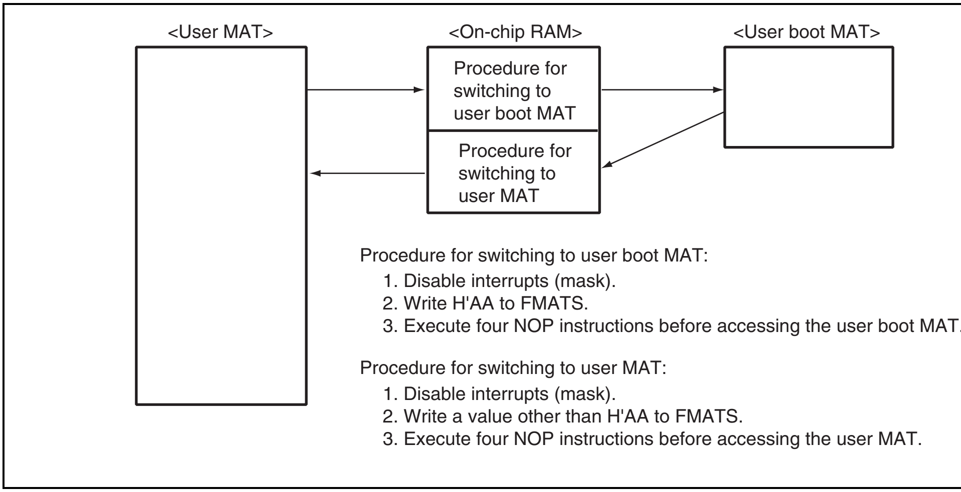
prevents access to the flash memory during MAT switching).

3. If an interrupt has occurred during switching, there is no guarantee of which memory being accessed.

   Always mask the maskable interrupts before switching between MATs. In addition, the system so that NMI interrupts do not occur during MAT switching.

4. After the MATs have been switched, take care because the interrupt vector table wil been switched.

5. Memory sizes of the user MAT and user boot MAT are different. Do not access a us MAT in a space of 8 kbytes or more. If access goes beyond the 8-kbyte space, the va are undefined.


<User MAT>
<On-chip RAM>
<User boot MAT>
Procedure for switching to user boot MAT
Procedure for switching to user MAT

Procedure for switching to user boot MAT:
1. Disable interrupts (mask).
2. Write H'AA to FMATS.
3. Execute four NOP instructions before accessing the user boot MAT.

Procedure for switching to user MAT:
1. Disable interrupts (mask).
2. Write a value other than H'AA to FMATS.
3. Execute four NOP instructions before accessing the user MAT.

Figure 24.17 Switching between User MAT and User Boot MAT

User boot MAT	8 kbytes	FZTATUSBTV3A

Note: * For the R4F2117R model, 160 kbytes of ROM space is available when the use selected. If programming is performed in programmer mode, H'FF data must b to address H'28000 to H'3FFFF with 256-kbyte capacity setting.

24.12 Standard Serial Communication Interface Specifications for B Mode

The boot program initiated in boot mode performs serial communication using the host ar chip SCI_1. The serial communication interface specifications are shown below.

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to achieve serial communication wit host. Initiating boot mode enables starting of the boot program and entry to the bit-rat adjustment state. The program receives the command from the host to adjust the bit ra adjusting the bit rate, the program enters the inquiry/selection state.

2. Inquiry/selection state

   In this state, the boot program responds to inquiry commands from the host. The devi clock mode, and bit rate are selected. After selection of these settings, the program is enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure to chip RAM and erases the user MATs and user boot MATs before the transition.

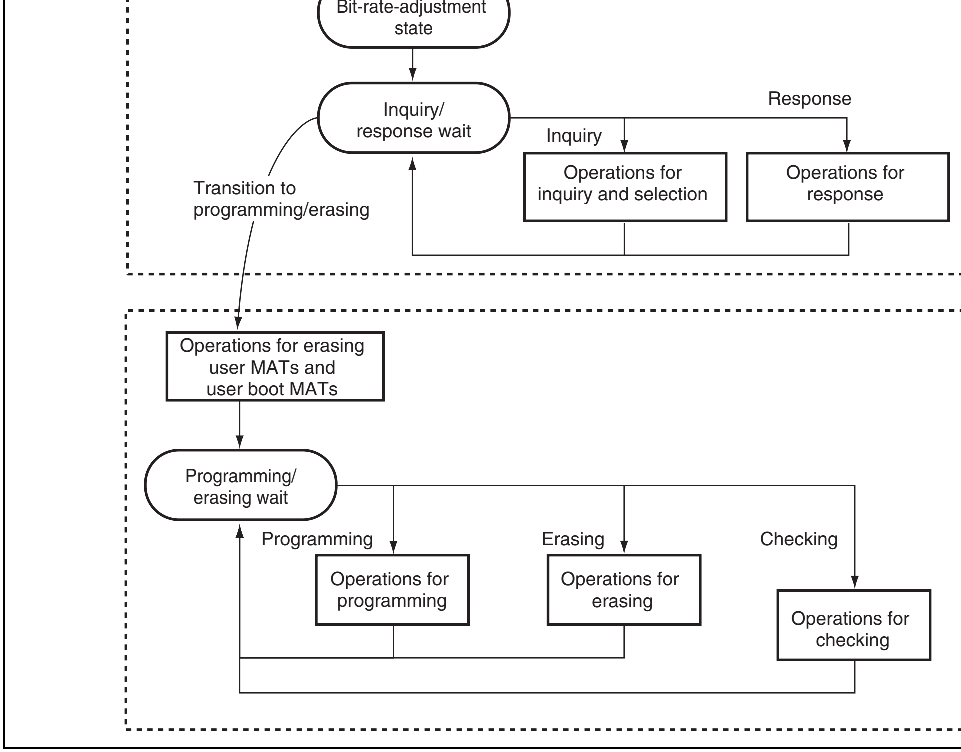

Bit-rate-adjustment state
Inquiry/ response wait
Response
Inquiry
Operations for inquiry and selection
Operations for response
Transition to programming/erasing
Operations for erasing user MATs and user boot MATs
Programming/ erasing wait
Programming
Erasing
Checking
Operations for programming
Operations for erasing
Operations for checking

Figure 24.18 Boot Program States

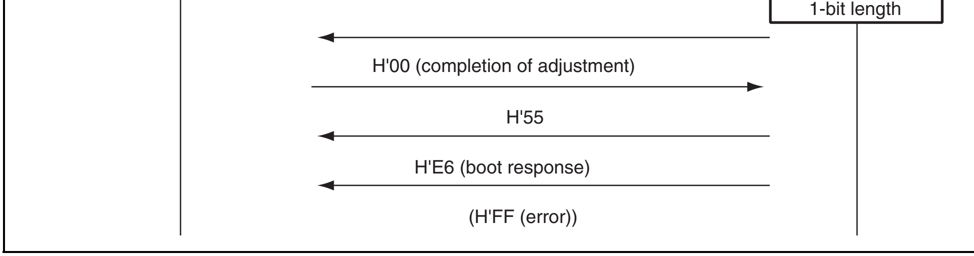


Figure 24.19 Bit-Rate-Adjustment Sequence

(2) Communications Protocol

After adjustment of the bit rate, the protocol for serial communications between the host a boot program is as shown below.

1. One-byte commands and one-byte responses

   These one-byte commands and one-byte responses consist of the inquiries and the AC successful completion.

2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selections responses to inquiries.

   The program data size is not included under this heading because it is determined in a command.

3. Error response

   The error response is a response to inquiries. It consists of an error response and an er and comes two bytes.

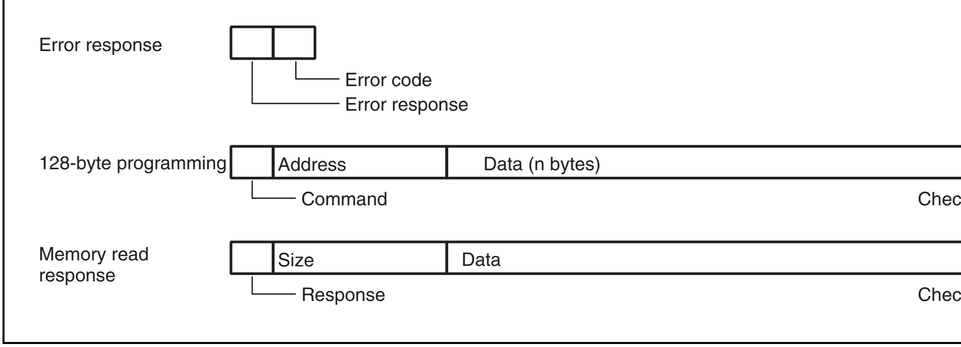


Figure 24.20 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasi checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amou and checksum
- Checksum (one byte): The checksum is calculated so that the total of all values from command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

H'10	Device selection	Selection of device code
H'21	Clock mode inquiry	Inquiry regarding numbers of clock modes values of each mode
H'11	Clock mode selection	Indication of the selected clock mode
H'22	Division ratio inquiry	Inquiry regarding the number of frequency-clock types, the number of division ratios a values of each division
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and minimu values of the main clock and peripheral clo
H'24	User boot MAT information inquiry	Inquiry regarding the a number of user boo and the start and last addresses of each M
H'25	User MAT information inquiry	Inquiry regarding the a number of user MA the start and last addresses of each MAT
H'26	Block for erasing information Inquiry	Inquiry regarding the number of blocks and and last addresses of each block
H'27	Programming unit inquiry	Inquiry regarding the unit of program data
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MATs or user boot MATs, a to programming/erasing state
H'4F	Boot program status inquiry	Inquiry into the operated status of the boot

response to the supported device inquiry.

Command	H'20

- Command, H'20, (one byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
…			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, an checksum, that is, the amount of data contributes by the number of devices, characte codes and product names
- Number of devices (one byte): The number of device types supported by the boot pr
- Number of characters (one byte): The number of characters in the device codes and program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one byte): Checksum

  The checksum is calculated so that the total number of all values from the command the SUM byte becomes H'00.

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response	H'90	ERROR

- Error response, H'90, (one byte): Error response to the device selection command

  ERROR : (one byte): Error code
  H'11: Sum check error
  H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode in

Command	H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	…	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the number of modes and modes
- Number of clock modes (one byte): The number of supported clock modes
  H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode 1
- SUM (one byte): Checksum

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91, (one byte): Error response to the clock mode selection comma
- ERROR : (one byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock n be selected using these respective values.

…						
SUM						

- Response, H'32, (one byte): Response to the division ratio inquiry
- Size (one byte): The total amount of data that represents the number of types, the num division ratios, and the division ratios
- Number of types (one byte): The number of supported divided clock types (e.g. when there are two divided clock types, which are the main and peripheral clock number of types will be H'02.)
- Number of division ratios (one byte): The number of division ratios for each type (e.g. the number of division ratios to which the main clock can be set and the peripher can be set.)
- Division ratio (one byte)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

  The number of division ratios returned is the same as the number of division ratios an many groups of data are returned as there are types.
- SUM (one byte): Checksum

operating clock frequency	frequency)
…	
SUM	

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operati frequency types

  (e.g. when there are two operating clock frequency types, which are the main and pe clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of th clock frequency.

  The minimum and maximum values of the operating clock frequency represent the v MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the v 20.00 MHz, it will be 2000, which is H'07D0.)
- Maximum value (two bytes): Maximum value among the divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clo frequencies.
- SUM (one byte): Checksum

- Response, H'34, (one byte): Response to the user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start ac and area-last address
- Number of areas (one byte): The number of consecutive user boot MAT areas
  When the user boot MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are
- SUM (one byte): Checksum

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start a and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there ar
- SUM (one byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command	H'26

- Command, H'26, (two bytes): Inquiry regarding erased block information

Response	H'36	Size	Number of blocks	
	Block start address			Block last address
	…			
	SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-s addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block

Response	H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fix
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

(k) New Bit-Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock mode selection command.

Command	H'3F	Size	Bit rate	Input frequency
	Number of division ratios	Division ratio 1	Division ratio 2	
	SUM			

- Command, H'3F, (one byte): Selection of new bit rate
- Size (one byte): The total number of bytes that represents the bit rate, input frequency of division ratios, and division ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 1(when the value is 20.00 MHz, it will be 2000, which is H'07D0.)

- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

Error Response	H'BF	ERROR

- Error response, H'BF, (one byte): Error response to selection of new bit rate
- ERROR: (one byte): Error code

H'11:	Sum checking error
H'24:	Bit-rate selection error The rate is not available.
H'25:	Error in input frequency This input frequency is not within the specified range.
H'26:	Division ratio error The ratio does not match an available ratio.
H'27:	Operating frequency error The frequency is not within the specified range.

3. Operating frequency error

Operating frequency is calculated from the received value of the input frequency and division ratio. The input frequency is input to the LSI and the LSI is operated at the op frequency. The expression is given below.

Operating frequency = Input frequency ÷ Division ratio

The calculated operating frequency should be checked to ensure that it is within the ra minimum to maximum frequencies which are available with the clock modes of the sp device. When it is out of this range, an operating frequency error is generated.

4. Bit rate

To facilitate error checking, the value (n) of clock select (CKS) in the serial mode reg (SMR), and the value (N) in the bit rate register (BRR), which are found from the peri operating clock frequency (ϕ) and bit rate (B), are used to calculate the error rate to en it is less than 4%. If the error is more than 4%, a bit rate error is generated. The error i calculated using the following expression:

$$\text{Error (\%)} = \{[\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2\times n - 1)}}] - 1\} \times 100$$

When the new bit rate is selectable, the rate will be set in the register after sending ACK i response. The host will send an ACK with the new bit rate for confirmation and the boot will response with that rate.

Confirmation | H'06 |
|---|

- Confirmation, H'06, (one byte): Confirmation of a new bit rate

Response | H'06 |
|---|

- Response, H'06, (one byte): Response to confirmation of a new bit rate

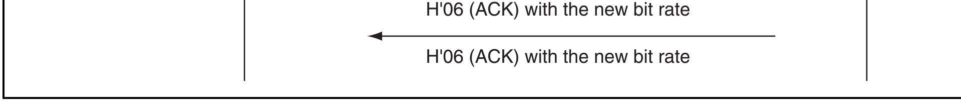

H'06 (ACK) with the new bit rate
H'06 (ACK) with the new bit rate

Figure 24.21 New Bit-Rate Selection Sequence

(5) Transition to Programming/Erasing State

The boot program will transfer the erasing program and erase the data in the user MATs the data in the user boot MATs. On completion of this erasure, ACK will be returned an program will enter the programming/erasing state.

The host should select the device code, clock mode, and new bit rate with device selecti mode selection, and new bit-rate selection commands, and then send the command for t transition to programming/erasing state. These procedures should be carried out before s the programming selection command or program data.

Command H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MATs and the user boot MATs hav erased by the transferred erasing program.

Error Response H'C0 H'51

- Error response, H'C0, (one byte): Error response to the bland check of the user boot
- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

The order for commands in the inquiry selection state is shown below.

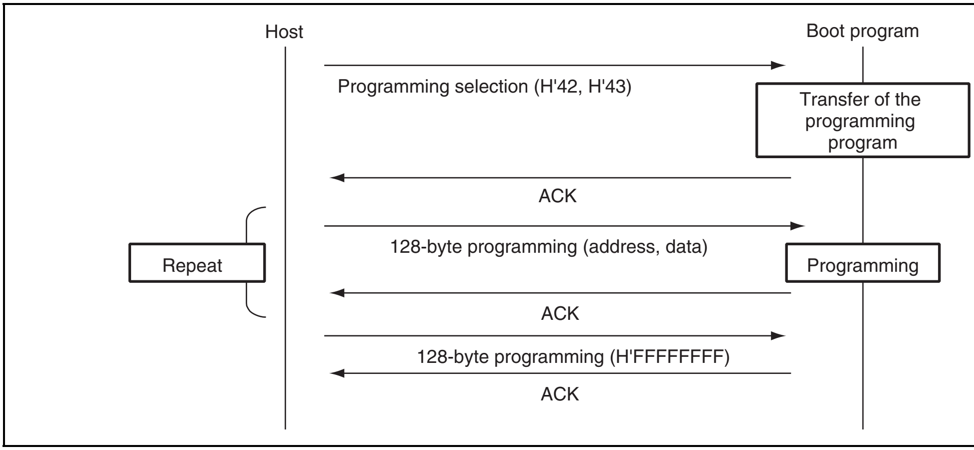
1. A supported device inquiry (H'20) should be made to inquire about the supported dev
2. The device should be selected from among those described by the returned informatic with a device-selection (H'10) command.
3. A clock-mode inquiry (H'21) should be made to inquire about the supported clock mc
4. The clock mode should be selected from among those described by the returned inforr and set.
5. After selection of the device and clock mode, inquiries for other required information be made, such as the division-ratio inquiry (H'22) or operating frequency inquiry (H'2 are needed for a new bit-rate selection.
6. A new bit rate should be selected with the new bit-rate selection (H'3F) command, acc to the returned information on division ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT a user MAT should be made to inquire about the user boot MATs information inquiry (user MATs information inquiry (H'25), erased block information inquiry (H'26), and programming unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

H'43	User MAT programming selection	Transfers the user MAT programming
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user boo
H'4B	User MAT sum check	Checks the checksum of the user MA
H'4C	User boot MAT blank check	Checks the blank data of the user boo
H'4D	User MAT blank check	Checks the blank data of the user MA
H'4F	Boot program status inquiry	Inquires into the boot program's statu

wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programmin another method or of another MAT, the procedure must be repeated from the program selection command.

The sequence for the programming selection and 128-byte programming commands is in figure 24.22.



Host
Boot program
Programming selection (H'42, H'43)
Transfer of the programming program
ACK
128-byte programming (address, data)
Repeat
Programming
ACK
128-byte programming (H'FFFFFFFF)
ACK

Figure 24.22 Programming Sequence

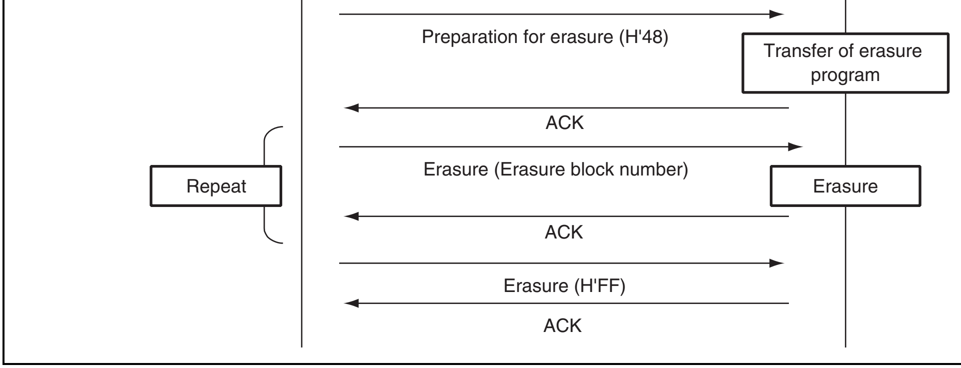

Preparation for erasure (H'48)
Transfer of erasure program
ACK
Erasure (Erasure block number)
Repeat
Erasure
ACK
Erasure (H'FF)
ACK

Figure 24.23   Erasure Sequence

When the programming program has been transferred, the boot program will return A

Error Response	H'C2	ERROR

- Error response: H'C2 (1 byte): Error response to user boot MAT programming selecti
- ERROR : (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complete

(b) User MAT Programming Selection

The boot program will transfer a program for user MAT programming selection. The data programmed to the user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (one byte): User-program programming selection

Response	H'06

- Response, H'06, (one byte): Response to user-program programming selection

  When the programming program has been transferred, the boot program will return A

Error Response	H'C3	ERROR

- Error response : H'C3 (1 byte): Error response to user-program programming selectio
- ERROR : (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complete

- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00 : H'00010000)
- Program data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum Error
  H'2A: Address Error
  The address is not within the specified MAT range.
  H'53: Programming error
  A programming error has occurred and programming cannot be contin

The specified address should match the unit for programming of data. For example, whe programming is in 128-byte units, the lower eight bits of the address should be H'00 or H When there are less than 128 bytes of data to be programmed, the host should fill the res H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop programming operation. The boot program will interpret this as the end of the programm wait for selection of programming or erasing.

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum error

  H'53: Programming error

  An error has occurred in programming and programming cannot be con

(d) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the trans erasure program.

Command	H'48

- Command, H'48, (one byte): Erasure selection

Response	H'06

- Response, H'06, (one byte): Response for erasure selection

  After the erasure program has been transferred, the boot program will return ACK.

Error Response	H'C8	ERROR

- ERROR: (one byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complet

Response	H'06	

- Response, H'06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code

H'11:	Sum check error
H'29:	Block number error Block number is incorrect.
H'51:	Erasure error An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a sele command.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proc should be executed from the erasure selection command.

H'01: User MAT

An address error occurs when the area setting is incorrect.

- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response

H'52	Read size							
Data	…							
SUM								

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

H'11: Sum check error
H'2A: Address error

The read address is not in the MAT.

H'2B: Size error

The read size exceeds the MAT.

This is fixed to 4.

- Checksum of MAT (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(h) User-Program Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the us program.

Command	H'4B

- Command, H'4B, (one byte): Sum check for user program

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

Error Response H'CC H'52

- Error Response, H'CC, (one byte): Error response to the blank check of user boot MA
- Error code, H'52, (one byte): Erasure has not been completed.

(j) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resul

Command H'4D

- Command, H'4D, (one byte): Blank check for user MATs

Response H'06

- Response, H'06, (one byte): Response to the blank check for user MATs
  If the contents of all user MATs are blank (H'FF), the boot program will return ACK.

Error Response H'CD H'52

- Error Response, H'CD, (one byte): Error response to the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

- Status (one byte): State of the boot program
- ERROR (one byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (one byte): Sum check

Table 24.19 Status Codes

Code	Description
H'11	Device selection wait
H'12	Clock mode selection wait
H'13	Bit rate selection wait
H'1F	Programming/erasing state transition wait (bit rate selection is completed)
H'31	Programming state for erasure
H'3F	Programming/erasing selection wait (erasure is completed)
H'4F	Program data receive wait
H'5F	Erase block specification wait (erasure is completed)

H'26	Division ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data length error
H'51	Erasure error
H'52	Erasure incomplete error
H'53	Programming error
H'54	Selection processing error
H'80	Command error
H'FF	Bit-rate-adjustment confirmation error

3.3 V programming voltage. Use only the specified socket adapter.

5. Do not power off the Vcc power supply (including the removal of the chip from the programmer) during programming/erasing in which a high voltage is applied to the f memory. Doing so may damage the flash memory permanently. If a reset is input, th must be released after the reset input period of at least 100μs.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing the operating mode is changed and this LSI is restarted by a reset immediately after programming/erasing has finished, secure the reset input period (period of $\overline{\text{RES}}$ = 0) 100μs. Transition to the reset state during programming/erasing is inhibited. If a rese accidentally, the reset must be released after the reset input period of at least 100μs.
7. At powering on the Vcc power supply, fix the $\overline{\text{RES}}$ pin to low and set the flash mem hardware protection state. This power on timing must also be satisfied at a power-of power-on caused by a power failure and other factors.
8. In on-board programming mode or programmer mode, programming of the 128-byte programming-unit block must be performed only once. Perform programming in the where the programming-unit block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra erasure in on-board programming mode, it is recommended that automatic programm performed after execution of automatic erasure.
10. To program the flash memory, the program data and program must be allocated to a which are higher than those of the external interrupt vector table and H'FF must be w all the system reserved areas in the exception handling vector table.
11. If data other than H'FF (4 bytes) is written to the key code area (H'00003C to H'0000 flash memory, reading cannot be performed in programmer mode. (In this case, data H'00. Rewrite is possible after erasing the data.) For reading in programmer mode, m to write H'FF to the entire key code area.

execute programming/erasing of the flash memory in this F-ZTAT H8/H8S microcom

15. Unlike a conventional F-ZTAT H8/H8S microcomputers, measures against a program are not taken by WDT while programming/erasing and downloading a programming/e program. When needed, measures should be taken by user. A periodic interrupt gener the WDT can be used as the measures, as an example. In this case, the interrupt gener period should take into consideration time to program/erase the flash memory.

16. When downloading the programming/erasing program, do not clear the SCO bit in FC after immediately setting it to 1. Otherwise, download cannot be performed normally. Immediately after executing the instruction to set the SCO bit to 1, dummy read of the must be executed twice.

17. The contents of some registers are not saved in a programming/programming end/eras program. When needed, save registers in the procedure program.

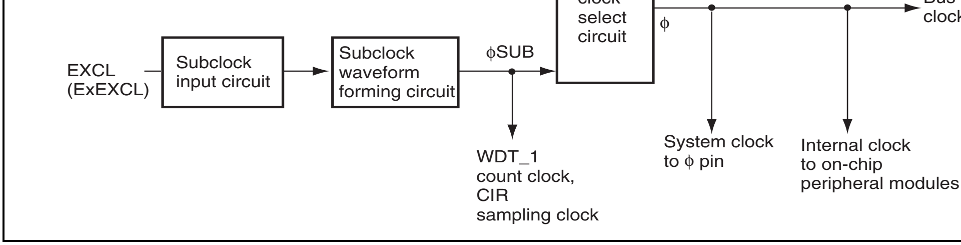


Figure 25.1 Block Diagram of Clock Pulse Generator

The subclock input is controlled by software according to the EXCLE bit and the EXCL
the port control register (PTCNT0) settings in the low power control register (LPWRCR
details on LPWRCR, see section 26.1.2, Low-Power Control Register (LPWRCR). For
PTCNT0, see section 7.3.1, Port Control Register 0 (PTCNT0).

Figure 25.3 shows an equivalent circuit of a crystal resonator. A crystal resonator having characteristics given in table 25.2 should be used.

The frequency of the crystal resonator should be the same as that of the system clock (ϕ).

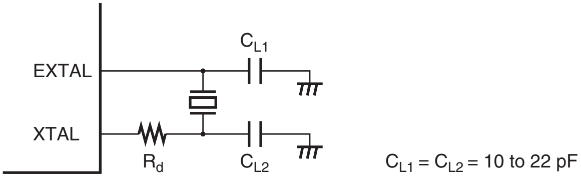


Figure 25.2 Typical Connection to Crystal Resonator

Table 25.1 Damping Resistor Values

Frequency (MHz)	8	10	12	16
R_d (Ω)	200	0	0	0

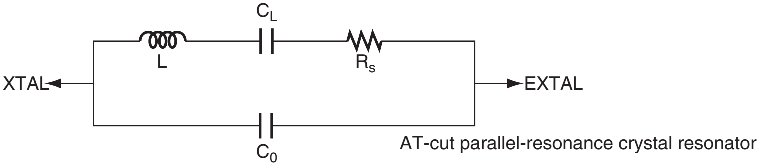
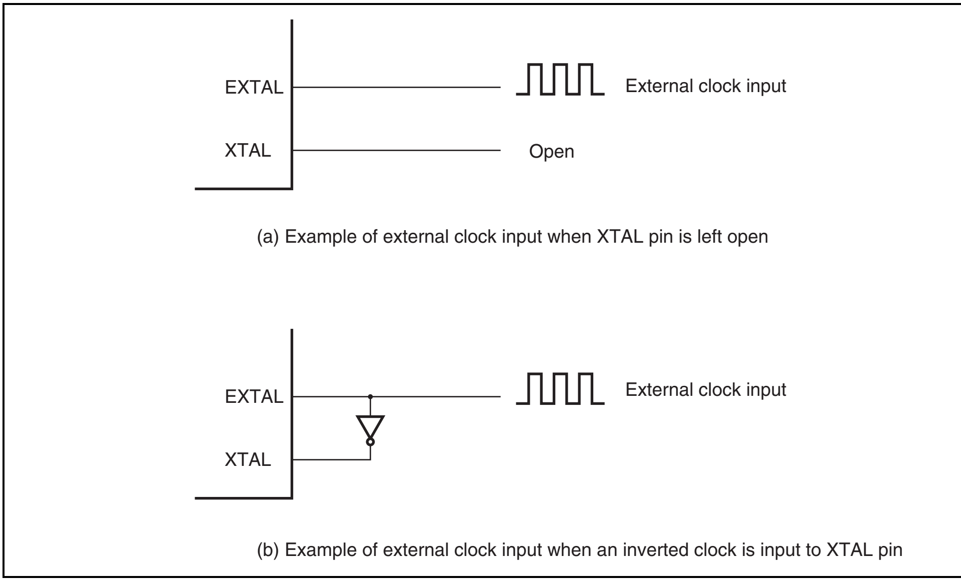


Figure 25.3 Equivalent Circuit of Crystal Resonator

the external clock should be set to high in standby mode or watch mode. External clock conditions are shown in table 25.3. The frequency of the external clock should be the sa of the system clock (ϕ).



(a) Example of external clock input when XTAL pin is left open

(b) Example of external clock input when an inverted clock is input to XTAL pin

Figure 25.4 Example of External Clock Input

Clock pulse width low level	t_{CL}	0.4	0.6	t_{cyc}	Figure 28.4
Clock pulse width high level	t_{CH}	0.4	0.6	t_{cyc}	

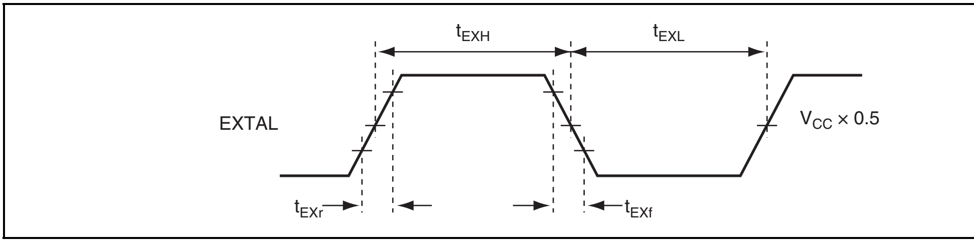
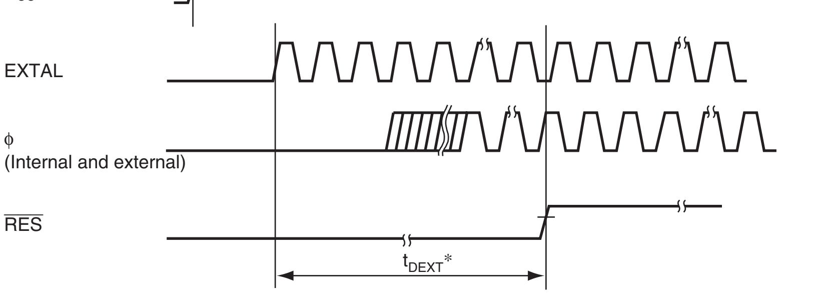


Figure 25.5 External Clock Input Timing

The oscillator and duty correction circuit can adjust the waveform of the external clock in is input from the EXTAL pin.

When a specified clock signal is input to the EXTAL pin, internal clock signal output is determined after the external clock output stabilization delay time (t_{DEXT}) has passed. As th signal output is not determined during the t_{DEXT} cycle, a reset signal should be set to low to maintain the reset state. Table 25.4 shows the external clock output stabilization delay tim 25.6 shows the timing of the external clock output stabilization delay time.



Note: * The external clock output stabilization delay time (t_{DEXT}) includes a $\overline{RES}$ pulse width (t_{RESW})

Figure 25.6 Timing of External Clock Output Stabilization Delay Time

Figure 25.7 shows the relationship of subclock input from the EXCL pin and the ExEXC When using a pin to input the subclock, specify input for the pin by clearing the DDR bit pin to 0. The EXCL pin is specified as an input pin by clearing the EXCLS bit in PTCNT The ExEXCL pin is specified as an input pin by setting the EXCLS bit in PTCNT0 to 1. T subclock input is enabled by setting the EXCLE bit in LPWRCR to 1.

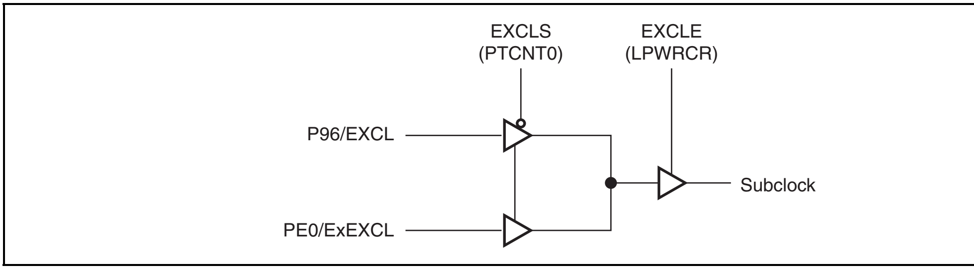


Figure 25.7 Subclock Input from EXCL Pin and ExEXCL Pin

Subclock input conditions are shown in table 25.5. When the subclock is not used, subclc should not be enabled.

Table 25.5 Subclock Input Conditions

		VCC = 3.0 to 3.6 V				
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Test Co**
Subclock input pulse width low level	t_{EXCLL}	—	15.26	—	µs	Figure 25
Subclock input pulse width high level	t_{EXCLH}	—	15.26	—	µs	
Subclock input rising time	t_{EXCLr}	—	—	10	ns	
Subclock input falling time	t_{EXCLf}	—	—	10	ns	

To remove noise from the subclock input at the EXCL (ExEXCL) pin, the subclock wav forming circuit samples the subclock using a divided ϕ clock. The sampling frequency is NESEL bit in LPWRCR.

The subclock is not sampled in watch mode.

25.5 Clock Select Circuit

The clock select circuit selects the system clock that is used in this LSI.

A clock generated by the oscillator to which the XTAL and EXTAL pins are connected as a system clock (ϕ) when returning from high-speed mode, sleep mode, the reset state, standby mode.

In watch mode, a subclock input from the EXCL (ExEXCL) pin is selected as a system when the EXCLE bit in LPWRCR is 1. At this time, on-chip peripheral modules such as and interrupt controller operate on the ϕSUB clock. The count clock and sampling clock timer are divided ϕSUB clocks.

25.6.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be pla
close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away
the oscillator to prevent inductive interference with correct oscillation as shown in figure

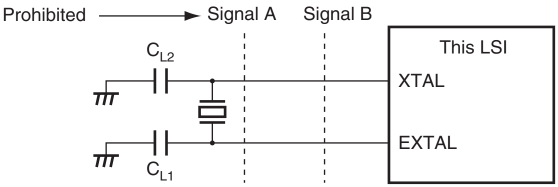


Figure 25.9 Note on Board Design of Oscillator Section

The CPU stops but on-chip peripheral modules continue operating.

- Watch mode

  The CPU stops, but on-chip peripheral module WDT_1 and CIR continue operating.

- Software standby mode

  The clock pulse generator stops, and the CPU and on-chip peripheral modules stop c

- Module stop mode

  Independently of above operating modes, on-chip peripheral modules that are not us stopped individually.

26.1 Register Descriptions

Power-down modes are controlled by the following registers. To access SBYCR, LPWF SYSCR2, MSTPCRH, and MSTPCRL the FLSHE bit in the serial timer control register must be cleared to 0. For details on STCR, see section 3.2.3, Serial Timer Control Regis (STCR). For details on the PSS bit in TSCR_1 (WDT_1), see TCSR_1 in section 13.3.5 Control/Status Register (TCSR).

Table 26.1 Register Configuration

					Da
Register Name	**Abbreviation**	**R/W**	**Initial Value**	**Address**	**W**
Standby control register	SBYCR	R/W	H'00	H'FF84	8
Low power control register	LPWRCR	R/W	H'00	H'FF85	8
Module stop control register H	MSTPCRH	R/W	H'3F	H'FF86	8
Module stop control register L	MSTPCRL	R/W	H'FF	H'FF87	8
Module stop control register A	MSTPCRA	R/W	H'FC	H'FE7E	8
Module stop control register B	MSTPCRB	R/W	H'FF	H'FE7F	8

				0: Shifts to sleep mode 1: Shifts to software standby mode or watch m Note that the SSBY bit is not changed even if transition is made by an interrupt.
6 5 4	STS2 STS1 STS0	0 0 0	R/W R/W R/W	Standby Timer Select 2 to 0 On canceling software standby mode or watcl these bits select the wait time for clock stabili from clock oscillation start. Select a wait time (oscillation stabilization time) or more, depend the operating frequency. Table 26.2 shows th relationship between the STS2 to STS0 value wait time. With an external clock, an arbitrary wait time selected. For normal cases, the minimum valu recommended.
3	—	0	R/W	Reserved The initial value should not be changed.

011: Medium-speed clock: ϕ/8

100: Medium-speed clock: ϕ/16

101: Medium-speed clock: ϕ/32

11X: Setting prohibited

[Legend]

X: Don’t care

Table 26.2 Operating Frequency and Wait Time

STS2	STS1	STS0	Wait Time	20 MHz	10 MHz	8 MHz	Un
0	0	0	8192 states	0.4	0.8	1.0	ms
0	0	1	16384 states	0.8	1.6	2.0	
0	1	0	32768 states	1.6	3.3	4.1	
0	1	1	65536 states	3.3	6.6	8.2	
1	0	0	131072 states	6.6	13.1	16.4	
1	0	1	262144 states	13.1	26.2	32.8	
1	1	0/1	Reserved*	—	—	—	—

▨ Recommended specification

Note: * Setting prohibited.

5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select Selects the frequency by which the subclock (input from the EXCL or ExEXCL pin is sample the clock (ϕ) generated by the system clock p generator. Clear this bit to 0 when ϕ is 5 MHz The initial value should not be changed. 0: Sampling using ϕ/32 clock 1: Sampling using ϕ/4 clock (not allowed)
4	EXCLE	0	R/W	Subclock Input Enable Enables or disables subclock input from the E ExEXCL pin. 0: Disables subclock input from the EXCL or E pin 1: Enables subclock input from the EXCL or E pin
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

6	MSTP14	0	R/W	Reserved The initial value should not be changed.
5	MSTP13	1	R/W	Reserved The initial value should not be changed.
4	MSTP12	1	R/W	8-bit timers (TMR_0 and TMR_1)
3	MSTP11	1	R/W	14-bit PWM timer (PWMX)
2	MSTP10	1	R/W	Reserved The initial value should not be changed.
1	MSTP9	1	R/W	A/D converter
0	MSTP8	1	R/W	8-bit timers (TMR_X and TMR_Y)

- MSTPCRL

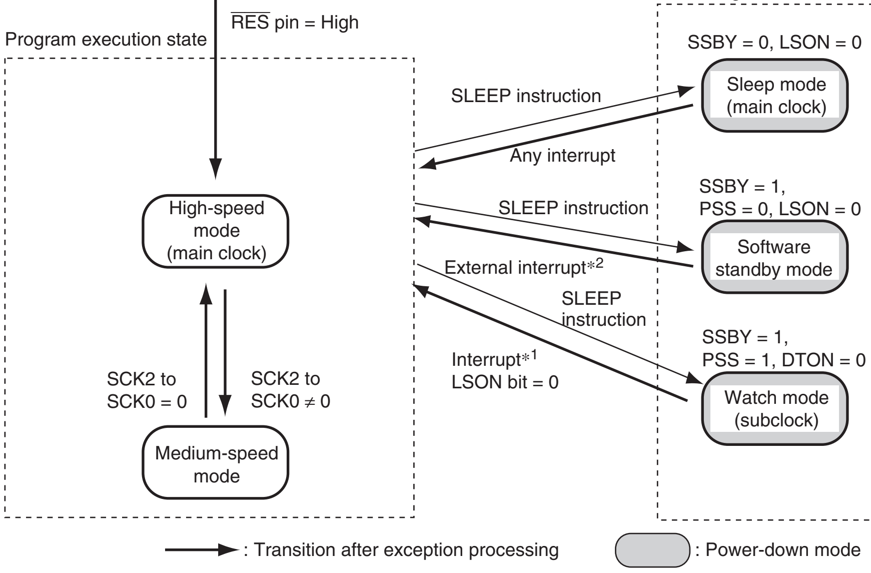
Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP7	1	R/W	Reserved The initial value should not be changed.
6	MSTP6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTP5	1	R/W	Serial communication interface 2 (SCI_2)
4	MSTP4	1	R/W	I^{2}C bus interface channel 0 (IIC_0)
3	MSTP3	1	R/W	I^{2}C bus interface channel 1 (IIC_1)
2	MSTP2	1	R/W	Keyboard buffer control unit_0 (PS2_0) Keyboard buffer control unit_1 (PS2_1) Keyboard buffer control unit_2 (PS2_2)
1	MSTP1	1	R/W	16-bit timer pulse unit (TPU)
0	MSTP0	1	R/W	LPC interface (LPC)

1	MSTPA1	0	R/W	14-bit PWM timer (PWMX)
0	MSTPA0	0	R/W	Reserved The initial value should not be changed.

Note: * Before accessing registers of the FSI interface, clear bit 0 in MSTPCRL (MSTP bit 2 in MSTPCRA (MSTPA2) to 0.

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPB7	1	R/W	Reserved The initial value should not be changed.
6	MSTPB6	1	R/W	Reserved The initial value should not be changed.
5	MSTPB5	1	R/W	Keyboard buffer control unit_3 (PS2_3)
4	MSTPB4	1	R/W	I^2C bus interface_2 (IIC_2)
3	MSTPB3	1	R/W	Serial communication interface with FIFO (SCIF)
2	MSTPB2	1	R/W	Cycle measurement timer_2 (TCM_2) Cycle measurement timer_3 (TCM_3)
1	MSTPB1	1	R/W	Cycle measurement timer_0 (TCM_0) Cycle measurement timer_1 (TCM_1)
0	MSTPB0	1	R/W	8-bit PWMU timer_A (TWMU_A) 8-bit PWMU timer_B (TWMU_B)



Notes: When a transition is made between modes by means of an interrupt, the transition cannot be mad on interrupt source generation alone. Ensure that interrupt handling is performed after accepting t interrupt request.

1. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, and WDT_1, PS2 and CIR interrupts
2. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, and PS2 interrupts

Figure 26.1 Mode Transition Diagram

	KIN0 to KIN15					
	WUE8 to WUE15					
On-chip peripheral modules	WDT_1	Functioning	Functioning	Functioning	Functioning	Subclock operation
	CIR					
	WDT_0					Stopped (retained)
	TMR_0, TMR_1				Functioning/ stopped (retained)	
	TPU TCM_0 to 3 TDP_0 to 2					
	TMR_X, TMR_Y					
	SCIF					
	IIC_0 to 2					
	LPC					
	FSI					
	PS2_0 to 3		Medium-speed operation/functioning			
	PWMU		Functioning		Functioning/ stopped (reset)	Stopped (reset)
	PWM					
	PWMX					
	SCI_1, SCI_2					
	A/D converter					
	RAM	Functioning	Functioning	Functioning	Functioning	Retained
	I/O	Functioning	Functioning	Functioning	Functioning	Retained

Note: Stopped (retained) means that the internal register values are retained and the in state is operation suspended. Stopped (reset) means that the internal register val the internal state are initialized. In module stop mode, only modules for which a s has been made are stopped (reset or retained).

A transition is made from medium-speed mode to high-speed mode at the end of the curr cycle by clearing all of bits SCK2 to SCK0 to 0.

If the SLEEP instruction is executed when the SSBY bit in SBYCR is 0 and the LSON bi LPWRCR is 0, a transition is made to sleep mode. When sleep mode is canceled by an in medium-speed mode is restored. When the SLEEP instruction is executed with the SSBY 1, the LSON bit in LPWRCR set to 0, and the PSS bit in TCSR (WDT_1) set to 0, operat to software standby mode. When software standby mode is canceled by an external interr medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low and medium-speed mode is cancelled, operation shifts to state. The same applies to a reset caused by an overflow of the watchdog timer.

Figure 26.2 shows the timing of medium-speed mode.

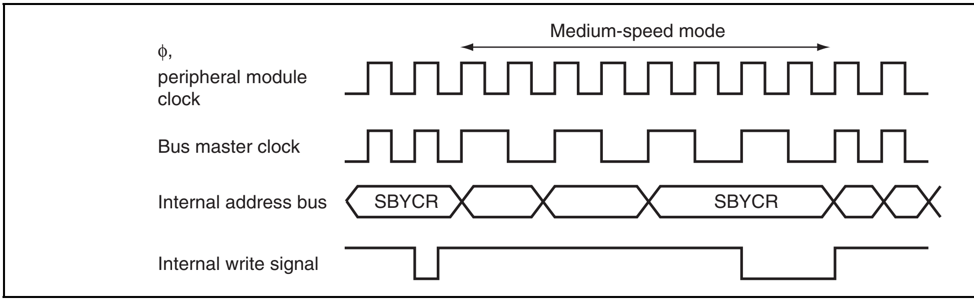


Figure 26.2 Timing of Medium-Speed Mode

the CPU.

When the $\overline{\text{RES}}$ pin is driven low and sleep mode is cleared, a transition is made to the re After the specified reset input time has elapsed, driving the $\overline{\text{RES}}$ pin high causes the CP reset exception handling.

26.5 Software Standby Mode

The CPU makes a transition to software standby mode when the SLEEP instruction is e with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, and the TCSR (WDT_1) cleared to 0. In software standby mode, the CPU, on-chip peripheral m and clock pulse generator all stop. However, the contents of the CPU registers, on-chip I/O ports, and the states of on-chip peripheral modules other than the SCI, PWMU, PW A/D converter are retained as long as the prescribed voltage is supplied.

Software standby mode is cleared by an external interrupt (NMI, IRQ0 to IRQ15, KIN0 or WUE8 to WUE15), PS2 interrupt, or $\overline{\text{RES}}$ pin input.

When an external interrupt request signal is input, system clock oscillation starts, and af elapse of the time set in bits STS2 to STS0 in SBYCR, software standby mode is cleare interrupt exception handling is started. When clearing software standby mode with an I IRQ15 interrupt, set the corresponding enable bit to 1. When clearing software standby a KIN0 to KIN15 or WUE8 to WUE15 interrupt, enable the input. In these cases, ensur interrupt with a higher priority than interrupts IRQ0 to IRQ15 is generated. In the case to IRQ15 interrupt, software standby mode is not cleared if the corresponding enable bi to 0 or if the interrupt has been masked by the CPU. In the case of a KIN0 to KIN15 or WUE15 interrupt, software standby mode is not cleared if the input is disabled or if the has been masked by the CPU.

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 of the NMI pin.

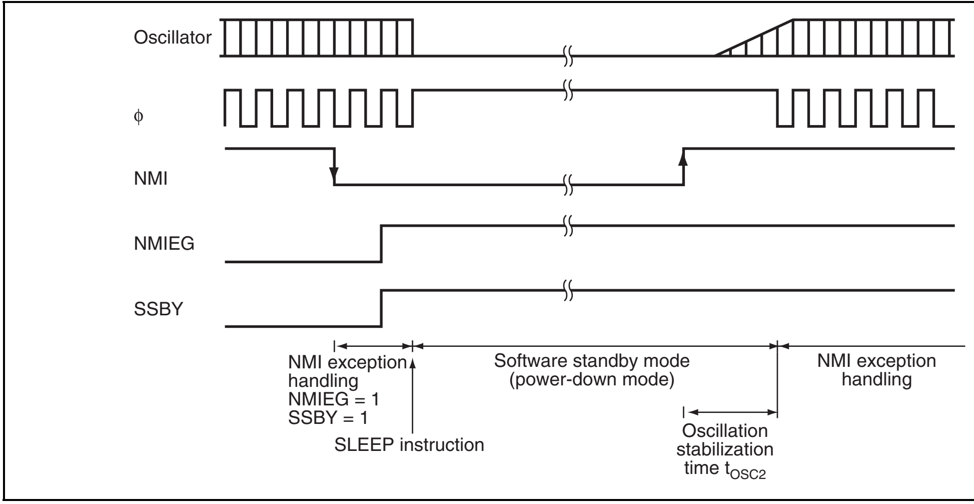


Figure 26.3 Software Standby Mode Application Example

Watch mode is cleared by an interrupt (WOVI1, NMI, IRQ0 to IRQ15, KIN0 to KIN15, to WUE15), PS2 interrupt, CIR interrupt, or $\overline{\text{RES}}$ pin input.

When an interrupt occurs, watch mode is cleared and a transition is made to high-speed medium-speed mode. When a transition is made to high-speed mode, a stable clock is su the entire LSI and interrupt exception handling starts after the time set in the STS2 to ST SBYCR has elapsed. In the case of an IRQ0 to IRQ15 interrupt, watch mode is not clear corresponding enable bit has been cleared to 0 or the interrupt has been masked by the C case of a KIN0 to KIN15 or WUE8 to WUE15 interrupt, watch mode is not cleared if th disabled or the interrupt has been masked by the CPU. In the case of an interrupt from a peripheral module, watch mode is not cleared if the interrupt enable register has been se the reception of that interrupt or the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, the clock pulse generator starts oscillation. Simultaneo the start of system clock oscillation, the system clock is supplied to the entire LSI. Note $\overline{\text{RES}}$ pin must be held low until clock oscillation is stabilized. If the $\overline{\text{RES}}$ pin is driven hi the clock oscillation stabilization time has elapsed, the CPU starts reset exception handli

While an on-chip peripheral module is in module stop mode, its registers cannot be read f written to.

26.8 Usage Notes

26.8.1 I/O Port Status

The status of the I/O ports is retained in software standby mode. Therefore, while a high l output or the pull-up MOS is on, the current consumption is not reduced by the amount o to support the high level output.

26.8.2 Current Consumption when Waiting for Oscillation Stabilization

The current consumption increases during oscillation stabilization.

- The access size is indicated.
- H8S/2140B Group compatible register addresses or extended register addresses are s depending on the RELOCATE bit in system control register 3 (SYSCR3).

  When the extended register addresses are selected, the some register addresses of IC TMR_Y, PWMX_0, and PORT are changed. Therefore, the selection with other mo registers that share the same addresses with these registers is not necessary.

2. Register bits

- Bit configurations of the registers are described in the same order as the register add section 27.1, Register Addresses (Address Order).
- Reserved bits are indicated by — in the bit name column.
- The bit number in the bit-name column indicates that the whole register is allocated counter or for holding data.
- Each line covers eight bits, and 16-bit register is shown as 2 lines, respectively.

3. Register states in each operating mode

- Register states are described in the same order as the register addresses in section 27 Register Addresses (Address Order).
- The register states described here are for the basic operating modes. If there is a spe for an on-chip peripheral module, see the section on that on-chip peripheral module.

4. Register selection conditions

- Register selection conditions are described in the same order as the register addresse section 27.1, Register Addresses (Address Order).
- For register selection conditions, see section 3.2.2, System Control Register (SYSC 3.2.3, Serial Timer Control Register (STCR), section 26.1.3, Module Stop Control R L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB), or register descript each module.

5. Register addresses (classification by type of module)

- The register addresses are described by modules
- The register addresses are described in channel order when the module has multiple

Port 1 data register	P1DR	8	H'F902 (PORTS = 1)	PORT	8
Port 2 data register	P2DR	8	H'F903 (PORTS = 1)	PORT	8
Port 1 input data register	P1PIN	8	H'F904 (Read) (PORTS = 1)	PORT	8
Port 2 input data register	P2PIN	8	H'F905 (Read) (PORTS = 1)	PORT	8
Port 1 pull-up MOS control register	P1PCR	8	H'F906 (PORTS = 1)	PORT	8
Port 2 pull-up MOS control register	P2PCR	8	H'F907 (PORTS = 1)	PORT	8
Port 3 data direction register	P3DDR	8	H'F910 (PORTS = 1)	PORT	8
Port 4 data direction register	P4DDR	8	H'F911 (PORTS = 1)	PORT	8
Port 3 data register	P3DR	8	H'F912 (PORTS = 1)	PORT	8
Port 4 data register	P4DR	8	H'F913 (PORTS = 1)	PORT	8
Port 3 input data register	P3PIN	8	H'F914 (Read) (PORTS = 1)	PORT	8
Port 4 input data register	P4PIN	8	H'F915 (Read) (PORTS = 1)	PORT	8
Port 3 pull-up MOS control register	P3PCR	8	H'F916 (PORTS = 1)	PORT	8

			(PORTS = 1)		
Port 6 input data register	P6PIN	8	H'F925 (Read) (PORTS = 1)	PORT	8
Port 6 noise canceler enable register	P6NCE	8	H'F92B (PORTS = 1)	PORT	8
Port 6 noise canceler decision control register	P6NCMC	8	H'F92D (PORTS = 1)	PORT	8
Port 6 noise cancel cycle setting register	P6NCCS	8	H'F92F (PORTS = 1)	PORT	8
Port 8 data direction register	P8DDR	8	H'F931 (PORTS = 1)	PORT	8
Port 8 data register	P8DR	8	H'F933 (PORTS = 1)	PORT	8
Port 7 input data register	P7PIN	8	H'F934 (Read) (PORTS = 1)	PORT	8
Port 8 input data register	P8PIN	8	H'F935 (Read) (PORTS = 1)	PORT	8
Port 9 data direction register	P9DDR	8	H'F940 (PORTS = 1)	PORT	8
Port 9 data register	P9DR	8	H'F942 (PORTS = 1)	PORT	8
Port 9 input data register	P9PIN	8	H'F944 (Read) (PORTS = 1)	PORT	8
Port 9 pull-up MOS control register	P9PCR	8	H'F946 (PORTS = 1)	PORT	8
Port A data direction register	PADDR	8	H'F950 (PORTS = 1)	PORT	8

			(PORTS = 1)		
Port B pull-up MOS control register	PBPCR	8	H'F957 (PORTS = 1)	PORT	8
Port C data direction register	PCDDR	8	H'F960 (PORTS = 1)	PORT	8
Port D data direction register	PDDDR	8	H'F961 (PORTS = 1)	PORT	8
Port C output data register	PCODR	8	H'F962 (PORTS = 1)	PORT	8
Port D output data register	PDODR	8	H'F963 (PORTS = 1)	PORT	8
Port C input data register	PCPIN	8	H'F964 (Read) (PORTS = 1)	PORT	8
Port D input data register	PDPIN	8	H'F965 (Read) (PORTS = 1)	PORT	8
Port C pull-up MOS control register	PCPCR	8	H'F966 (PORTS = 1)	PORT	8
Port D pull-up MOS control register	PDPCR	8	H'F967 (PORTS = 1)	PORT	8
Port C Nch-OD control register	PCNOCR	8	H'F968 (PORTS = 1)	PORT	8
Port D Nch-OD control register	PDNOCR	8	H'F969 (PORTS = 1)	PORT	8
Port C noise canceler enable register	PCNCE	8	H'F96A (PORTS = 1)	PORT	8
Port C noise canceler decision control register	PCNCMC	8	H'F96C (PORTS = 1)	PORT	8
Port C noise cancel cycle setting register	PCNCCS	8	H'F96E (PORTS = 1)	PORT	8

			(PORTS = 1)		
Port F Nch-OD control register	PFNOCR	8	H'F979 (PORTS = 1)	PORT	8
Port G data direction register	PGDDR	8	H'F980 (PORTS = 1)	PORT	8
Port H data direction register	PHDDR	8	H'F981 (PORTS = 1)	PORT	8
Port G output data register	PGODR	8	H'F982 (PORTS = 1)	PORT	8
Port H output data register	PHODR	8	H'F983 (PORTS = 1)	PORT	8
Port G input data register	PGPIN	8	H'F984 (Read) (PORTS = 1)	PORT	8
Port H input data register	PHPIN	8	H'F985 (Read) (PORTS = 1)	PORT	8
Port H pull-up MOS control register	PHPCR	8	H'F987 (PORTS = 1)	PORT	8
Port G Nch-OD control register	PGNOCR	8	H'F988 (PORTS = 1)	PORT	8
Port H Nch-OD control register	PHNOCR	8	H'F989 (PORTS = 1)	PORT	8
Port G noise canceler enable register	PGNCE	8	H'F98A (PORTS = 1)	PORT	8
Port G noise canceler decision control register	PGNCMC	8	H'F98C (PORTS = 1)	PORT	8
Port G noise cancel cycle setting register	PGNCCS	8	H'F98E (PORTS = 1)	PORT	8

Port J Nch-OD control register	PJNOCR	8	H'F999	PORT	8
Receive control register 1	CCR1	8	H'FA40	CIR	8
Receive control register 2	CCR2	8	H'FA41	CIR	8
Receive status register	CSTR	8	H'FA42	CIR	8
Interrupt enable register	CEIR	8	H'FA43	CIR	8
Bit rate register	BRR	8	H'FA44	CIR	8
Receive data register 0 to 7	CIRRDR0 to CIRRDR7	8	H'FA45	CIR	8
Header minimum high-level period register	HHMIN	16	H'FA46	CIR	8
Header maximum high-level period register	HHMAX	16	H'FA48	CIR	8
Header minimum low-level period register	HLMIN	8	H'FA4A	CIR	8
Header maximum low-level period register	HLMAX	8	H'FA4B	CIR	8
Data level 0 minimum period register	DT0MIN	8	H'FA4C	CIR	8
Data level 0 maximum period register	DT0MAX	8	H'FA4D	CIR	8
Data level 1 minimum period register	DT1MIN	8	H'FA4E	CIR	8
Data level 1 maximum period register	DT1MAX	8	H'FA4F	CIR	8
Repeat header minimum low-level period register	RMIN	8	H'FA50	CIR	8
Repeat header maximum low-level period register	RMAX	8	H'FA51	CIR	8

TDP status register_0	TDPCSR_0	8	H'FB4C	TDP_0	8
TDP control register 1_0	TDPCR1_0	8	H'FB4D	TDP_0	8
TDP interrupt enable register_0	TDPIER_0	8	H'FB4E	TDP_0	8
TDP control register 2_0	TDPCR2_0	8	H'FB4F	TDP_0	8
TDP cycle lower limit register_0	TDPPDMN_0	16	H'FB50	TDP_0	16
TDP timer counter_1	TDPCNT_1	16	H'FB60	TDP_1	16
TDP pulse width upper limit register_1	TDPWDMX_1	16	H'FB62	TDP_1	16
TDP pulse width lower limit register_1	TDPWDMN_1	16	H'FB64	TDP_1	16
TDP cycle upper limit register_1	TDPPDMX_1	16	H'FB66	TDP_1	16
TDP input capture register_1	TDPICR_1	16	H'FB68	TDP_1	16
TDP input capture buffer register_1	TDPICRF_1	16	H'FB6A	TDP_1	16
TDP status register_1	TDPCSR_1	8	H'FB6C	TDP_1	8
TDP control register 1_1	TDPCR1_1	8	H'FB6D	TDP_1	8
TDP interrupt enable register_1	TDPIER_1	8	H'FB6E	TDP_1	8
TDP control register 2_1	TDPCR2_1	8	H'FB6F	TDP_1	8
TDP cycle lower limit register_1	TDPPDMN_1	16	H'FB70	TDP_1	16
TDP timer counter_2	TDPCNT_2	16	H'FB80	TDP_2	16
TDP pulse width upper limit register_2	TDPWDMX_2	16	H'FB82	TDP_2	16
TDP pulse width lower limit register_2	TDPWDMN_2	16	H'FB84	TDP_2	16
TDP cycle upper limit register_2	TDPPDMX_2	16	H'FB86	TDP_2	16

TCM cycle upper limit register	TCMMLCM_0	16	H'FBC2	TCM_0	16
TCM input capture register_0	TCMICR_0	16	H'FBC4	TCM_0	16
TCM input capture buffer register_0	TCMICRF_0	16	H'FBC6	TCM_0	16
TCM status register_0	TCMCSR_0	8	H'FBC8	TCM_0	8
TCM control register_0	TCMCR_0	8	H'FBC9	TCM_0	8
TCM interrupt enable register_0	TCMIER_0	8	H'FBCA	TCM_0	8
TCM cycle lower limit register_0	TCMMINCM_0	16	H'FBCC	TCM_0	16
TCM timer counter_1	TCMCNT_1	16	H'FBD0	TCM_1	16
TCM cycle upper limit register_0	TCMMLCM_1	16	H'FBD2	TCM_1	16
TCM input capture register_1	TCMICR_1	16	H'FBD4	TCM_1	16
TCM input capture buffer register_1	TCMICRF_1	16	H'FBD6	TCM_1	16
TCM status register_1	TCMCSR_1	8	H'FBD8	TCM_1	8
TCM control register_1	TCMCR_1	8	H'FBD9	TCM_1	8
TCM interrupt enable register_1	TCMIER_1	8	H'FBDA	TCM_1	8
TCM cycle lower limit register_1	TCMMINCM_1	16	H'FBDC	TCM_1	16
TCM timer counter_2	TCMCNT_2	16	H'FBE0	TCM_2	16
TCM cycle upper limit register_2	TCMMLCM_2	16	H'FBE2	TCM_2	16
TCM input capture register_2	TCMICR_2	16	H'FBE4	TCM_2	16
TCM input capture buffer register_2	TCMICRF_2	16	H'FBE6	TCM_2	16
TCM status register_2	TCMCSR_2	8	H'FBE8	TCM_2	8
TCM control register_2	TCMCR_2	8	H'FBE9	TCM_2	8
TCM interrupt enable register_2	TCMIER_2	8	H'FBEA	TCM_2	8
TCM cycle lower limit register_2	TCMMINCM_2	16	H'FBEC	TCM_2	16
TCM timer counter_3	TCMCNT_3	16	H'FBF0	TCM_3	16
TCM cycle upper limit register_3	TCMMLCM_3	16	H'FBF2	TCM_3	16

A/D data register B	ADDRB	16	H'FC02	A/D converter	16
A/D data register C	ADDRC	16	H'FC04	A/D converter	16
A/D data register D	ADDRD	16	H'FC06	A/D converter	16
A/D data register E	ADDRE	16	H'FC08	A/D converter	16
A/D data register F	ADDRF	16	H'FC0A	A/D converter	16
A/D data register G	ADDRG	16	H'FC0C	A/D converter	16
A/D data register H	ADDRH	16	H'FC0E	A/D converter	16
A/D control/status register	ADCSR	8	H'FC10	A/D converter	8
A/D control register	ADCR	8	H'FC11	A/D converter	8
Receive buffer register	FRBR	8	H'FC20	SCIF	8
Transmitter holding register	FTHR	8	H'FC20	SCIF	8
Divisor latch L	FDLL	8	H'FC20	SCIF	8
Interrupt enable register	FIER	8	H'FC21	SCIF	8
Divisor latch H	FDLH	8	H'FC21	SCIF	8
Interrupt identification register	FIIR	8	H'FC22	SCIF	8
FIFO control register	FFCR	8	H'FC22	SCIF	8
Line control register	FLCR	8	H'FC23	SCIF	8
Modem control register	FMCR	8	H'FC24	SCIF	8
Line status register	FLSR	8	H'FC25	SCIF	8

FSI command host base address register H	CMDHBARH	8	H'FC53	FSI	8
FSI command host base address register L	CMDHBARL	8	H'FC54	FSI	8
FSI command register	FSICMDR	8	H'FC55	FSI	8
FSILPC command status register 1	FSILSTR1	8	H'FC56	FSI	8
FSI general-purpose register 1	FSIGPR1	8	H'FC57	FSI	8
FSI general-purpose register 2	FSIGPR2	8	H'FC58	FSI	8
FSI general-purpose register 3	FSIGPR3	8	H'FC59	FSI	8
FSI general-purpose register 4	FSIGPR4	8	H'FC5A	FSI	8
FSI general-purpose register 5	FSIGPR5	8	H'FC5B	FSI	8
FSI general-purpose register 6	FSIGPR6	8	H'FC5C	FSI	8
FSI general-purpose register 7	FSIGPR7	8	H'FC5D	FSI	8
FSI general-purpose register 8	FSIGPR8	8	H'FC5E	FSI	8
FSI general-purpose register 9	FSIGPR9	8	H'FC5F	FSI	8
FSI general-purpose register A	FSIGPRA	8	H'FC60	FSI	8
FSI general-purpose register B	FSIGPRB	8	H'FC61	FSI	8
FSI general-purpose register C	FSIGPRC	8	H'FC62	FSI	8
FSI general-purpose register D	FSIGPRD	8	H'FC63	FSI	8
FSI general-purpose register E	FSIGPRE	8	H'FC64	FSI	8
FSI general-purpose register F	FSIGPRF	8	H'FC65	FSI	8
FSILPC control register	SLCR	8	H'FC66	FSI	8
FSI address register H	FSIARH	8	H'FC67	FSI	8
FSI address register M	FSIARM	8	H'FC68	FSI	8
FSI address register L	FSIARL	8	H'FC69	FSI	8

FSI byte count register	FSIBNR	8	H'FC92	FSI	8
FSI instruction register	FSIINS	8	H'FC93	FSI	8
FSI read instruction register	FSIRDINS	8	H'FC94	FSI	8
FSI program instruction register	FSIPPINS	8	H'FC95	FSI	8
FSI status register	FSISTR	8	H'FC96	FSI	8
FSI transmit data register 0	FSITDR0	8	H'FC98	FSI	8
FSI transmit data register 1	FSITDR1	8	H'FC99	FSI	8
FSI transmit data register 2	FSITDR2	8	H'FC9A	FSI	8
FSI transmit data register 3	FSITDR3	8	H'FC9B	FSI	8
FSI transmit data register 4	FSITDR4	8	H'FC9C	FSI	8
FSI transmit data register 5	FSITDR5	8	H'FC9D	FSI	8
FSI transmit data register 6	FSITDR6	8	H'FC9E	FSI	8
FSI transmit data register 7	FSITDR7	8	H'FC9F	FSI	8
FSI receive data register	FSIRDR	8	H'FCA0	FSI	8
PWM duty setting register 0_A	PWMREG0_A	8	H'FD00	PWMU_A	8
PWM prescaler register 0_A	PWMPRE0_A	8	H'FD01	PWMU_A	8
PWM duty setting register 1_A	PWMREG1_A	8	H'FD02	PWMU_A	8
PWM prescaler register 1_A	PWMPRE1_A	8	H'FD03	PWMU_A	8
PWM duty setting register 2_A	PWMREG2_A	8	H'FD04	PWMU_A	8
PWM prescaler register 2_A	PWMPRE2_A	8	H'FD05	PWMU_A	8
PWM duty setting register 3_A	PWMREG3_A	8	H'FD06	PWMU_A	8
PWM prescaler register 3_A	PWMPRE3_A	8	H'FD07	PWMU_A	8
PWM duty setting register 4_A	PWMREG4_A	8	H'FD08	PWMU_A	8
PWM prescaler register 4_A	PWMPRE4_A	8	H'FD09	PWMU_A	8
PWM duty setting register 5_A	PWMREG5_A	8	H'FD0A	PWMU_A	8

PWM prescaler register 1_B	PWMPRE1_B	8	H'FD13	PWMU_B	8
PWM duty setting register 2_B	PWMREG2_B	8	H'FD14	PWMU_B	8
PWM prescaler register 2_B	PWMPRE2_B	8	H'FD15	PWMU_B	8
PWM duty setting register 3_B	PWMREG3_B	8	H'FD16	PWMU_B	8
PWM prescaler register 3_B	PWMPRE3_B	8	H'FD17	PWMU_B	8
PWM duty setting register 4_B	PWMREG4_B	8	H'FD18	PWMU_B	8
PWM prescaler register 4_B	PWMPRE4_B	8	H'FD19	PWMU_B	8
PWM duty setting register 5_B	PWMREG5_B	8	H'FD1A	PWMU_B	8
PWM prescaler register 5_B	PWMPRE5_B	8	H'FD1B	PWMU_B	8
PWM control register A_B	PWMCONA_B	8	H'FD1C	PWMU_B	8
PWM control register B_B	PWMCONB_B	8	H'FD1D	PWMU_B	8
PWM control register C_B	PWMCONC_B	8	H'FD1E	PWMU_B	8
PWM control register D_B	PWMCOND_B	8	H'FD1F	PWMU_B	8
Timer control register_1	TCR_1	8	H'FD40	TPU_1	8
Timer mode register_1	TMDR_1	8	H'FD41	TPU_1	8
Timer I/O control register_1	TIOR_1	8	H'FD42	TPU_1	8
Timer interrupt enable register_1	TIER_1	8	H'FD44	TPU_1	8
Timer status register_1	TSR_1	8	H'FD45	TPU_1	8
Timer counter _1	TCNT_1	16	H'FD46	TPU_1	16
Timer general register A_1	TGRA_1	16	H'FD48	TPU_1	16
Timer general register B_1	TGRB_1	16	H'FD4A	TPU_1	16
LPC channel 1 address register H	LADR1H	8	H'FDC0	LPC	8
LPC channel 1 address register L	LADR1L	8	H'FDC1	LPC	8
LPC channel 2 address register H	LADR2H	8	H'FDC2	LPC	8
LPC channel 2 address register L	LADR2L	8	H'FDC3	LPC	8

SERIRQ control register 2	SIRQCR2	8	H'FDDA	LPC	8
SERIRQ control register 3	SIRQCR3	8	H'FDDB	LPC	8
Port 6 noise canceler enable register	P6NCE	8	H'FE00 (PORTS = 0)	PORT	8
Port 6 noise canceler decision control register	P6NCMC	8	H'FE01 (PORTS = 0)	PORT	8
Port 6 noise cancel cycle setting register	P6NCCS	8	H'FE02 (PORTS = 0)	PORT	8
Port C noise canceler enable register	PCNCE	8	H'FE03 (PORTS = 0)	PORT	8
Port C noise canceler decision control register	PCNCMC	8	H'FE04 (PORTS = 0)	PORT	8
Port C noise cancel cycle setting register	PCNCCS	8	H'FE05 (PORTS = 0)	PORT	8
Port G noise canceler enable register	PGNCE	8	H'FE06 (PORTS = 0)	PORT	8
Port G noise canceler decision control register	PGNCMC	8	H'FE07 (PORTS = 0)	PORT	8
Port G noise cancel cycle setting register	PGNCCS	8	H'FE08 (PORTS = 0)	PORT	8
Port H input data register	PHPIN	8	H'FE0C (Read) (PORTS = 0)	PORT	8
Port H data direction register	PHDDR	8	H'FE0C (Write) (PORTS = 0)	PORT	8
Port H output data register	PHODR	8	H'FE0D (PORTS = 0)	PORT	8
Port H Nch-OD control register	PHNOCR	8	H'FE0E (PORTS = 0)	PORT	8
Port control register 0	PTCNT0	8	H'FE10	PORT	8

			(PORTS = 0)		
Port D Nch-OD control register	PDNOCR	8	H'FE1D (PORTS = 0)	PORT	8
Bidirectional data register 0MW	TWR0MW	8	H'FE20	LPC	8
Bidirectional data register 0SW	TWR0SW	8	H'FE20	LPC	8
Bidirectional data register 1	TWR1	8	H'FE21	LPC	8
Bidirectional data register 2	TWR2	8	H'FE22	LPC	8
Bidirectional data register 3	TWR3	8	H'FE23	LPC	8
Bidirectional data register 4	TWR4	8	H'FE24	LPC	8
Bidirectional data register 5	TWR5	8	H'FE25	LPC	8
Bidirectional data register 6	TWR6	8	H'FE26	LPC	8
Bidirectional data register 7	TWR7	8	H'FE27	LPC	8
Bidirectional data register 8	TWR8	8	H'FE28	LPC	8
Bidirectional data register 9	TWR9	8	H'FE29	LPC	8
Bidirectional data register 10	TWR10	8	H'FE2A	LPC	8
Bidirectional data register 11	TWR11	8	H'FE2B	LPC	8
Bidirectional data register 12	TWR12	8	H'FE2C	LPC	8
Bidirectional data register 13	TWR13	8	H'FE2D	LPC	8
Bidirectional data register 14	TWR14	8	H'FE2E	LPC	8
Bidirectional data register 15	TWR15	8	H'FE2F	LPC	8
Input data register 3	IDR3	8	H'FE30	LPC	8
Output data register 3	ODR3	8	H'FE31	LPC	8
Status register 3	STR3	8	H'FE32	LPC	8
Host interface control register 5	HICR5	8	H'FE33	LPC	8
LPC channel 3 address register H	LADR3H	8	H'FE34	LPC	8

Output data register 2	ODR2	8	H'FE3D	LPC	8
Status register 2	STR2	8	H'FE3E	LPC	8
Host interface select register	HISEL	8	H'FE3F	LPC	8
Host interface control register 0	HICR0	8	H'FE40	LPC	8
Host interface control register 1	HICR1	8	H'FE41	LPC	8
Host interface control register 2	HICR2	8	H'FE42	LPC	8
Host interface control register 3	HICR3	8	H'FE43	LPC	8
Wakeup event interrupt mask register	WUEMR	8	H'FE45	INT	8
Port G output data register	PGODR	8	H'FE46 (PORTS = 0)	PORT	8
Port G input data register	PGPIN	8	H'FE47 (Read) (PORTS = 0)	PORT	8
Port G data direction register	PGDDR	8	H'FE47 (Write) (PORTS = 0)	PORT	8
Port F output data register	PFODR	8	H'FE49 (PORTS = 0)	PORT	8
Port E input data register	PEPIN	8	H'FE4A (Read) (write prohibited) (PORTS = 0)	PORT	8
Port F input data register	PFPIN	8	H'FE4B (Read) (PORTS = 0)	PORT	8
Port F data direction register	PFDDR	8	H'FE4B (Write) (PORTS = 0)	PORT	8
Port C output data register	PCODR	8	H'FE4C (PORTS = 0)	PORT	8
Port D output data register	PDODR	8	H'FE4D (PORTS = 0)	PORT	8

Timer mode register_0	TMDR_0	8	H'FE51	TPU_0	8
Timer I/O control register H_0	TIORH_0	8	H'FE52	TPU_0	8
Timer I/O control register L_0	TIORL_0	8	H'FE53	TPU_0	8
Timer interrupt enable register_0	TIER_0	8	H'FE54	TPU_0	8
Timer status register_0	TSR_0	8	H'FE55	TPU_0	8
Timer counter _0	TCNT_0	16	H'FE56	TPU_0	16
Timer general register A_0	TGRA_0	16	H'FE58	TPU_0	16
Timer general register B_0	TGRB_0	16	H'FE5A	TPU_0	16
Timer general register C_0	TGRC_0	16	H'FE5C	TPU_0	16
Timer general register D_0	TGRD_0	16	H'FE5E	TPU_0	16
Timer control register_2	TCR_2	8	H'FE70	TPU_2	8
Timer mode register_2	TMDR_2	8	H'FE71	TPU_2	8
Timer I/O control register_2	TIOR_2	8	H'FE72	TPU_2	8
Timer interrupt enable register_2	TIER_2	8	H'FE74	TPU_2	8
Timer status register_2	TSR_2	8	H'FE75	TPU_2	8
Timer counter _2	TCNT_2	16	H'FE76	TPU_2	16
Timer general register A_2	TGRA_2	16	H'FE78	TPU_2	16
Timer general register B_2	TGRB_2	16	H'FE7A	TPU_2	16
System control register 3	SYSCR3	8	H'FE7D	SYSTEM	8
Module stop control register A	MSTPCRA	8	H'FE7E	SYSTEM	8
Module stop control register B	MSTPCRB	8	H'FE7F	SYSTEM	8
Keyboard matrix interrupt register	KMIMR	8	H'FE81 (RELOCATE = 1)	INT	8
Pull-up MOS control register	KMPCR	8	H'FE82 (RELOCATE = 1)	PORT	8

I^{2}C bus status register_2	ICSR_2	8	H'FE89	IIC_2	8
I^{2}C bus control Initialization register_2	ICRES_2	8	H'FE8A	IIC_2	8
I^{2}C bus control extended register_2	ICXR_2	8	H'FE8C	IIC_2	8
I^{2}C bus data register_2	ICDR_2	8	H'FE8E	IIC_2	8
Second slave address register_2	SARX_2	8	H'FE8E	IIC_2	8
I^{2}C bus mode register_2	ICMR_2	8	H'FE8F	IIC_2	8
Slave address register_2	SAR_2	8	H'FE8F	IIC_2	8
PWMX(D/A) control register	DACR	8	H'FEA0 (RELOCATE = 1)	PWMX	8
PWMX(D/A) data register AH	DADRAH	8	H'FEA0 (RELOCATE = 1)	PWMX	8
PWMX(D/A) data register AL	DADRAL	8	H'FEA1 (RELOCATE = 1)	PWMX	8
PWMX(D/A) data register BH	DADRBH	8	H'FEA6 (RELOCATE = 1)	PWMX	8
PWMX(D/A) counter H	DACNTH	8	H'FEA6 (RELOCATE = 1)	PWMX	8
PWMX(D/A) data register BL	DADRBL	8	H'FEA7 (RELOCATE = 1)	PWMX	8
PWMX(D/A) counter L	DACNTL	8	H'FEA7 (RELOCATE = 1)	PWMX	8
Flash code control status register	FCCS	8	H'FEA8	ROM	8
Flash program code select register	FPCS	8	H'FEA9	ROM	8
Flash erace code select register	FECS	8	H'FEAA	ROM	8
Flash key code register	FKEY	8	H'FEAC	ROM	8
Flash MAT select register	FMATS	8	H'FEAD	ROM	8

Keyboard control register 1_1	KBCR1_1	8	H'FEC2	PS2_1	8
Keyboard data buffer transmit data register_1	KBTR_1	8	H'FEC3	PS2_1	8
Keyboard control register 1_2	KBCR1_2	8	H'FEC4	PS2_2	8
Keyboard data buffer transmit data register_2	KBTR_2	8	H'FEC5	PS2_2	8
Timer XY control register	TCRXY	8	H'FEC6	TMR_XY	8
Timer control register_Y	TCR_Y	8	H'FEC8 (RELOCATE = 1)	TMR_Y	8
Timer control/status register_Y	TCSR_Y	8	H'FEC9 (RELOCATE = 1)	TMR_Y	8
Time constant register A_Y	TCORA_Y	8	H'FECA (RELOCATE = 1)	TMR_Y	8
Time constant register B_Y	TCORB_Y	8	H'FECB (RELOCATE = 1)	TMR_Y	8
Timer counter _Y	TCNT_Y	8	H'FECC (RELOCATE = 1)	TMR_Y	8
I^{2}C bus data register_1	ICDR_1	8	H'FECE (RELOCATE = 1)	IIC_1	8
Second slave address register_1	SARX_1	8	H'FECE (RELOCATE = 1)	IIC_1	8
I^{2}C bus mode register_1	ICMR_1	8	H'FECF (RELOCATE = 1)	IIC_1	8
Slave address register_1	SAR_1	8	H'FECF (RELOCATE = 1)	IIC_1	8
I^{2}C bus control register_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	IIC_1	8
I^{2}C bus status register_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	IIC_1	8

Serial mode register_1	SMR_1	8	H'FF88	SCI_1	8
I^{2}C bus control register_1	ICCR_1	8	H'FF88 (RELOCATE = 0)	IIC_1	8
Bit rate register_1	BRR_1	8	H'FF89	SCI_1	8
I^{2}C bus status register_1	ICSR_1	8	H'FF89 (RELOCATE = 0)	IIC_1	8
Serial control register_1	SCR_1	8	H'FF8A	SCI_1	8
Transmit data register_1	TDR_1	8	H'FF8B	SCI_1	8
Serial status register_1	SSR_1	8	H'FF8C	SCI_1	8
Receive data register_1	RDR_1	8	H'FF8D	SCI_1	8
Smart card mode register_1	SCMR_1	8	H'FF8E	SCI_1	8
I^{2}C bus data register_1	ICDR_1	8	H'FF8E (RELOCATE = 0)	IIC_1	8
Second slave address register_1	SARX_1	8	H'FF8E (RELOCATE = 0)	IIC_1	8
I^{2}C bus mode register_1	ICMR_1	8	H'FF8F (RELOCATE = 0)	IIC_1	8
Slave address register_1	SAR_1	8	H'FF8F (RELOCATE = 0)	IIC_1	8
PWMX(D/A) control register	DACR	8	H'FFA0 (RELOCATE = 0)	PWMX	8
PWMX(D/A) data register AH	DADRAH	8	H'FFA0 (RELOCATE = 0)	PWMX	8
Serial mode register_2	SMR_2	8	H'FFA0	SCI_2	8
PWMX(D/A) data register AL	DADRAL	8	H'FFA1 (RELOCATE = 0)	PWMX	8
Bit rate register_2	BRR_2	8	H'FFA1	SCI_2	8
Serial control register_2	SCR_2	8	H'FFA2	SCI_2	8

PWMX(D/A) counter L	DACNTL	8	H'FFA7 (RELOCATE = 0)	PWMX	8
PWMX(D/A) data register BL	DADRBL	8	H'FFA7 (RELOCATE = 0)	PWMX	8
Timer control/status register_0	TCSR_0	8	H'FFA8 (Write)	WDT_0	16
Timer control/status register_0	TCSR_0	8	H'FFA8 (Read)	WDT_0	8
Timer counter _0	TCNT_0	8	H'FFA8 (Write)	WDT_0	16
Timer counter _0	TCNT_0	8	H'FFA9 (Read)	WDT_0	8
Port A output data register	PAODR	8	H'FFAA (PORTS = 0)	PORT	8
Port A input data register	PAPIN	8	H'FFAB (Read) (PORTS = 0)	PORT	8
Port A data direction register	PADDR	8	H'FFAB (Write) (PORTS = 0)	PORT	8
Port 1 pull-up MOS control register	P1PCR	8	H'FFAC (PORTS = 0)	PORT	8
Port 2 pull-up MOS control register	P2PCR	8	H'FFAD (PORTS = 0)	PORT	8
Port 3 pull-up MOS control register	P3PCR	8	H'FFAE (PORTS = 0)	PORT	8
Port 1 data direction register	P1DDR	8	H'FFB0 (PORTS = 0)	PORT	8
Port 2 data direction register	P2DDR	8	H'FFB1 (PORTS = 0)	PORT	8
Port 1 data register	P1DR	8	H'FFB2 (PORTS = 0)	PORT	8
Port 2 data register	P2DR	8	H'FFB3 (PORTS = 0)	PORT	8

			(PORTS = 0)		
Port 6 data direction register	P6DDR	8	H'FFB9 (PORTS = 0)	PORT	8
Port 5 data register	P5DR	8	H'FFBA (PORTS = 0)	PORT	8
Port 6 data register	P6DR	8	H'FFBB (PORTS = 0)	PORT	8
Port B output data register	PBODR	8	H'FFBC (PORTS = 0)	PORT	8
Port 8 data direction register	P8DDR	8	H'FFBD (Write) (PORTS = 0)	PORT	8
Port B input data register	PBPIN	8	H'FFBD (Read) (PORTS = 0)	PORT	8
Port 7 input data register	P7PIN	8	H'FFBE (Read) (PORTS = 0)	PORT	8
Port B data direction register	PBDDR	8	H'FFBE (Write) (PORTS = 0)	PORT	8
Port 8 data register	P8DR	8	H'FFBF (PORTS = 0)	PORT	8
Port 9 data direction register	P9DDR	8	H'FFC0 (PORTS = 0)	PORT	8
Port 9 data register	P9DR	8	H'FFC1 (PORTS = 0)	PORT	8
Interrupt enable register	IER	8	H'FFC2	INT	8
Serial timer control register	STCR	8	H'FFC3	SYSTEM	8
System control register	SYSCR	8	H'FFC4	SYSTEM	8
Mode control register	MDCR	8	H'FFC5	SYSTEM	8
Bus control register	BCR	8	H'FFC6	BSC	8
Wait state control register	WSCR	8	H'FFC7	BSC	8

Timer counter _0	TCNT_0	8	H'FFD0	TMR_0	16
Timer counter _1	TCNT_1	8	H'FFD1	TMR_1	16
I^{2}C bus control register_0	ICCR_0	8	H'FFD8	IIC_0	8
I^{2}C bus status register_0	ICSR_0	8	H'FFD9	IIC_0	8
I^{2}C bus data register_0	ICDR_0	8	H'FFDE	IIC_0	8
Second slave address register_0	SARX_0	8	H'FFDE	IIC_0	8
I^{2}C bus mode register_0	ICMR_0	8	H'FFDF	IIC_0	8
Slave address register_0	SAR_0	8	H'FFDF	IIC_0	8
Keyboard control register H_3	KBCRH_3	8	H'FFE0	PS2_3	8
Keyboard control register L_3	KBCRL_3	8	H'FFE1	PS2_3	8
Keyboard data buffer register_3	KBBR_3	8	H'FFE2	PS2_3	8
Keyboard control register 2_3	KBCR2_3	8	H'FFE3	PS2_3	8
Timer control/status register	TCSR_1	8	H'FFEA (Write)	WDT_1	16
Timer control/status register	TCSR_1	8	H'FFEA (Read)	WDT_1	8
Timer counter _1	TCNT_1	8	H'FFEA (Write)	WDT_1	16
Timer counter _1	TCNT_1	8	H'FFEB (Read)	WDT_1	8
Timer control register_X	TCR_X	8	H'FFF0	TMR_X	8
Timer control register_Y	TCR_Y	8	H'FFF0 (RELOCATE = 0)	TMR_Y	8
Keyboard matrix interrupt register	KMIMR	8	H'FFF1 (RELOCATE = 0)	INT	8
Timer control/status register_X	TCSR_X	8	H'FFF1	TMR_X	8
Timer control/status register_Y	TCSR_Y	8	H'FFF1 (RELOCATE = 0)	TMR_Y	8
Pull-up MOS control register	KMPCR	8	H'FFF2 (RELOCATE = 0)	PORT	8

Timer counter _Y	TCNT_Y	8	H'FFF4 (RELOCATE = 0)	TMR_Y	8
Time constant register C	TCORC	8	H'FFF5	TMR_X	8
Time constant register A_X	TCORA_X	8	H'FFF6	TMR_X	8
Time constant register B_X	TCORB_X	8	H'FFF7	TMR_X	8
Timer connection register I	TCONRI	8	H'FFFC	TMR_X	8
Timer connection register S	TCONRS	8	H'FFFE	TMR_X, TMR_Y	8

PIODR	PI7ODR	PI6ODR	PI5ODR	PI4ODR	PI3ODR	PI2ODR	PI1ODR	PI0ODR
PJODR	PJ7ODR	PJ6ODR	PJ5ODR	PJ4ODR	PJ3ODR	PJ2ODR	PJ1ODR	PJ0ODR
PIPIN	PI7PIN	PI6PIN	PI5PIN	PI4PIN	PI3PIN	PI2PIN	PI1PIN	PI0PIN
PJPIN	PJ7PIN	PJ6PIN	PJ5PIN	PJ4PIN	PJ3PIN	PJ2PIN	PJ1PIN	PJ0PIN
PJPCR	PJ7PCR	PJ6PCR	PJ5PCR	PJ4PCR	PJ3PCR	PJ2PCR	PJ1PCR	PJ0PCR
PINOCR	PI7NOCR	PI6NOCR	PI5NOCR	PI4NOCR	PI3NOCR	PI2NOCR	PI1NOCR	PI0NOCR
PJNOCR	PJ7NOCR	PJ6NOCR	PJ5NOCR	PJ4NOCR	PJ3NOCR	PJ2NOCR	PJ1NOCR	PJ0NOCR
CCR1	CIRE	SRES	CPHS	MLS	REPRCVE	—	CLK1	CLK0
CCR2	TFM1	TFM0	—	—	—	—	—	—
CSTR	CIRBUSY	CIRRDRF	REPF	OVRF	REND	ABF	FRF	HEADF
CEIR	—	—	REPIE	OVEIE	RENDIE	ABIE	FREIE	HEADFIE
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
CIRRDR0 to 7	CIRRDR7	CIRRDR6	CIRRDR5	CIRRDR4	CIRRDR3	CIRRDR2	CIRRDR1	CIRRDR0
HHMIN	RFMBIN4	RFMBIN3	RFMBIN2	RFMBIN1	RFMBIN0	—	HHMIN9	HHMIN8
	HHMIN7	HHMIN6	HHMIN5	HHMIN4	HHMIN3	HHMIN2	HHMIN1	HHMIN0
HHMAX	FLT1	FLT0	FLTE	FLTCK1	FLTCK0	—	HHMAX9	HHMAX8
	HHMAX7	HHMAX6	HHMAX5	HHMAX4	HHMAX3	HHMAX2	HHMAX1	HHMAX0
HLMIN	HLMIN7	HLMIN6	HLMIN5	HLMIN4	HLMIN3	HLMIN2	HLMIN1	HLMIN0
HLMAX	HLMAX7	HLMAX6	HLMAX5	HLMAX4	HLMAX3	HLMAX2	HLMAX1	HLMAX0
DT1MIN	DT1MIN7	DT1MIN6	DT1MIN5	DT1MIN4	DT1MIN3	DT1MIN2	DT1MIN1	DT1MIN0
DT1MAX	DT1MAX7	DT1MAX6	DT1MAX5	DT1MAX4	DT1MAX3	DT1MAX2	DT1MAX1	DT1MAX0
DT0MIN	DT0MIN7	DT0MIN6	DT0MIN5	DT0MIN4	DT0MIN3	DT0MIN2	DT0MIN1	DT0MIN0
DT0MAX	DT0MAX7	DT0MAX6	DT0MAX5	DT0MAX4	DT0MAX3	DT0MAX2	DT0MAX1	DT0MAX0
RMIN	RMIN7	RMIN6	RMIN5	RMIN4	RMIN3	RMIN2	RMIN1	RMIN0
RMAX	RMAX7	RMAX6	RMAX5	RMAX4	RMAX3	RMAX2	RMAX1	RMAX0

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPICR_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPICRF_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPCSR_0	OVF	TWDMXOVF	TWDMNUDF	TPDMXOVF	ICPF	CMF	CKSEG	TPDMNU
TDPCR1_0	CST	POCTL	CPSPE	IEDG	TDPMDS	CKS2	CKS1	CKS0
TDPIER_0	OVIE	TWDMXIE	TWDMNIE	TPDMXIE	ICPIE	CMIE	TDPIPE	TPDMN
TDPCR2_0	PMMS	MCICTL	—	—	—	—	—	—
TDPWDMN_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPCNT_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPPDMX_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPPDMN_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPWDMX_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPICR_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPICRF_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPPDMX_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPPDMN_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPWDMX_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPICR_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPICRF_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TDPCSR_2	OVF	TWDMXOVF	TWDMNUDF	TPDMXOVF	ICPF	CMF	CKSEG	TPDMNUDF
TDPCR1_2	CST	POCTL	CPSPE	IEDG	TDPMDS	CKS2	CKS1	CKS0
TDPIER_2	OVIE	TWDMXIE	TWDMNIE	TPDMXIE	ICPIE	CMIE	TDPIPE	TPDMNIE
TDPCR2_2	PMMS	MCICTL	—	—	—	—	—	—
TDPWDMN_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMCNT_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMMLCM_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMICR_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMICRF_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

TCMMLCM_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMICR_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMICRF_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMCSR_1	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—
TCMCR_1	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0
TCMIER_1	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—
TCMMINCM_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMCNT_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMMLCM_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMICR_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMICRF_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMCSR_2	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—
TCMCR_2	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0
TCMIER_2	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—
TCMMINCM_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCMCSR_3	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—
TCMCR_3	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0
TCMIER_3	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—
TCMMINCM_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRA	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRB	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRC	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRD	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRE	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRF	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRG	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADDRH	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	—

FLCR	DLAB	BREAK	STICKPARITY	EPS	PEN	STOP	CLS1	CLS0
FMCR	—	—	—	LOOPBACK	OUT2	OUT1	RTS	DTR
FLSR	RXFIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
FMSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
FSCR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCIFCR	SCIFOE1	SCIFOE0	—	OUT2LOOP	CKSEL1	CKSEL0	SCIFRST	REGRST
FSIHBARH	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
FSIHBARL	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
FSISR	—	—	—	—	—	—	FSIMS1	FSIMS0
CMDHBARH	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
CMDHBARL	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
FSICMDR	bit7	bt6	bit5	bit4	bit3	bit2	bit1	bit0
FSILSTR1	CMDBUSY	FSICMDI	FSIDMYE	FSIWBUS	YFSIWI	LFBUSY	—	—
FSILSTR2	—	—	—	FSIDWBUSY	FSIDRBUSY	SIZE2	SIZE1	SIZE0
FSIGPR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR7	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPR9	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPRA	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIGPRB	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

FSIARL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIWDRHH	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
FSIWDRHL	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
FSIWDRLH	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
FSIWDRLL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSICR1	SRES	FSIE	FRDE	AAIE	CPHS	CPOS	—	CKSEL
FSICR2	TE	RE	FSITEIE	FSIRXIE	—	—	—	—
FSIBNR	TBN3	TBN2	TBN1	TBN0	—	RBN2	RBN1	RBN0
FSINS	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIRDINS	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIPPINS	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSISTR	FSITEI	OBF	FSIRXI	—	—	—	—	—
FSITDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSITDR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSITDR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSITDR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSITDR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSITDR5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSITDR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSITDR7	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FSIRDR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

PWMPRE3_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMREG4_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE4_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMREG5_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE5_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMCONA_A	CLK1	CLK0	—	—	—	—	—	—
PWMCONB_A	—	—	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E
PWMCONC_A	—	CNTMD01	PWMSL5	PWMSL4	PWMSL3	PWMSL2	PWMSL1	PWMSL
PWMCOND_A	PH5S	PH4S	PH3S	PH2S	PH1S	PH0S	CNTMD45	CNTMD
PWMREG0_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE0_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMREG1_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE1_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMREG2_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE2_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMREG3_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE3_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMREG4_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE4_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMREG5_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMPRE5_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PWMCONA_B	CLK1	CLK0	—	—	—	—	—	—
PWMCONB_B	—	—	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E
PWMCONC_B	—	CNTMD01	PWMSL5	PWMSL4	PWMSL3	PWMSL2	PWMSL1	PWMSL
PWMCOND_B	PH5S	PH4S	PH3S	PH2S	PH1S	PH0S	CNTMD45	CNTMD

TGRA_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TGRB_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LADR1H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
LADR1L	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LADR2H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
LADR2L	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCIFADRH	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
SCIFADRL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LADR4H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
LADR4L	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
IDR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR4	DBU47	DBU46	DBU45	DBU44	C/$\overline{\text{D}}$4	DBU42	IBF4	OBF4
HICR4	—	LPC4E	IBFIE4	—	—	—	—	—
SIRQCR2	IEDIR3	IEDIR4	IRQ11E4	IRQ10E4	IRQ9E4	IRQ6E4	SMIE4	—
SIRQCR3	SELIRQ15	SELIRQ14	SELIRQ13	SELIRQ8	SELIRQ7	SELIRQ5	SELIRQ4	SELIRQ3
P6NCE	P67NCE	P66NCE	P65NCE	P64NCE	P63NCE	P62NCE	P61NCE	P60NCE
P6NCMC	P67NCMC	P66NCMC	P65NCMC	P64NCMC	P63NCMC	P62NCMC	P61NCMC	P60NCMC
P6NCCS	—	—	—	—	—	P6NCCK2	P6NCCK1	P6NCCK0
PCNCE	PC7NCE	PC6NCE	PC5NCE	PC4NCE	PC3NCE	PC2NCE	PC1NCE	PC0NCE
PCNCMC	PC7NCMC	PC6NCMC	PC5NCMC	PC4NCMC	PC3NCMC	PC2NCMC	PC1NCMC	PC0NCMC
PCNCCS	—	—	—	—	—	PCNCCK2	PCNCCK1	PCNCCK0
PGNCE	PG7NCE	PG6NCE	PG5NCE	PG4NCE	PG3NCE	PG2NCE	PG1NCE	PG0NCE

PTCNT1	IIC1BS	IIC1AS	—	—	IIC0BS	IIC0AS	—	—
PTCNT2	—	TxD2RS	RxD2RS	TxD1RS	RxD1RS	—	PORTS	—
P9PCR	—	—	P95PCR	P94PCR	P93PCR	P92PCR	P91PCR	P90PCR
PGNOCR	PG7NOCR	PG6NOCR	PG5NOCR	PG4NOCR	PG3NOCR	PG2NOCR	PG1NOCR	PG0NOC
PFNOCR	PF7NOCR	PF6NOCR	PF5NOCR	PF4NOCR	PF3NOCR	PF2NOCR	PF1NOCR	PF0NOC
PCNOCR	PC7NOCR	PC6NOCR	PC5NOCR	PC4NOCR	PC3NOCR	PC2NOCR	PC1NOCR	PC0NOC
PDNOCR	PD7NOCR	PD6NOCR	PD5NOCR	PD4NOCR	PD3NOCR	PD2NOCR	PD1NOCR	PD0NOC
TWR0MW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR0SW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR5	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR6	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR7	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR9	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR10	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR11	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR12	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR13	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR14	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR15	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
IDR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

SIRQCR1	IRQ11E3	IRQ10E3	IRQ9E3	IRQ6E3	IRQ11E2	IRQ10E2	IRQ9E2	IRQ6E2
IDR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR1	DBU17	DBU16	DBU15	DBU14	C/$\overline{\text{D}}$1	DBU12	IBF1	OBF1
SIRQCR4	—	—	—	—	SCSIRQ3	SCSIRQ2	SCSIRQ1	SCSIRQ0
IDR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR2	DBU27	DBU26	DBU25	DBU24	C/$\overline{\text{D}}$2	DBU22	IBF2	OBF2
HISEL	SELSTR3	SELIRQ11	SELIRQ10	SELIRQ9	SELIRQ6	SELSMI	SELIRQ12	SELIRQ1
HICR0	LPC3E	LPC2E	LPC1E	FGA20E	SDWNE	PMEE	LSMIE	LSCIE
HICR1	LPCBSY	CLKREQ	IRQBSY	LRSTB	SDWNB	PMEB	LSMIB	LSCIB
HICR2	GA20	LRST	SDWN	ABRT	IBFIE3	IBFIE2	IBFIE1	ERRIE
HICR3	LFRAME	CLKRUN	SERIRQ	LRESET	LPCPD	PME	LSMI	LSCI
WUEMR	WUEMR15	WUEMR14	WUEMR13	WUEMR12	WUEMR11	WUEMR10	WUEMR9	WUEMR8
PGODR	PG7ODR	PG6ODR	PG5ODR	PG4ODR	PG3ODR	PG2ODR	PG1ODR	PG0ODR
PGPIN	PG7PIN	PG6PIN	PG5PIN	PG4PIN	PG3PIN	PG2PIN	PG1PIN	PG0PIN
PGDDR	PG7DDR	PG6DDR	PG5DDR	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
PFODR	PF7ODR	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODR
PEPIN	—	—	—	PE4PIN	PE3PIN	PE2PIN	PE1PIN	PE0PIN
PFPIN	PF7PIN	PF6PIN	PF5PIN	PF4PIN	PF3PIN	PF2PIN	PF1PIN	PF0PIN
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR
PDODR	PD7ODR	PD6ODR	PD5ODR	PD4ODR	PD3ODR	PD2ODR	PD1ODR	PD0ODR
PCPIN	PC7PIN	PC6PIN	PC5PIN	PC4PIN	PC3PIN	PC2PIN	PC1PIN	PC0PIN
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR

TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TGRA_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TGRB_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TGRC_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TGRD_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TGRA_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TGRB_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SYSCR3	—	EIVS	RELOCATE	—	—	—	—	—
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0

ICCR_2	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_2	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICRES_2	—	—	—	—	CLR3	CLR2	CLR1	CLR0
ICXR_2	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
SARX_2	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICDR_2	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
ICMR_2	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
DACR	—	PWME	—	—	OEB	OEA	OS	CKS
DADRA	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
DADRB	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
DACNTH	DACNT7	DACNT6	DACNT5	DACNT4	DACNT3	DACNT2	DACNT1	DACNT0
DACNTL	DACNT8	DACNT9	DACNT10	DACNT11	DACNT12	DACNT13	—	REGS
FCCS	—	—	—	FLER	—	—	—	SCO
FPCS	—	—	—	—	—	—	—	PPVS
FECS	—	—	—	—	—	—	—	EPVB
FKEY	K7	K6	K5	K4	K3	K2	K1	K0
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0

KBTR_2	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0
TCRXY	—	—	CKSX	CKSY	—	—	—	—
TCR_Y	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_Y	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0
TCORA_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCNT_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ICDR_1	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
SARX_1	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
KBCR1_3	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER
KBTR_3	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0
ICXR_0	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICXR_1	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
KBCRH_0	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS
KBCRL_0	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0
KBBR_0	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0
KBCR2_0	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0
KBCRH_1	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS
KBCRL_1	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0
KBBR_1	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0
KBCR2_1	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0

ICRC	ICRC7	ICRC6	ICRC5	ICRC4	ICRC3	ICRC2	ICRC1	ICRC0
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
ABRKCR	CMF	—	—	—	—	—	—	BIE
BARA	A23	A22	A21	A20	A19	A18	A17	A16
BARB	A15	A14	A13	A12	A11	A10	A9	A8
BARC	A7	A6	A5	A4	A3	A2	A1	—
IER16	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E
ISR16	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
ISCR16H	IRQ15SCB	IRQ15SCA	IRQ14SCB	IRQ14SCA	IRQ13SCB	IRQ13SCA	IRQ12SCB	IRQ12SC
ISCR16L	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SCA
ISSR16	ISS15	ISS14	ISS13	ISS12	ISS11	ISS10	ISS9	ISS8
ISSR	ISS7	—	—	—	—	—	—	—
PCSR	—	—	PWCKXB	PWCKXA	—	—	—	PWCKXC
SBYCR	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0
LPWRCR	DTON	LSON	NESEL	EXCLE	—	—	—	—
MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
SMR_1*[1]	C/$\overline{\text{A}}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{\text{E}}$ (O/$\overline{\text{E}}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
BRR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

TDR_2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SSR_2*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF
TCSR_0	OVF	WT/$\overline{\text{IT}}$	TME	—	RST/$\overline{\text{NMI}}$	CKS2	CKS1	CKS0
TCNT_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR
PAPIN	PA7PIN	PA6PIN	PA5PIN	PA4PIN	PA3PIN	PA2PIN	PA1PIN	PA0PIN
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
P1PCR	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR
P2PCR	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR
P3PCR	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR
P5DDR	—	—	—	—	—	P52DDR	P51DDR	P50DDR
P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
P5DR	—	—	—	—	—	P52DR	P51DR	P50DR
P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR

P9DR	P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	P91DR	P90DR
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
STCR	IICX2	IICX1	IICX0	IICE	FLSHE	IICS	ICKS1	ICKS0
SYSCR	—	—	INTM1	INTM0	XRST	NMIEG	KINWUE	RAME
MDCR	EXPE	—	—	—	—	MDS2	MDS1	MDS0
BCR	—	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0
WSCR	—	—	ABW	AST	WMS1	WMS0	WC1	WC0
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORA_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCNT_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCNT_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_0	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
SARX_0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS

TCSR_X	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0
TICRR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TICRF	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCNT_X	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORC	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORA_X	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_X	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCONRI	—	—	—	ICST	—	—	—	—
TCONRS	TMRX/Y	—	—	—	—	—	—	—

Notes: 1. In normal mode and smart card interface mode, bit names differ in part.
( ) : Bit name in smart card interface mode.
2. When TWRE = 1 or SELSTR3 = 0.
3. When TWRE = 0 and SELSTR3 = 1.

PJPIN	—	—	—	—	—	—	
PJPCR	Initialized	—	—	—	—	—	
PINOCR	Initialized	—	—	—	—	—	
PJNOCR	Initialized	—	—	—	—	—	
CCR1	Initialized	—	—	—	—	—	C
CCR2	Initialized	—	—	—	—	—	
CSTR	Initialized	—	—	—	—	—	
CEIR	Initialized	—	—	—	—	—	
BRR	Initialized	—	—	—	—	—	
CIRRDR0 to 7	Initialized	—	—	—	—	—	
HHMIN	Initialized	—	—	—	—	—	
HHMAX	Initialized	—	—	—	—	—	
HLMIN	Initialized	—	—	—	—	—	
HLMAX	Initialized	—	—	—	—	—	
DT1MIN	Initialized	—	—	—	—	—	
DT1MAX	Initialized	—	—	—	—	—	
DT0MIN	Initialized	—	—	—	—	—	
DT0MAX	Initialized	—	—	—	—	—	
RMIN	Initialized	—	—	—	—	—	
RMAX	Initialized	—	—	—	—	—	

TDPCR1_0	Initialized	—	—	—	—	—
TDPIER_0	Initialized	—	—	—	—	—
TDPCR2_0	Initialized	—	—	—	—	—
TDPWDMN_0	Initialized	—	—	—	—	—
TDPCNT_1	Initialized	—	—	—	—	—
TDPPDMX_1	Initialized	—	—	—	—	—
TDPPDMN_1	Initialized	—	—	—	—	—
TDPWDMX_1	Initialized	—	—	—	—	—
TDPICR_1	Initialized	—	—	—	—	—
TDPICRF_1	Initialized	—	—	—	—	—
TDPCSR_1	Initialized	—	—	—	—	—
TDPCR1_1	Initialized	—	—	—	—	—
TDPIER_1	Initialized	—	—	—	—	—
TDPCR2_1	Initialized	—	—	—	—	—
TDPWDMN_1	Initialized	—	—	—	—	—
TDPCNT_2	Initialized	—	—	—	—	—
TDPPDMX_2	Initialized	—	—	—	—	—
TDPPDMN_1	Initialized	—	—	—	—	—
TDPWDMX_2	Initialized	—	—	—	—	—
TDPICR_2	Initialized	—	—	—	—	—
TDPICRF_2	Initialized	—	—	—	—	—
TDPCSR_2	Initialized	—	—	—	—	—

TCMICRF_0	Initialized	—	—	—	—	—	
TCMCSR_0	Initialized	—	—	—	—	—	
TCMCR_0	Initialized	—	—	—	—	—	
TCMIER_0	Initialized	—	—	—	—	—	
TCMMINCM_0	Initialized	—	—	—	—	—	
TCMCNT_1	Initialized	—	—	—	—	—	T
TCMMLCM_1	Initialized	—	—	—	—	—	
TCMICR_1	Initialized	—	—	—	—	—	
TCMICRF_1	Initialized	—	—	—	—	—	
TCMCSR_1	Initialized	—	—	—	—	—	
TCMCR_1	Initialized	—	—	—	—	—	
TCMIER_1	Initialized	—	—	—	—	—	
TCMMINCM_1	Initialized	—	—	—	—	—	
TCMCNT_2	Initialized	—	—	—	—	—	T
TCMMLCM_2	Initialized	—	—	—	—	—	
TCMICR_2	Initialized	—	—	—	—	—	
TCMICRF_2	Initialized	—	—	—	—	—	
TCMCSR_2	Initialized	—	—	—	—	—	
TCMCR_2	Initialized	—	—	—	—	—	
TCMIER_2	Initialized	—	—	—	—	—	
TCMMINCM_2	Initialized	—	—	—	—	—	

TCMMINCM_3	Initialized	—	—	—	—	—
ADDRA	Initialized	—	Initialized	—	Initialized	Initialized
ADDRB	Initialized	—	Initialized	—	Initialized	Initialized
ADDRC	Initialized	—	Initialized	—	Initialized	Initialized
ADDRD	Initialized	—	Initialized	—	Initialized	Initialized
ADDRE	Initialized	—	Initialized	—	Initialized	Initialized
ADDRF	Initialized	—	Initialized	—	Initialized	Initialized
ADDRG	Initialized	—	Initialized	—	Initialized	Initialized
ADDRH	Initialized	—	Initialized	—	Initialized	Initialized
ADCSR	Initialized	—	Initialized	—	Initialized	Initialized
ADCR	Initialized	—	Initialized	—	Initialized	Initialized
FRBR	Initialized	—	—	—	—	—
FTHR	—	—	—	—	—	—
FDLL	Initialized	—	—	—	—	—
FIER	Initialized	—	—	—	—	—
FDLH	Initialized	—	—	—	—	—
FIIR	Initialized	—	—	—	—	—
FFCR	Initialized	—	—	—	—	—
FLCR	Initialized	—	—	—	—	—
FMCR	Initialized	—	—	—	—	—
FLSR	Initialized	—	—	—	—	—
FMSR	—	—	—	—	—	—
FSCR	Initialized	—	—	—	—	—
SCIFCR	Initialized	—	—	—	—	—

FSILSTR2	Initialized	—	—	—	—	—
FSIGPR1	Initialized	—	—	—	—	—
FSIGPR2	Initialized	—	—	—	—	—
FSIGPR3	Initialized	—	—	—	—	—
FSIGPR4	Initialized	—	—	—	—	—
FSIGPR5	Initialized	—	—	—	—	—
FSIGPR6	Initialized	—	—	—	—	—
FSIGPR7	Initialized	—	—	—	—	—
FSIGPR8	Initialized	—	—	—	—	—
FSIGPR9	Initialized	—	—	—	—	—
FSIGPRA	Initialized	—	—	—	—	—
FSIGPRB	Initialized	—	—	—	—	—
FSIGPRC	Initialized	—	—	—	—	—
FSIGPRD	Initialized	—	—	—	—	—
FSIGPRE	Initialized	—	—	—	—	—
FSIGPRF	Initialized	—	—	—	—	—
SLCR	Initialized	—	—	—	—	—
FSIARH	Initialized	—	—	—	—	—
FSIARM	Initialized	—	—	—	—	—
FSIARL	Initialized	—	—	—	—	—
FSIWDRHH	Initialized	—	—	—	—	—
FSIWDRHL	Initialized	—	—	—	—	—

FSIPPINS	Initialized	—	—	—	—	—
FSISTR	Initialized	—	—	—	—	—
FSITDR0	Initialized	—	—	—	—	—
FSITDR1	Initialized	—	—	—	—	—
FSITDR2	Initialized	—	—	—	—	—
FSITDR3	Initialized	—	—	—	—	—
FSITDR4	Initialized	—	—	—	—	—
FSITDR5	Initialized	—	—	—	—	—
FSITDR6	Initialized	—	—	—	—	—
FSITDR7	Initialized	—	—	—	—	—
FSIRDR	Initialized	—	—	—	—	—
PWMREG0_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMPRE0_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMREG1_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMPRE1_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMREG2_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMPRE2_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMREG3_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMPRE3_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMREG4_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMPRE4_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMREG5_A	Initialized	—	Initialized	—	Initialized	Initialized
PWMPRE5_A	Initialized	—	Initialized	—	Initialized	Initialized

PWMPRE1_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG2_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE2_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG3_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE3_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG4_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE4_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG5_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE5_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCONA_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCONB_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCONC_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCOND_B	Initialized	—	Initialized	—	Initialized	Initialized	
TCR_1	Initialized	—	—	—	—	—	T
TMDR_1	Initialized	—	—	—	—	—	
TIOR_1	Initialized	—	—	—	—	—	
TIER_1	Initialized	—	—	—	—	—	
TSR_1	Initialized	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	
TGRA_1	Initialized	—	—	—	—	—	
TGRB_1	Initialized	—	—	—	—	—	

LADR4L	Initialized	—	—	—	—	—
IDR4	Initialized	—	—	—	—	—
ODR4	Initialized	—	—	—	—	—
STR4	Initialized	—	—	—	—	—
HICR4	Initialized	—	—	—	—	—
SIRQCR2	Initialized	—	—	—	—	—
SIRQCR3	Initialized	—	—	—	—	—
P6NCE	Initialized	—	—	—	—	—
P6NCMC	Initialized	—	—	—	—	—
P6NCCS	Initialized	—	—	—	—	—
PCNCE	Initialized	—	—	—	—	—
PCNCMC	Initialized	—	—	—	—	—
PCNCCS	Initialized	—	—	—	—	—
PGNCE	Initialized	—	—	—	—	—
PGNCMC	Initialized	—	—	—	—	—
PGNCCS	Initialized	—	—	—	—	—
PHPIN	Initialized	—	—	—	—	—
PHDDR	Initialized	—	—	—	—	—
PHODR	Initialized	—	—	—	—	—
PHNOCR	Initialized	—	—	—	—	—
PTCNT0	Initialized	—	—	—	—	—
PTCNT1	Initialized	—	—	—	—	—
PTCNT2	Initialized	—	—	—	—	—

TWR1	Initialized	—	—	—	—	—
TWR2	Initialized	—	—	—	—	—
TWR3	Initialized	—	—	—	—	—
TWR4	Initialized	—	—	—	—	—
TWR5	Initialized	—	—	—	—	—
TWR6	Initialized	—	—	—	—	—
TWR7	Initialized	—	—	—	—	—
TWR8	Initialized	—	—	—	—	—
TWR9	Initialized	—	—	—	—	—
TWR10	Initialized	—	—	—	—	—
TWR11	Initialized	—	—	—	—	—
TWR12	Initialized	—	—	—	—	—
TWR13	Initialized	—	—	—	—	—
TWR14	Initialized	—	—	—	—	—
TWR15	Initialized	—	—	—	—	—
IDR3	Initialized	—	—	—	—	—
ODR3	Initialized	—	—	—	—	—
STR3	Initialized	—	—	—	—	—
HICR5	Initialized	—	—	—	—	—
LADR3H	Initialized	—	—	—	—	—
LADR3L	Initialized	—	—	—	—	—
SIRQCR0	Initialized	—	—	—	—	—
SIRQCR1	Initialized	—	—	—	—	—

HISEL	Initialized	—	—	—	—	—
HICR0	Initialized	—	—	—	—	—
HICR1	Initialized	—	—	—	—	—
HICR2	—	—	—	—	—	—
HICR3	—	—	—	—	—	—
WUEMR	Initialized	—	—	—	—	—
PGODR	Initialized	—	—	—	—	—
PGPIN	—	—	—	—	—	—
PGDDR	Initialized	—	—	—	—	—
PFODR	Initialized	—	—	—	—	—
PEPIN	—	—	—	—	—	—
PFPIN	—	—	—	—	—	—
PFDDR	Initialized	—	—	—	—	—
PCODR	Initialized	—	—	—	—	—
PDODR	Initialized	—	—	—	—	—
PCPIN	—	—	—	—	—	—
PCDDR	Initialized	—	—	—	—	—
PDPIN	—	—	—	—	—	—
PDDDR	Initialized	—	—	—	—	—
TCR_0	Initialized	—	—	—	—	—
TMDR_0	Initialized	—	—	—	—	—
TIORH_0	Initialized	—	—	—	—	—
TIORL_0	Initialized	—	—	—	—	—
TIER_0	Initialized	—	—	—	—	—
TSR_0	Initialized	—	—	—	—	—
TCNT_0	Initialized	—	—	—	—	—

TIER_2	Initialized	—	—	—	—	—	
TSR_2	Initialized	—	—	—	—	—	
TCNT_2	Initialized	—	—	—	—	—	
TGRA_2	Initialized	—	—	—	—	—	
TGRB_2	Initialized	—	—	—	—	—	
SYSCR3	Initialized	—	—	—	—	—	S
MSTPCRA	Initialized	—	—	—	—	—	
MSTPCRB	Initialized	—	—	—	—	—	
KMIMR	Initialized	—	—	—	—	—	IN
KMPCR	Initialized	—	—	—	—	—	P
KMIMRA	Initialized	—	—	—	—	—	IN
WUESCR	Initialized	—	—	—	—	—	
WUESR	Initialized	—	—	—	—	—	
WER	Initialized	—	—	—	—	—	
ICRD	Initialized	—	—	—	—	—	
ICCR_2	Initialized	—	—	—	—	—	II
ICSR_2	Initialized	—	—	—	—	—	
ICRES_2	Initialized	—	—	—	—	—	
ICXR_2	Initialized	—	—	—	—	—	
ICDR_2	—	—	—	—	—	—	
SARX_2	Initialized	—	—	—	—	—	
ICMR_2	Initialized	—	—	—	—	—	
SAR_2	Initialized	—	—	—	—	—	

FKEY	Initialized	—	—	—	—	—
FMATS	Initialized	—	—	—	—	—
FTDAR	Initialized	—	—	—	—	—
TSTR	Initialized	—	—	—	—	—
TSYR	Initialized	—	—	—	—	—
KBCR1_0	Initialized	—	—	—	—	—
KBTR_0	Initialized	—	—	—	—	—
KBCR1_1	Initialized	—	—	—	—	—
KBTR_1	Initialized	—	—	—	—	—
KBCR1_2	Initialized	—	—	—	—	—
KBTR_2	Initialized	—	—	—	—	—
TCRXY	Initialized	—	—	—	—	—
TCR_Y	Initialized	—	—	—	—	—
TCSR_Y	Initialized	—	—	—	—	—
TCORA_Y	Initialized	—	—	—	—	—
TCORB_Y	Initialized	—	—	—	—	—
TCNT_Y	Initialized	—	—	—	—	—
ICDR_1	—	—	—	—	—	—
SARX_1	Initialized	—	—	—	—	—
ICMR_1	Initialized	—	—	—	—	—
SAR_1	Initialized	—	—	—	—	—
ICCR_1	Initialized	—	—	—	—	—
ICSR_1	Initialized	—	—	—	—	—

KBCR2_0	Initialized	—	—	—	—	—	
KBCRH_1	Initialized	—	—	—	—	—	P
KBCRL_1	Initialized	—	—	—	—	—	
KBBR_1	Initialized	—	—	—	—	—	
KBCR2_1	Initialized	—	—	—	—	—	
KBCRH_2	Initialized	—	—	—	—	—	P
KBCRL_2	Initialized	—	—	—	—	—	
KBBR_2	Initialized	—	—	—	—	—	
KBCR2_2	Initialized	—	—	—	—	—	
ICRES_0	Initialized	—	—	—	—	—	II
ICRA	Initialized	—	—	—	—	—	IN
ICRB	Initialized	—	—	—	—	—	
ICRC	Initialized	—	—	—	—	—	
ISR	Initialized	—	—	—	—	—	
ISCRH	Initialized	—	—	—	—	—	
ISCRL	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	
BARA	Initialized	—	—	—	—	—	
BARB	Initialized	—	—	—	—	—	
BARC	Initialized	—	—	—	—	—	
IER16	Initialized	—	—	—	—	—	
ISR16	Initialized	—	—	—	—	—	
ISCR16H	Initialized	—	—	—	—	—	

MSTPCRL	Initialized	—	—	—	—	—
SMR_1	Initialized	—	—	—	—	—
BRR_1	Initialized	—	—	—	—	—
SCR_1	Initialized	—	—	—	—	—
TDR_1	Initialized	—	Initialized	—	Initialized	Initialized
SSR_1	Initialized	—	Initialized	—	Initialized	Initialized
RDR_1	Initialized	—	Initialized	—	Initialized	Initialized
SCMR_1	Initialized	—	—	—	—	—
SMR_2	Initialized	—	—	—	—	—
BRR_2	Initialized	—	—	—	—	—
SCR_2	Initialized	—	—	—	—	—
TDR_2	Initialized	—	Initialized	—	Initialized	Initialized
SSR_2	Initialized	—	Initialized	—	Initialized	Initialized
RDR_2	Initialized	—	Initialized	—	Initialized	Initialized
SCMR_2	Initialized	—	—	—	—	—
TCSR_0	Initialized	—	—	—	—	—
TCNT_0	Initialized	—	—	—	—	—
PAODR	Initialized	—	—	—	—	—
PAPIN	—	—	—	—	—	—
PADDR	Initialized	—	—	—	—	—
P1PCR	Initialized	—	—	—	—	—
P2PCR	Initialized	—	—	—	—	—
P3PCR	Initialized	—	—	—	—	—

P4DR	Initialized	—	—	—	—	—	
P5DDR	Initialized	—	—	—	—	—	
P6DDR	Initialized	—	—	—	—	—	
P5DR	Initialized	—	—	—	—	—	
P6DR	Initialized	—	—	—	—	—	
PBODR	Initialized	—	—	—	—	—	
PBPIN	—	—	—	—	—	—	
P8DDR	Initialized	—	—	—	—	—	
P7PIN	—	—	—	—	—	—	
PBDDR	Initialized	—	—	—	—	—	
P8DR	Initialized	—	—	—	—	—	
P9DDR	Initialized	—	—	—	—	—	
P9DR	Initialized	—	—	—	—	—	
IER	Initialized	—	—	—	—	—	IN
STCR	Initialized	—	—	—	—	—	S
SYSCR	Initialized	—	—	—	—	—	
MDCR	Initialized	—	—	—	—	—	
BCR	Initialized	—	—	—	—	—	B
WSCR	Initialized	—	—	—	—	—	
TCR_0	Initialized	—	—	—	—	—	T T
TCR_1	Initialized	—	—	—	—	—	
TCSR_0	Initialized	—	—	—	—	—	
TCSR_1	Initialized	—	—	—	—	—	

ICSR_0	Initialized	—	—	—	—	—
ICDR_0	—	—	—	—	—	—
SARX_0	Initialized	—	—	—	—	—
ICMR_0	Initialized	—	—	—	—	—
SAR_0	Initialized	—	—	—	—	—
KBCRH_3	Initialized	—	—	—	—	—
KBCRL_3	Initialized	—	—	—	—	—
KBBR_3	Initialized	—	—	—	—	—
KBCR2_3	Initialized	—	—	—	—	—
TCSR_1	Initialized	—	—	—	—	—
TCNT_1	Initialized	—	—	—	—	—
TCR_X	Initialized	—	—	—	—	—
TCSR_X	Initialized	—	—	—	—	—
TICRR	Initialized	—	—	—	—	—
TICRF	Initialized	—	—	—	—	—
TCNT_X	Initialized	—	—	—	—	—
TCORC	Initialized	—	—	—	—	—
TCORA_X	Initialized	—	—	—	—	—
TCORB_X	Initialized	—	—	—	—	—
TCONRI	Initialized	—	—	—	—	—
TCONRS	Initialized	—	—	—	—	—

H'F905	P2PIN (Read)		
H'F907	P2PCR		
H'F910	P3DDR		
H'F911	P4DDR		
H'F912	P3DR		
H'F913	P4DR		
H'F914	P3PIN (Read)		
H'F915	P4PIN (Read)		
H'F916	P3PCR		
H'F920	P5DDR		
H'F921	P6DDR		
H'F922	P5DR		
H'F923	P6DR		
H'F924	P5PIN (Read)		
H'F925	P6PIN (Read)		
H'F92B	P6NCE		
H'F92D	P6NCMC		
H'F92F	P6NCCS		
H'F931	P8DDR		
H'F933	P8DR		
H'F934	P7PIN (Read)		
H'F935	P8PIN (Read)		
H'F940	P9DDR		
H'F942	P9DR		

H'F955	PBPIN (Read)		
H'F957	PBPCR		
H'F960	PCDDR		
H'F961	PDDDR		
H'F962	PCODR		
H'F963	PDODR		
H'F964	PCPIN (Read)		
H'F965	PDPIN (Read)		
H'F966	PCPCR		
H'F967	PDPCR		
H'F968	PCNOCR		
H'F969	PDNOCR		
H'F96A	PCNCE		
H'F96C	PCNCMC		
H'F96E	PCNCCS		
H'F971	PFDDR		
H'F973	PFODR		
H'F974	PEPIN (Read)		
H'F975	PFPIN (Read)		
H'F977	PFPCR		
H'F979	PFNOCR		
H'F980	PGDDR		
H'F981	PHDDR		
H'F982	PGODR		

H'F98C	PGNCMC		
H'F98E	PGNCCS		
H'F990	PIDDR	No condition	
H'F991	PJDDR		
H'F992	PIODR		
H'F993	PJODR		
H'F994	PIPIN (Read)		
H'F995	PJPIN (Read)		
H'F996	PIPCR		
H'F997	PJPCR		
H'F998	PINOCR		
H'F999	PJNOCR		
H'FA40	CCR1	MSTPA3 = 0	CIR
H'FA41	CCR2		
H'FA42	CSTR		
H'FA43	CEIR		
H'FA44	BRR		
H'FA45	CIRRDR0 to 7		
H'FA46	HHMIN		
H'FA48	HHMAX		
H'FA4A	HLMIN		
H'FA4B	HLMAX		
H'FA4C	DT0MIN		
H'FA4D	DT0MAX		
H'FA4E	DT1MIN		

H'FB48	TDPICR_0		
H'FB4A	TDPICRF_0		
H'FB4C	TDPCSR_0		
H'FB4D	TDPCR1_0		
H'FB4E	TDPIER_0		
H'FB4F	TDPCR2_0		
H'FB50	TDPPDMN_0		
H'FB60	TDPCNT_1	MSTPA5 = 0	TDP_1
H'FB62	TDPWDMX_1		
H'FB64	TDPWDMN_1		
H'FB66	TDPPDMX_1		
H'FB68	TDPICR_1		
H'FB6A	TDPICRF_1		
H'FB6C	TDPCSR_1		
H'FB6D	TDPCR1_1		
H'FB6E	TDPIER_1		
H'FB6F	TDPCR2_1		
H'FB70	TDPPDMN_1		
H'FB80	TDPCNT_1	MSTPA4 = 0	TDP_2
H'FB82	TDPWDMX_1		
H'FB84	TDPWDMN_1		
H'FB86	TDPPDMX_1		
H'FB88	TDPICR_1		

H'FBC2	TCMMLCM_0		
H'FBC4	TCMICR_0		
H'FBC6	TCMICRF_0		
H'FBC8	TCMCSR_0		
H'FBC9	TCMCR_0		
H'FBCA	TCMIER_0		
H'FBCC	TCMMINCM_0		
H'FBD0	TCMCNT_1	MSTPB1 = 0	TCM_1
H'FBD2	TCMMLCM_1		
H'FBD4	TCMICR_1		
H'FBD6	TCMICRF_1		
H'FBD8	TCMCSR_1		
H'FBD9	TCMCR_1		
H'FBDA	TCMIER_1		
H'FBDC	TCMMINCM_1		
H'FBE0	TCMCNT_2	MSTPB2 = 0	TCM_2
H'FBE2	TCMMLCM_2		
H'FBE4	TCMICR_2		
H'FBE6	TCMICRF_2		
H'FBE8	TCMCSR_2		
H'FBE9	TCMCR_2		
H'FBEA	TCMIER_2		
H'FBEC	TCMMINCM_2		

H'FBFC	TCMMINCM_3		
H'FC00	ADDRA	MSTP9 = 0	A/D converter
H'FC02	ADDRB		
H'FC04	ADDRC		
H'FC06	ADDRD		
H'FC08	ADDRE		
H'FC0A	ADDRF		
H'FC0C	ADDRG		
H'FC0E	ADDRH		
H'FC10	ADCSR		
H'FC11	ADCR		
H'FC20	FRBR	MSTPB3 = 0	SCIF
H'FC20	FTHR		
H'FC20	FDLL		
H'FC21	FIER		
H'FC21	FDLH		
H'FC22	FIIR		
H'FC22	FFCR		
H'FC23	FLCR		
H'FC24	FMCR		
H'FC25	FLSR		
H'FC26	FMSR		
H'FC27	FSCR		
H'FC28	SCIFCR		

H'FC57	FSIGPR1		
H'FC58	FSIGPR2		
H'FC59	FSIGPR3		
H'FC5A	FSIGPR4		
H'FC5B	FSIGPR5		
H'FC5C	FSIGPR6		
H'FC5D	FSIGPR7		
H'FC5E	FSIGPR8		
H'FC5F	FSIGPR9		
H'FC60	FSIGPRA		
H'FC61	FSIGPRB		
H'FC62	FSIGPRC		
H'FC63	FSIGPRD		
H'FC64	FSIGPRE		
H'FC65	FSIGPRF		
H'FC66	SLCR		
H'FC67	FSIARH		
H'FC68	FSIARM		
H'FC69	FSIARL		
H'FC6A	FSIWDRHH		
H'FC6B	FSIWDRHL		
H'FC6C	FSIWDRLH		
H'FC6D	FSIWDRLL		
H'FC6E	FSILSTR2		

H'FC98	FSITDR0		
H'FC99	FSITDR1		
H'FC9A	FSITDR2		
H'FC9B	FSITDR3		
H'FC9C	FSITDR4		
H'FC9D	FSITDR5		
H'FC9E	FSITDR6		
H'FC9F	FSITDR7		
H'FCA0	FSIRDR		
H'FD00	PWMREG0_A	MSTPB0 = 0	PWMU_A
H'FD01	PWMPRE0_A		
H'FD02	PWMREG1_A		
H'FD03	PWMPRE1_A		
H'FD04	PWMREG2_A		
H'FD05	PWMPRE2_A		
H'FD06	PWMREG3_A		
H'FD07	PWMPRE3_A		
H'FD08	PWMREG4_A		
H'FD09	PWMPRE4_A		
H'FD0A	PWMREG5_A		
H'FD0B	PWMPRE5_A		
H'FD0C	PWMCONA_A		
H'FD0D	PWMCONB_A		
H'FD0E	PWMCONC_A		
H'FD0F	PWMCOND_A		

H'FD17	PWMPRE3_B		
H'FD18	PWMREG4_B		
H'FD19	PWMPRE4_B		
H'FD1A	PWMREG5_B		
H'FD1B	PWMPRE5_B		
H'FD1C	PWMCONA_B		
H'FD1D	PWMCONB_B		
H'FD1E	PWMCONC_B		
H'FD1F	PWMCOND_B		
H'FD3A	SYTSR0	No condition	SYSTEM
H'FD3B	SYTSR1		
H'FD40	TCR_1	MSTP1 = 0	TPU_1
H'FD41	TMDR_1		
H'FD42	TIOR_1		
H'FD44	TIER_1		
H'FD45	TSR_1		
H'FD46	TCNT_1		
H'FD48	TGRA_1		
H'FD4A	TGRB_1		

H'FDD5	LADR4L		
H'FDD6	IDR4		
H'FDD7	ODR4		
H'FDD8	STR4		
H'FDD9	HICR4		
H'FDDA	SIRQCR2		
H'FDDB	SIRQCR3		
H'FE00	P6NCE	PORTS = 0	PORT
H'FE01	P6NCMC		
H'FE02	P6NCCS		
H'FE03	PCNCE		
H'FE04	PCNCMC		
H'FE05	PCNCCS		
H'FE06	PGNCE		
H'FE07	PGNCMC		
H'FE08	PGNCCS		
H'FE0C	PHPIN (Read)		
	PHDDR (Write)		
H'FE0D	PHODR		
H'FE0E	PHNOCR		

H'FE1D	PDNOCR		
H'FE20	TWR0MW	MSTP0 = 0	LPC
	TWR0SW		
H'FE21	TWR1		
H'FE22	TWR2		
H'FE23	TWR3		
H'FE24	TWR4		
H'FE25	TWR5		
H'FE26	TWR6		
H'FE27	TWR7		
H'FE28	TWR8		
H'FE29	TWR9		
H'FE2A	TWR10		
H'FE2B	TWR11		
H'FE2C	TWR12		
H'FE2D	TWR13		
H'FE2E	TWR14		
H'FE2F	TWR15		
H'FE30	IDR3		
H'FE31	ODR3		

H'FE39	ODR1		
H'FE3A	STR1		
H'FE3B	SIRQCR4		
H'FE3C	IDR2		
H'FE3D	ODR2		
H'FE3E	STR2		
H'FE3F	HISEL		
H'FE40	HICR0		
H'FE41	HICR1		
H'FE42	HICR2		
H'FE43	HICR3		
H'FE45	WUEMR	No condition	INT
H'FE46	PGODR	PORTS = 0	PORT
H'FE47	PGPIN (Read)		
	PGDDR (Write)		
H'FE49	PFODR		
H'FE4A	PEPIN (Read) (write prohibited)		
H'FE4B	PFPIN (Read)		
H'FE4C	PCODR		
H'FE4D	PDODR		
H'FE4E	PCPIN (Read)		
	PCDDR (Write)		
H'FE4F	PDPIN (Read)		
	PDDDR (Write)		

H'FE58	TGRA_0		
H'FE5A	TGRB_0		
H'FE5C	TGRC_0		
H'FE5E	TGRD_0		
H'FE70	TCR_2		TPU_2
H'FE71	TMDR_2		
H'FE72	TIOR_2		
H'FE74	TIER_2		
H'FE75	TSR_2		
H'FE76	TCNT_2		
H'FE78	TGRA_2		
H'FE7A	TGRB_2		
H'FE7D	SYSCR3	No condition	SYSTEM
H'FE7E	MSTPCRA		
H'FE7F	MSTPCRB		
H'FE81	KMIMR (RELOCATE = 1)		INT
H'FE82	KMPCR (RELOCATE = 1)		PORT
H'FE83	KMIMRA (RELOCATE = 1)		INT
H'FE84	WUESCR		
H'FE85	WUESR		
H'FE86	WER		
H'FE87	ICRD		

H'FE8F	SAR_2		ICE in ICCR_2 = 0	

H'FEA0	DACR
	(RELOCATE = 1)
		MSTP11 = 0
		MSTPA1 = 0
			REGS in
			DACNT/DADRB = 1
				PWMX

	DADRAH (RELOCATE = 1)		REGS in DACNT/DADRB = 0	
H'FEA1	DADRAL (RELOCATE = 1)			
H'FEA6	DADRBH (RELOCATE = 1)			
	DACNTH (RELOCATE = 1)		REGS in DACNT/DADRB = 1	
H'FEA7	DADRBL (RELOCATE = 1)		REGS in DACNT/DADRB = 0	
	DACNTL (RELOCATE = 1)		REGS in DACNT/DADRB = 1	
H'FEA8	FCCS	FLSHE = 1		ROM
H'FEA9	FPCS			
H'FEAA	FECS			
H'FEAC	FKEY			
H'FEAD	FMATS			
H'FEAE	FTDAR			

H'FEC5	KBTR_2			
H'FEC6	TCRXY	MSTP8 = 0		TMR_XY
H'FEC8	TCR_Y (RELOCATE = 1)			TMR_Y
H'FEC9	TCSR_Y (RELOCATE = 1)			
H'FECA	TCORA_Y (RELOCATE = 1)			
H'FECB	TCORB_Y (RELOCATE = 1)			
H'FECC	TCNT_Y (RELOCATE = 1)			
H'FECE	ICDR_1 (RELOCATE = 1)	MSTP3 = 0	ICE in ICCR_1 = 1	IIC_1
	SARX_1 (RELOCATE = 1)		ICE in ICCR_1 = 0	
H'FECF	ICMR_1 (RELOCATE = 1)		ICE in ICCR_1 = 1	
	SAR_1 (RELOCATE = 1)		ICE in ICCR_1 = 0	
H'FED0	ICCR_1 (RELOCATE = 1)			
H'FED1	ICSR_1 (RELOCATE = 1)			

H'FEDB	KBCR2_0		
H'FEDC	KBCRH_1		
H'FEDD	KBCRL_1		
H'FEDE	KBBR_1		
H'FEDF	KBCR2_1		
H'FEE0	KBCRH_2		
H'FEE1	KBCRL_2		
H'FEE2	KBBR_2		
H'FEE3	KBCR2_2		
H'FEE6	ICRES_0	MSTP4 = 0, IICE in STCR = 1	IIC_0
H'FEE8	ICRA	No condition	INT
H'FEE9	ICRB		
H'FEEA	ICRC		
H'FEEB	ISR		
H'FEEC	ISCRH		
H'FEED	ISCRL		
H'FEF4	ABRKCR		
H'FEF5	BARA		
H'FEF6	BARB		
H'FEF7	BARC		
H'FEF8	IER16		
H'FEF9	ISR16		
H'FEFA	ISCR16H		
H'FEFB	ISCR16L		

	SBYCR (RELOCATE = 1)	No condition	
H'FF85	LPWRCR (RELOCATE = 0)	FLSHE in STCR = 0	
	LPWRCR (RELOCATE = 1)	No condition	
H'FF86	MSTPCRH (RELOCATE = 0)	FLSHE in STCR = 0	
	MSTPCRH (RELOCATE = 1)	No condition	
H'FF87	MSTPCRL (RELOCATE = 0)	FLSHE in STCR = 0	
	MSTPCRL (RELOCATE = 1)	No condition	
H'FF88	SMR_1 (RELOCATE = 1)	MSTP6 = 0	SCI_1
	SMR_1 (RELOCATE = 0)	MSTP6 = 0, IICE in STCR = 0	
	ICCR_1 (RELOCATE = 0)	MSTP3 = 0, IICE in STCR = 1	IIC_1
H'FF89	BRR_1 (RELOCATE = 1)	MSTP6 = 0	SCI_1
	BRR_1 (RELOCATE = 0)	MSTP6 = 0, IICE in STCR = 0	
	ICSR_1 (RELOCATE = 0)	MSTP3 = 0, IICE in STCR = 1	IIC_1

	ICDR_1 (RELOCATE = 0)	MSTP3 = 0 IICE in STCR = 1	ICE in ICCR_1 = 1	IIC_1
	SARX_1 (RELOCATE = 0)		ICE in ICCR_1 = 0	
H'FF8F	ICMR_1 (RELOCATE = 0)		ICE in ICCR_1 = 1	
	SAR_1 (RELOCATE = 0)		ICE in ICCR_1 = 0	

H'FFA0	DADRAH
	(RELOCATE = 0)
		MSTP11 = 0
		MSTPA1 = 0
		IICE in STCR = 1
			REGS in DACNT/ DADRB = 0	PWMX

	DACR (RELOCATE = 0)		REGS in DACNT/ DADRB = 1	
	SMR_2 (RELOCATE = 0)	MSTP5 = 0, IICE in STCR = 0		SCI_2

H'FFA1	DADRAL
	(RELOCATE = 0)
		MSTP11 = 0
		MSTPA1 = 0
		IICE in STCR = 1
			REGS in DACNT/ DADRB = 0	PWMX

	DADRBH
	(RELOCATE = 0)
		MSTP11 = 0
		MSTPA1 = 0
		IICE in STCR = 1
			REGS in DACNT/ DADRB = 0	

	DACNTH (RELOCATE = 0)		REGS in DACNT/ DADRB = 1	
	BRR_2 (RELOCATE = 0)	MSTP5 = 0, IICE in STCR = 0		SCI_2
H'FFA2	SCR_2	MSTP5 = 0		
H'FFA3	TDR_2			
H'FFA4	SSR_2			

H'FFA9	TCNT_0 (Read)		
H'FFAA	PAODR	PORTS = 0	PORT
H'FFAB	PAPIN (Read)		
	PADDR (Write)		
H'FFAC	P1PCR		
H'FFAD	P2PCR		
H'FFAE	P3PCR		
H'FFB0	P1DDR		
H'FFB1	P2DDR		
H'FFB2	P1DR		
H'FFB3	P2DR		
H'FFB4	P3DDR		
H'FFB5	P4DDR		
H'FFB6	P3DR		
H'FFB7	P4DR		
H'FFB8	P5DDR		
H'FFB9	P6DDR		
H'FFBA	P5DR		
H'FFBB	P6DR		
H'FFBC	PBODR		
H'FFBD	P8DDR (Write)		
	PBPIN (Read)		

H'FFC4	SYSCR				
H'FFC5	MDCR				
H'FFC6	BCR	No condition			BSC
H'FFC7	WSCR				
H'FFC8	TCR_0	MSTP12 = 0			TMR
H'FFC9	TCR_1				
H'FFCA	TCSR_0				
H'FFCB	TCSR_1				
H'FFCC	TCORA_0				
H'FFCD	TCORA_1				
H'FFCE	TCORB_0				
H'FFCF	TCORB_1				
H'FFD8	ICCR_0	MSTP4 = 0 IICE in STCR = 1 (RELOCATE = 0)		No condition of IICE = 1 when RELOCATE = 1	IIC_0
H'FFD9	ICSR_0				
H'FFDE	ICDR_0	MSTP4 = 0 IICE in STCR = 1 (RELOCATE = 0)	ICE in ICCR_0 = 1		
	SARX_0		ICE in ICCR_0 = 0		
H'FFDF	ICMR_0	MSTP4 = 0 IICE in STCR = 1 (RELOCATE = 0)	ICE in ICCR_0 = 1		
	SAR_0		ICE in ICCR_0 = 0		

H'FFF0	TCR_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCR_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCR_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
H'FFF1	KMIMR (RELOCATE = 0)	MSTP2 = 0 KINWUE in SYSCR = 1		INT
	TCSR_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCSR_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCSR_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
H'FFF2	KMPCR (RELOCATE = 0)	MSTP2 = 0 KINWUE in SYSCR = 1		PORT
	TICRR (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TICRR (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCORA_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y

H'FFF4	TCNT_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCNT_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCNT_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
H'FFF5	TCORC (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCORC (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFF6	TCORA_X (RELOCATE = 1)	MSTP8 = 0		
	TCORA_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFF7	TCORB_X (RELOCATE = 1)	MSTP8 = 0		
	TCORB_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFFC	TCONRI (RELOCATE = 1)	MSTP8 = 0		
	TCONRI (RELOCATE = 0)	MSTP8 = 0, KINWUE in SYSCR = 0		
H'FFFE	TCONRS (RELOCATE = 1)	MSTP8 = 0		TMR_X, TMR
	TCONRS (RELOCATE = 0)	MSTP8 = 0, KINWUE in SYSCR = 0		

INT	WUESR	8	H'FE85	8	2
INT	WER	8	H'FE86	8	2
INT	ICRD	8	H'FE87	8	2
INT	ICRA	8	H'FEE8	8	2
INT	ICRB	8	H'FEE9	8	2
INT	ICRC	8	H'FEEA	8	2
INT	ISR	8	H'FEEB	8	2
INT	ISCRH	8	H'FEEC	8	2
INT	ISCRL	8	H'FEED	8	2
INT	KMIMR	8	H'FFF1 (RELOCATE = 0)	8	2
INT	ABRKCR	8	H'FEF4	8	2
INT	BARA	8	H'FEF5	8	2
INT	BARB	8	H'FEF6	8	2
INT	BARC	8	H'FEF7	8	2
INT	IER16	8	H'FEF8	8	2
INT	ISR16	8	H'FEF9	8	2
INT	ISCR16H	8	H'FEFA	8	2
INT	ISCR16L	8	H'FEFB	8	2
INT	ISSR16	8	H'FEFC	8	2
INT	ISSR	8	H'FEFD	8	2
INT	IER	8	H'FFC2	8	2
INT	KMIMRA	8	H'FFF3 (RELOCATE = 0)	8	2

PORT	P2DR	8	H'F903 (PORTS = 1)	8	2
PORT	P1PIN	8	H'F904 (Read) (PORTS = 1)	8	2
PORT	P2PIN	8	H'F905 (Read) (PORTS = 1)	8	2
PORT	P1PCR	8	H'F906 (PORTS = 1)	8	2
PORT	P2PCR	8	H'F907 (PORTS = 1)	8	2
PORT	P3DDR	8	H'F910 (PORTS = 1)	8	2
PORT	P4DDR	8	H'F911 (PORTS = 1)	8	2
PORT	P3DR	8	H'F912 (PORTS = 1)	8	2
PORT	P4DR	8	H'F913 (PORTS = 1)	8	2
PORT	P3PIN	8	H'F914 (Read) (PORTS = 1)	8	2
PORT	P4PIN	8	H'F915 (Read) (PORTS = 1)	8	2
PORT	P3PCR	8	H'F916 (PORTS = 1)	8	2
PORT	P5DDR	8	H'F920 (PORTS = 1)	8	2
PORT	P6DDR	8	H'F921 (PORTS = 1)	8	2

			(PORTS = 1)		
PORT	P6NCMC	8	H'F92D (PORTS = 1)	8	2
PORT	P6NCCS	8	H'F92F (PORTS = 1)	8	2
PORT	P8DDR	8	H'F931 (PORTS = 1)	8	2
PORT	P8DR	8	H'F933 (PORTS = 1)	8	2
PORT	P7PIN	8	H'F934 (Read) (PORTS = 1)	8	2
PORT	P8PIN	8	H'F935 (Read) (PORTS = 1)	8	2
PORT	P9DDR	8	H'F940 (PORTS = 1)	8	2
PORT	P9DR	8	H'F942 (PORTS = 1)	8	2
PORT	P9PIN	8	H'F944 (Read) (PORTS = 1)	8	2
PORT	P9PCR	8	H'F946 (PORTS = 1)	8	2
PORT	PADDR	8	H'F950 (PORTS = 1)	8	2
PORT	PBDDR	8	H'F951 (PORTS = 1)	8	2
PORT	PAODR	8	H'F952 (PORTS = 1)	8	2

			(PORTS = 1)		
PORT	PDDDR	8	H'F961 (PORTS = 1)	8	2
PORT	PCODR	8	H'F962 (PORTS = 1)	8	2
PORT	PDODR	8	H'F963 (PORTS = 1)	8	2
PORT	PCPIN	8	H'F964 (Read) (PORTS = 1)	8	2
PORT	PDPIN	8	H'F965 (Read) (PORTS = 1)	8	2
PORT	PCPCR	8	H'F966 (PORTS = 1)	8	2
PORT	PDPCR	8	H'F967 (PORTS = 1)	8	2
PORT	PCNOCR	8	H'F968 (PORTS = 1)	8	2
PORT	PDNOCR	8	H'F969 (PORTS = 1)	8	2
PORT	PCNCE	8	H'F96A (PORTS = 1)	8	2
PORT	PCNCMC	8	H'F96C (PORTS = 1)	8	2
PORT	PCNCCS	8	H'F96E (PORTS = 1)	8	2
PORT	PFDDR	8	H'F971 (PORTS = 1)	8	2

			(PORTS = 1)		
PORT	PGDDR	8	H'F980 (PORTS = 1)	8	2
PORT	PHDDR	8	H'F981 (PORTS = 1)	8	2
PORT	PGODR	8	H'F982 (PORTS = 1)	8	2
PORT	PHODR	8	H'F983 (PORTS = 1)	8	2
PORT	PGPIN	8	H'F984 (Read) (PORTS = 1)	8	2
PORT	PHPIN	8	H'F985 (Read) (PORTS = 1)	8	2
PORT	PHPCR	8	H'F987 (PORTS = 1)	8	2
PORT	PGNOCR	8	H'F988 (PORTS = 1)	8	2
PORT	PHNOCR	8	H'F989 (PORTS = 1)	8	2
PORT	PGNCE	8	H'F98A (PORTS = 1)	8	2
PORT	PGNCMC	8	H'F98C (PORTS = 1)	8	2
PORT	PGNCCS	8	H'F98E (PORTS = 1)	8	2
PORT	PIDDR	8	H'F990	8	2
PORT	PJDDR	8	H'F991	8	2

PORT	P6NCE	8	H'FE00 (PORTS = 0)	8	2
PORT	P6NCMC	8	H'FE01 (PORTS = 0)	8	2
PORT	P6NCCS	8	H'FE02 (PORTS = 0)	8	2
PORT	PCNCE	8	H'FE03 (PORTS = 0)	8	2
PORT	PCNCMC	8	H'FE04 (PORTS = 0)	8	2
PORT	PCNCCS	8	H'FE05 (PORTS = 0)	8	2
PORT	PGNCE	8	H'FE06 (PORTS = 0)	8	2
PORT	PGNCMC	8	H'FE07 (PORTS = 0)	8	2
PORT	PGNCCS	8	H'FE08 (PORTS = 0)	8	2
PORT	PHPIN	8	H'FE0C (Read) (PORTS = 0)	8	2
PORT	PHDDR	8	H'FE0C (Write) (PORTS = 0)	8	2
PORT	PHODR	8	H'FE0D (PORTS = 0)	8	2
PORT	PHNOCR	8	H'FE0E (PORTS = 0)	8	2

			(PORTS = 0)		
PORT	PFNOCR	8	H'FE19 (PORTS = 0)	8	2
PORT	PCNOCR	8	H'FE1C (PORTS = 0)	8	2
PORT	PDNOCR	8	H'FE1D (PORTS = 0)	8	2
PORT	PGODR	8	H'FE46 (PORTS = 0)	8	2
PORT	PGPIN	8	H'FE47 (Read) (PORTS = 0)	8	2
PORT	PGDDR	8	H'FE47 (Write) (PORTS = 0)	8	2
PORT	PFODR	8	H'FE49 (PORTS = 0)	8	2
PORT	PEPIN	8	H'FE4A (Read) (write prohibited) (PORTS = 0)	8	2
PORT	PFPIN	8	H'FE4B (Read) (PORTS = 0)	8	2
PORT	PFDDR	8	H'FE4B (Write) (PORTS = 0)	8	2

			(PORTS = 0)		
PORT	PDPIN	8	H'FE4F (Read) (PORTS = 0)	8	2
PORT	PDDDR	8	H'FE4F (Write) (PORTS = 0)	8	2
PORT	KMPCR	8	H'FE82 (RELOCATE = 1) (PORTS = 0)	8	2
PORT	PAODR	8	H'FFAA (PORTS = 0)	8	2
PORT	PAPIN	8	H'FFAB (Read) (PORTS = 0)	8	2
PORT	PADDR	8	H'FFAB (Write) (PORTS = 0)	8	2
PORT	P1PCR	8	H'FFAC (PORTS = 0)	8	2
PORT	P2PCR	8	H'FFAD (PORTS = 0)	8	2
PORT	P3PCR	8	H'FFAE (PORTS = 0)	8	2
PORT	P1DDR	8	H'FFB0 (PORTS = 0)	8	2
PORT	P2DDR	8	H'FFB1 (PORTS = 0)	8	2
PORT	P1DR	8	H'FFB2 (PORTS = 0)	8	2

			(PORTS = 0)		
PORT	P5DDR	8	H'FFB8 (PORTS = 0)	8	2
PORT	P6DDR	8	H'FFB9 (PORTS = 0)	8	2
PORT	P5DR	8	H'FFBA (PORTS = 0)	8	2
PORT	P6DR	8	H'FFBB (PORTS = 0)	8	2
PORT	PBODR	8	H'FFBC (PORTS = 0)	8	2
PORT	P8DDR	8	H'FFBD (Write) (PORTS = 0)	8	2
PORT	PBPIN	8	H'FFBD (Read) (PORTS = 0)	8	2
PORT	P7PIN	8	H'FFBE (Read) (PORTS = 0)	8	2
PORT	PBDDR	8	H'FFBE (Write) (PORTS = 0)	8	2
PORT	P8DR	8	H'FFBF (PORTS = 0)	8	2
PORT	P9DDR	8	H'FFC0 (PORTS = 0)	8	2
PORT	P9DR	8	H'FFC1 (PORTS = 0)	8	2
PORT	KMPCR	8	H'FFF2 (RELOCATE = 0) (PORTS = 0)	8	2

TDP_0	TDPCR1_0	8	H'FB4D	8	2
TDP_0	TDPIER_0	8	H'FB4E	8	2
TDP_0	TDPCR2_0	8	H'FB4F	8	2
TDP_0	TDPPDMN_0	16	H'FB50	16	2
TDP_1	TDPCNT_1	16	H'FB60	16	2
TDP_1	TDPWDMX_1	16	H'FB62	6	2
TDP_1	TDPWDMN_1	16	H'FB64	16	2
TDP_1	TDPPDMX_1	16	H'FB66	16	2
TDP_1	TDPICR_1	16	H'FB68	16	2
TDP_1	TDPICRF_1	16	H'FB6A	16	2
TDP_1	TDPCSR_1	8	H'FB6C	8	2
TDP_1	TDPCR1_1	8	H'FB6D	8	2
TDP_1	TDPIER_1	8	H'FB6E	8	2
TDP_1	TDPCR2_1	8	H'FB6F	8	2
TDP_1	TDPPDMN_1	16	H'FB70	16	2
TDP_2	TDPCNT_2	16	H'FB80	16	2
TDP_2	TDPWDMX_2	16	H'FB82	16	2
TDP_2	TDPWDMN_2	16	H'FB84	16	2
TDP_2	TDPPDMX_2	16	H'FB86	16	2
TDP_2	TDPICR_2	16	H'FB88	16	2
TDP_2	TDPICRF_2	16	H'FB8A	16	2
TDP_2	TDPCSR_2	8	H'FB8C	8	2
TDP_2	TDPCR1_2	8	H'FB8D	8	2
TDP_2	TPDIER_2	8	H'FB8E	8	2

TCM_0	TCMCR_0	8	H'FBC9	8	2
TCM_0	TCMIER_0	8	H'FBCA	8	2
TCM_0	TCMMINCM_0	16	H'FBCC	16	2
TCM_1	TCMCNT_1	16	H'FBD0	16	2
TCM_1	TCMMLCM_1	16	H'FBD2	16	2
TCM_1	TCMICR_1	16	H'FBD4	16	2
TCM_1	TCMICRF_1	16	H'FBD6	16	2
TCM_1	TCMCSR_1	8	H'FBD8	8	2
TCM_1	TCMCR_1	8	H'FBD9	8	2
TCM_1	TCMIER_1	8	H'FBDA	8	2
TCM_1	TCMMINCM_1	16	H'FBDC	16	2
TCM_2	TCMCNT_2	16	H'FBE0	16	2
TCM_2	TCMMLCM_2	16	H'FBE2	16	2
TCM_2	TCMICR_2	16	H'FBE4	16	2
TCM_2	TCMICRF_2	16	H'FBE6	16	2
TCM_2	TCMCSR_2	8	H'FBE8	8	2
TCM_2	TCMCR_2	8	H'FBE9	8	2
TCM_2	TCMIER_2	8	H'FBEA	8	2
TCM_2	TCMMINCM_2	16	H'FBEC	16	2
TCM_3	TCMCNT_3	16	H'FBF0	16	2
TCM_3	TCMMLCM_3	16	H'FBF2	16	2
TCM_3	TCMICR_3	16	H'FBF4	16	2
TCM_3	TCMICRF_3	16	H'FBF6	16	2
TCM_3	TCMCSR_3	8	H'FBF8	8	2

FSI	CMDHBARH	8	H'FC54	8	2
FSI	FSICMDR	8	H'FC55	8	2
FSI	FSILSTR1	8	H'FC56	8	2
FSI	FSIGPR1	8	H'FC57	8	2
FSI	FSIGPR2	8	H'FC58	8	2
FSI	FSIGPR3	8	H'FC59	8	2
FSI	FSIGPR4	8	H'FC5A	8	2
FSI	FSIGPR5	8	H'FC5B	8	2
FSI	FSIGPR6	8	H'FC5C	8	2
FSI	FSIGPR7	8	H'FC5D	8	2
FSI	FSIGPR8	8	H'FC5E	8	2
FSI	FSIGPR9	8	H'FC5F	8	2
FSI	FSIGPRA	8	H'FC60	8	2
FSI	FSIGPRB	8	H'FC61	8	2
FSI	FSIGPRC	8	H'FC62	8	2
FSI	FSIGPRD	8	H'FC63	8	2
FSI	FSIGPRE	8	H'FC64	8	2
FSI	FSIGPRF	8	H'FC65	8	2
FSI	SLCR	8	H'FC66	8	2
FSI	FSIARH	8	H'FC67	8	2
FSI	FSIARM	8	H'FC68	8	2
FSI	FSIARL	8	H'FC69	8	2
FSI	FSIWDRHH	8	H'FC6A	8	2
FSI	FSIWDRHL	8	H'FC6B	8	2

FSI	FSIRDINS	8	H'FC94	8	2
FSI	FSIPPINS	8	H'FC95	8	2
FSI	FSISTR	8	H'FC96	8	2
FSI	FSITDR0	8	H'FC98	8	2
FSI	FSITDR1	8	H'FC99	8	2
FSI	FSITDR2	8	H'FC9A	8	2
FSI	FSITDR3	8	H'FC9B	8	2
FSI	FSITDR4	8	H'FC9C	8	2
FSI	FSITDR5	8	H'FC9D	8	2
FSI	FSITDR6	8	H'FC9E	8	2
FSI	FSITDR7	8	H'FC9F	8	2
FSI	FSIRDR	8	H'FCA0	8	2
CIR	CCR1	8	H'FA40	8	2
CIR	CCR2	8	H'FA41	8	2
CIR	CSTR	8	H'FA42	8	2
CIR	CEIR	8	H'FA43	8	2
CIR	BRR	8	H'FA44	8	2
CIR	CIRRDR0 to 7	8	H'FA45	8	2
CIR	HHMIN	16	H'FA46	8	2
CIR	HHMAX	16	H'FA48	8	2
CIR	HLMIN	8	H'FA4A	8	2
CIR	HLMAX	8	H'FA4B	8	2
CIR	DT0MIN	8	H'FA4C	8	2
CIR	DT0MAX	8	H'FA4D	8	2

PWMU_A	PWMREG2	8	H'FD04	8	2
PWMU_A	PWMPRE2	8	H'FD05	8	2
PWMU_A	PWMREG3	8	H'FD06	8	2
PWMU_A	PWMPRE3	8	H'FD07	8	2
PWMU_A	PWMREG4	8	H'FD08	8	2
PWMU_A	PWMPRE4	8	H'FD09	8	2
PWMU_A	PWMREG5	8	H'FD0A	8	2
PWMU_A	PWMPRE5	8	H'FD0B	8	2
PWMU_A	PWMCONA	8	H'FD0C	8	2
PWMU_A	PWMCONB	8	H'FD0D	8	2
PWMU_A	PWMCONC	8	H'FD0E	8	2
PWMU_A	PWMCOND	8	H'FD0F	8	2
PWMU_B	PWMREG0	8	H'FD10	8	2
PWMU_B	PWMPRE0	8	H'FD11	8	2
PWMU_B	PWMREG1	8	H'FD12	8	2
PWMU_B	PWMPRE1	8	H'FD13	8	2
PWMU_B	PWMREG2	8	H'FD14	8	2
PWMU_B	PWMPRE2	8	H'FD15	8	2
PWMU_B	PWMREG3	8	H'FD16	8	2
PWMU_B	PWMPRE3	8	H'FD17	8	2
PWMU_B	PWMREG4	8	H'FD18	8	2
PWMU_B	PWMPRE4	8	H'FD19	8	2
PWMU_B	PWMREG5	8	H'FD1A	8	2
PWMU_B	PWMPRE5	8	H'FD1B	8	2

PWMX	DADRAL	8	H'FEA1 (RELOCATE = 1)	8	2
PWMX	DADRBH	8	H'FEA6 (RELOCATE = 1)	8	2
PWMX	DACNTH	8	H'FEA6 (RELOCATE = 1)	8	2
PWMX	DADRBL	8	H'FEA7 (RELOCATE = 1)	8	2
PWMX	DACNTL	8	H'FEA7 (RELOCATE = 1)	8	2
PWMX	PCSR	8	H'FF82	8	2
PWMX	DACR	8	H'FFA0 (RELOCATE = 0)	8	2
PWMX	DADRAH	8	H'FFA0 (RELOCATE = 0)	8	2
PWMX	DADRAL	8	H'FFA1 (RELOCATE = 0)	8	2
PWMX	DACNTH	8	H'FFA6 (RELOCATE = 0)	8	2
PWMX	DADRBH	8	H'FFA6 (RELOCATE = 0)	8	2
PWMX	DACNTL	8	H'FFA7 (RELOCATE = 0)	8	2
PWMX	DADRBL	8	H'FFA7 (RELOCATE = 0)	8	2
TPU_0	TCR_0	8	H'FE50	8	2
TPU_0	TMDR_0	8	H'FE51	8	2

TPU_0	TGRC_0	16	H'FE5C	16	2
TPU_0	TGRD_0	16	H'FE5E	16	2
TPU_1	TCR_1	8	H'FD40	8	2
TPU_1	TMDR_1	8	H'FD41	8	2
TPU_1	TIOR_1	8	H'FD42	8	2
TPU_1	TIER_1	8	H'FD44	8	2
TPU_1	TSR_1	8	H'FD45	8	2
TPU_1	TCNT_1	16	H'FD46	16	2
TPU_1	TGRA_1	16	H'FD48	16	2
TPU_1	TGRB_1	16	H'FD4A	16	2
TPU_2	TCR_2	8	H'FE70	8	2
TPU_2	TMDR_2	8	H'FE71	8	2
TPU_2	TIOR_2	8	H'FE72	8	2
TPU_2	TIER_2	8	H'FE74	8	2
TPU_2	TSR_2	8	H'FE75	8	2
TPU_2	TCNT_2	16	H'FE76	16	2
TPU_2	TGRA_2	16	H'FE78	16	2
TPU_2	TGRB_2	16	H'FE7A	16	2
TPU common	TSTR	8	H'FEB0	8	2
TPU common	TSYR	8	H'FEB1	8	2
TMR_0	TCR_0	8	H'FFC8	8	2
TMR_0	TCSR_0	8	H'FFCA	8	2
TMR_0	TCORA_0	8	H'FFCC	16	2

TMR_X	TCR_X	8	H'FFF0	8	2
TMR_X	TCSR_X	8	H'FFF1	8	2
TMR_X	TICRR	8	H'FFF2	8	2
TMR_X	TICRF	8	H'FFF3	8	2
TMR_X	TCNT_X	8	H'FFF4	8	2
TMR_X	TCORC	8	H'FFF5	8	2
TMR_X	TCORA_X	8	H'FFF6	8	2
TMR_X	TCORB_X	8	H'FFF7	8	2
TMR_X	TCONRI	8	H'FFFC	8	2
TMR_Y	TCR_Y	8	H'FEC8 (RELOCATE = 1)	8	2
TMR_Y	TCSR_Y	8	H'FEC9 (RELOCATE = 1)	8	2
TMR_Y	TCORA_Y	8	H'FECA (RELOCATE = 1)	8	2
TMR_Y	TCORB_Y	8	H'FECB (RELOCATE = 1)	8	2
TMR_Y	TCNT_Y	8	H'FECC (RELOCATE = 1)	8	2
TMR_Y	TCR_Y	8	H'FFF0 (RELOCATE = 0)	8	2
TMR_Y	TCSR_Y	8	H'FFF1 (RELOCATE = 0)	8	2
TMR_Y	TCORA_Y	8	H'FFF2 (RELOCATE = 0)	8	2
TMR_Y	TCORB_Y	8	H'FFF3 (RELOCATE = 0)	8	2

WDT_0	TCNT_0	8	H'FFA8 (Write)	16	2
WDT_0	TCNT_0	8	H'FFA9 (Read)	8	2
WDT_1	TCSR_1	8	H'FFEA (Write)	16	2
WDT_1	TCSR_1	8	H'FFEA (Read)	8	2
WDT_1	TCNT_1	8	H'FFEA (Write)	16	2
WDT_1	TCNT_1	8	H'FFEB (Read)	8	2
SCI_1	SMR_1	8	H'FF88	8	2
SCI_1	BRR_1	8	H'FF89	8	2
SCI_1	SCR_1	8	H'FF8A	8	2
SCI_1	TDR_1	8	H'FF8B	8	2
SCI_1	SSR_1	8	H'FF8C	8	2
SCI_1	RDR_1	8	H'FF8D	8	2
SCI_1	SCMR_1	8	H'FF8E	8	2
SCI2	SMR_2	8	H'FFA0	8	2
SCI2	BRR_2	8	H'FFA1	8	2
SCI2	SCR_2	8	H'FFA2	8	2
SCI2	TDR_2	8	H'FFA3	8	2
SCI2	SSR_2	8	H'FFA4	8	2
SCI2	RDR_2	8	H'FFA5	8	2
SCI2	SCMR_2	8	H'FFA6	8	2
IIC_0	ICXR_0	8	H'FED4	8	2
IIC_0	ICCR_0	8	H'FFD8	8	2
IIC_0	ICSR_0	8	H'FFD9	8	2
IIC_0	ICDR_0	8	H'FFDE	8	2

			(RELOCATE = 1)		
IIC_1	SAR_1	8	H'FECF (RELOCATE = 1)	8	2
IIC_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	8	2
IIC_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	8	2
IIC_1	ICXR_1	8	H'FED5	8	2
IIC_1	ICCR_1	8	H'FF88 (RELOCATE = 0)	8	2
IIC_1	ICSR_1	8	H'FF89 (RELOCATE = 0)	8	2
IIC_1	ICDR_1	8	H'FF8E (RELOCATE = 0)	8	2
IIC_1	SARX_1	8	H'FF8E (RELOCATE = 0)	8	2
IIC_1	ICMR_1	8	H'FF8F (RELOCATE = 0)	8	2
IIC_1	SAR_1	8	H'FF8F (RELOCATE = 0)	8	2
IIC_2	ICCR_2	8	H'FE88	8	2
IIC_2	ICSR_2	8	H'FE89	8	2
IIC_2	ICRES_2	8	H'FE8A	8	2
IIC_2	ICXR_2	8	H'FE8C	8	2
IIC_2	ICDR_2	8	H'FE8E	8	2
IIC_2	SARX_2	8	H'FE8E	8	2

PS2_0	KBBR_0	8	H'FEDA	8	2
PS2_0	KBCR2_0	8	H'FEDB	8	2
PS2_1	KBCR1_1	8	H'FEC2	8	2
PS2_1	KBTR_1	8	H'FEC3	8	2
PS2_1	KBCRH_1	8	H'FEDC	8	2
PS2_1	KBCRL_1	8	H'FEDD	8	2
PS2_1	KBBR_1	8	H'FEDE	8	2
PS2_1	KBCR2_1	8	H'FEDF	8	2
PS2_2	KBCR1_2	8	H'FEC4	8	2
PS2_2	KBTR_2	8	H'FEC5	8	2
PS2_2	KBCRH_2	8	H'FEE0	8	2
PS2_2	KBCRL_2	8	H'FEE1	8	2
PS2_2	KBBR_2	8	H'FEE2	8	2
PS2_2	KBCR2_2	8	H'FEE3	8	2
PS2_3	KBCR1_3	8	H'FED2	8	2
PS2_3	KBTR_3	8	H'FED3	8	2
PS2_3	KBCRH_3	8	H'FFE0	8	2
PS2_3	KBCRL_3	8	H'FFE1	8	2
PS2_3	KBBR_3	8	H'FFE2	8	2
PS2_3	KBCR2_3	8	H'FFE3	8	2
LPC	LADR1H	8	H'FDC0	8	2
LPC	LADR1L	8	H'FDC1	8	2
LPC	LADR2H	8	H'FDC2	8	2
LPC	LADR2L	8	H'FDC3	8	2

LPC	HICR4	8	H'FDD9	8	2
LPC	SIRQCR2	8	H'FDDA	8	2
LPC	SIRQCR3	8	H'FDDB	8	2
LPC	TWR0MW	8	H'FE20	8	2
LPC	TWR0SW	8	H'FE20	8	2
LPC	TWR1	8	H'FE21	8	2
LPC	TWR2	8	H'FE22	8	2
LPC	TWR3	8	H'FE23	8	2
LPC	TWR4	8	H'FE24	8	2
LPC	TWR5	8	H'FE25	8	2
LPC	TWR6	8	H'FE26	8	2
LPC	TWR7	8	H'FE27	8	2
LPC	TWR8	8	H'FE28	8	2
LPC	TWR9	8	H'FE29	8	2
LPC	TWR10	8	H'FE2A	8	2
LPC	TWR11	8	H'FE2B	8	2
LPC	TWR12	8	H'FE2C	8	2
LPC	TWR13	8	H'FE2D	8	2
LPC	TWR14	8	H'FE2E	8	2
LPC	TWR15	8	H'FE2F	8	2
LPC	IDR3	8	H'FE30	8	2
LPC	ODR3	8	H'FE31	8	2
LPC	STR3	8	H'FE32	8	2
LPC	HICR5	8	H'FE33	8	2

LPC	SIRQCR4	8	H'FE3B	8	2
LPC	IDR2	8	H'FE3C	8	2
LPC	ODR2	8	H'FE3D	8	2
LPC	STR2	8	H'FE3E	8	2
LPC	HISEL	8	H'FE3F	8	2
LPC	HICR0	8	H'FE40	8	2
LPC	HICR1	8	H'FE41	8	2
LPC	HICR2	8	H'FE42	8	2
LPC	HICR3	8	H'FE43	8	2
A/D converter	ADDRA	16	H'FC00	16	2
A/D converter	ADDRB	16	H'FC02	16	2
A/D converter	ADDRC	16	H'FC04	16	2
A/D converter	ADDRD	16	H'FC06	16	2
A/D converter	ADDRE	16	H'FC08	16	2
A/D converter	ADDRF	16	H'FC0A	16	2
A/D converter	ADDRG	16	H'FC0C	16	2
A/D converter	ADDRH	16	H'FC0E	16	2
A/D converter	ADCSR	8	H'FC10	8	2
A/D converter	ADCR	8	H'FC11	8	2
SCIF	FRBR	8	H'FC20	8	2
SCIF	FTHR	8	H'FC20	8	2
SCIF	FDLL	8	H'FC20	8	2
SCIF	FIER	8	H'FC21	8	2
SCIF	FDLH	8	H'FC21	8	2
SCIF	FIIR	8	H'FC22	8	2

ROM	FCCS	8	H'FEA8	8	2
ROM	FPCS	8	H'FEA9	8	2
ROM	FECS	8	H'FEAA	8	2
ROM	FKEY	8	H'FEAC	8	2
ROM	FMATS	8	H'FEAD	8	2
ROM	FTDAR	8	H'FEAE	8	2
SYSTEM	SYSCR3	8	H'FE7D	8	2
SYSTEM	MSTPCRA	8	H'FE7E	8	2
SYSTEM	MSTPCRB	8	H'FE7F	8	2
SYSTEM	SBYCR	8	H'FF84	8	2
SYSTEM	LPWRCR	8	H'FF85	8	2
SYSTEM	MSTPCRH	8	H'FF86	8	2
SYSTEM	MSTPCRL	8	H'FF87	8	2
SYSTEM	STCR	8	H'FFC3	8	2
SYSTEM	SYSCR	8	H'FFC4	8	2
SYSTEM	MDCR	8	H'FFC5	8	2

Input voltage (except ports 7, D, A, G, I, PE4, PE2 to PE0, P97, P86, P52, and P42)	V_{in}	–0.3 to V_{CC} + 0.3
Input voltage (ports A, G, I, PE4, PE2 to PE0, P97, P86, P52, and P42)	V_{in}	–0.3 to +7.0
Input voltage (AN input is not selected for port D)	V_{in}	–0.3 to V_{CC} + 0.3
Input voltage (AN input is selected for port D)	V_{in}	–0.3 to V_{CC} +0.3 or –0.3 to AV_{C} +0.3 whichever is lower
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} + 0.3
Reference power supply voltage	AVref	–0.3 to AV_{CC} + 0.3
Analog power supply voltage	AV_{CC}	–0.3 to +4.3
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3
Operating temperature	T_{opr}	–20 to +75
Operating temperature (when flash memory is programmed or erased)	T_{opr}	0 to +75
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to this LSI may result if absolute maximum ratings are ex

Make sure the applied power supply does not exceed 4.3 V.

Note: * Voltage applied to the VCC pin.

The VCL pin should not be applied a voltage.

Item			Symbol	Min.	Typ.	Max.	Unit	Co
Schmitt trigger input voltage	P67 to P60, $\overline{IRQ7}$ to $\overline{IRQ0}$, $\overline{IRQ15}$ to $\overline{IRQ8}$ $\overline{KIN7}$ to $\overline{KIN0}$, $\overline{KIN15}$ to $\overline{KIN8}$, $\overline{WUE15}$ to $\overline{WUE8}$ $\overline{ExIRQ7}$ to $\overline{ExIRQ6}$, and $\overline{ExIRQ15}$ to $\overline{ExIRQ8}$	(1)	V_T^-	V_{CC} × 0.2	—	—	V	
			V_T^+	—	—	V_{CC} × 0.7		
			$V_T^+ - V_T^-$	V_{CC} × 0.05	—	—		
Input high voltage	$\overline{RES}$, NMI, MD2, MD1, and $\overline{ETRST}$	(2)	V_{IH}	V_{CC} × 0.9	—	V_{CC} + 0.3		
	EXTAL			V_{CC} × 0.7	—	V_{CC} + 0.3		
	Port 7			AV_{CC} × 0.7	—	AV_{CC} + 0.3		
	Ports A, G, I, PE4, PE2 to PE0, P97, P86, P52, and P42			V_{CC} × 0.7	—	5.5		
	Input pins other than (1) and (2) above			V_{CC} × 0.7	—	V_{CC} + 0.3		
Input low voltage	$\overline{RES}$, MD2, MD1, and $\overline{ETRST}$	(3)	V_{IL}	– 0.3	—	V_{CC} × 0.1		
	NMI, EXTAL, and input pins other than (1) and (3) above			–0.3	—	V_{CC} × 0.2		
Output high voltage	All output pins (except for ports A, G, I, P97, P86, P52, and P42)		V_{OH}	V_{CC} – 0.5	—	—		I_{OH}
				V_{CC} – 1.0	—	—		I_{OH}
	Ports A, G, I, P97, P86, P52, and P42*[2]			0.5	—	—		I_{OH}
Output low voltage	All output pins *[3]		V_{OL}	—	—	0.4		I_{OL}
	Ports 1, 2, 3, C, and D			—	—	1.0		I_{OL}

leakage current (off state)	and F to J						V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3, P95 to P90, ports 6, B to D, F, H, and J	$-I_p$	20	—	150	μA	V_{in} = 0 V
Input capacitance	All pins	C_{in}	—	—	10	pF	V_{in} = 0 V f = 1 MHz T_a = 25 °C
Supply current*[4]	Normal operation	I_{CC}	—	25	40	mA	V_{CC} = 3.0 V to 3 f = 20 MHz, all operating, high
	Sleep mode		—	20	35		V_{CC} = 3.0 V to 3 f = 20 MHz
	Standby mode		—	35	70	μA	Ta ≤ 50 °C
			—	—	200		50 °C < Ta
Analog power supply current	During A/D conversion	AI_{CC}	—	1	2	mA	
	A/D conversion standby		—	0.01	5	μA	AV_{CC} = 3.0 V to
Reference power supply current	During A/D conversion	AI_{ref}	—	1	2	mA	
	A/D conversion standby		—	0.01	5	μA	AVref = 3.0 V t
VCC start voltage		VCC_{START}	—	0	0.8	V	
VCC rising edge		SVCC	—	—	20	ms/V	

function is selected is rated separately.

4. Current consumption values are for V_{IH} min = V_{CC} – 0.2 V and V_{IL} max = 0.2 V with all out unloaded and the on-chip pull-up MOSs in the off state.

Table 28.2 DC Characteristics (3) Using LPC Function

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V

Item		**Symbol**	**Min.**	**Max.**	**Unit**	**Te Co**
Input high voltage	P37 to P30, P82 to P80, PB1, PB0	V_{IH}	V_{CC} × 0.5	—	V	
Input low voltage	P37 to P30, P82 to P80, PB1, PB0	V_{IL}	—	V_{CC} × 0.3	V	
Output high voltage	P37, P33 to P30, P82 to P80, PB1, PB0	V_{OH}	V_{CC} × 0.9	—	V	I_{OH} m
Output low voltage	P37, P33 to P30, P82 to P80, PB1, PB0	V_{OL}	—	V_{CC}×0.1	V	I_{OL}

Input pull-up MOS current	$-I_P$	30	—	300	μA	V_{in} = 0 V
Input capacitance	C_{in}	—	—	10	pF	V_{in} = 0 V, Ta = 25 °

Table 28.3 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0V

Item		Symbol	Min.	Typ.	M
Permissible output low current (per pin)	SCL0, SDA0, SCL1, SDA1, SCL2, SDA2, ExSCLA, ExSDAA, ExSCLB, ExSDAB, PS2AC to PS2DC, PS2AD to S2DD, and PA7 to PA4 (bus drive function selected)	I_{OL}	—	—	8
	Ports 1, 2, 3, C, and D		—	—	5
	Other output pins		—	—	2
Permissible output low current (total)	Total of ports 1, 2, 3, C, and D	ΣI_{OL}	—	—	4
	Total of all output pins, including the above		—	—	6
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2
Permissible output high current (total)	Total of all output pins	$\Sigma - I_{OH}$	—	—	3

Notes: 1. To protect LSI reliability, do not exceed the output current values in table 28.3

2. When driving a Darlington transistor or LED, always insert a current-limiting r the output line, as show in figures 28.1 and 28.2.

Input high voltage	V_{IH}	V_{CC} × 0.7	—	5.5		
Input low voltage	V_{IL}	–0.5	—	V_{CC} × 0.3		
Output low voltage	V_{OL}	—	—	0.5		I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA
Input capacitance	C_{in}	—	—	10	pF	V_{in} = 0 V, f = 1 MH 25 °C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0.5 to V_{CC} – 0.

Conditions: V_{CC} = 3.0 V to 3.6V, V_{SS} = 0 V

Applicable Pins: PS2AC to PS2DC, PS2AD to PS2DD, and PA7 to PA4 (bus drive func selected)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	V_{OL}	—	—	0.5	V	I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA

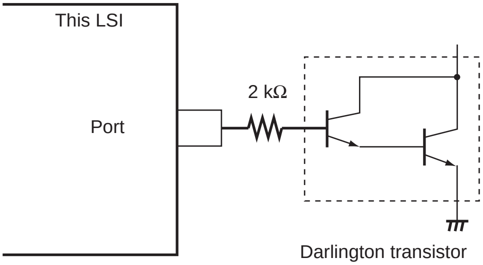


Figure 28.1 Darlington Transistor Drive Circuit (Example)

28.3 AC Characteristics

Figure 28.3 shows the test conditions for the AC characteristics.

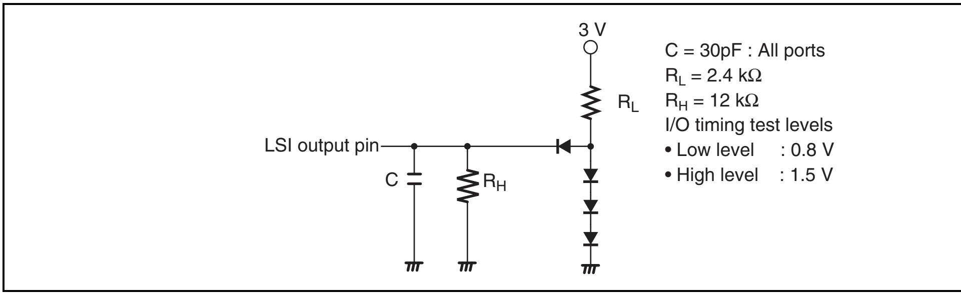


Figure 28.3 Output Load Circuit

Condition B: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, φ = 10 MHz to 20 MHz

		Condition A		Condition B			
Item	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**R**
Clock cycle time	t_{cyc}	100	125	50	100	ns	F
Clock high pulse width	t_{CH}	30	—	20	—		
Clock low pulse width	t_{CL}	30	—	20	—		
Clock rise time	t_{Cr}	—	20	—	5		
Clock fall time	t_{Cf}	—	20	—	5		
Reset oscillation stabilization (crystal)	t_{OSC1}	20	—	20	—	ms	F
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—	8	—		F
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	µs	F

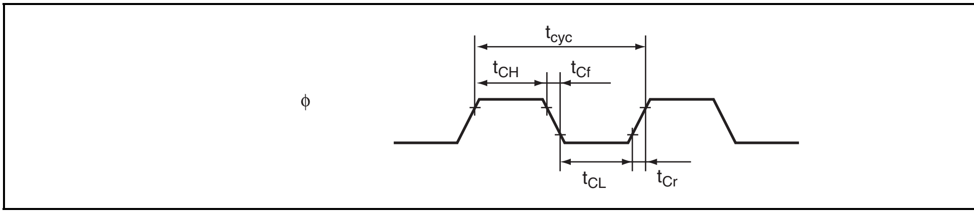


Figure 28.4 System Clock Timing

Figure 28.5 Oscillation Stabilization Timing

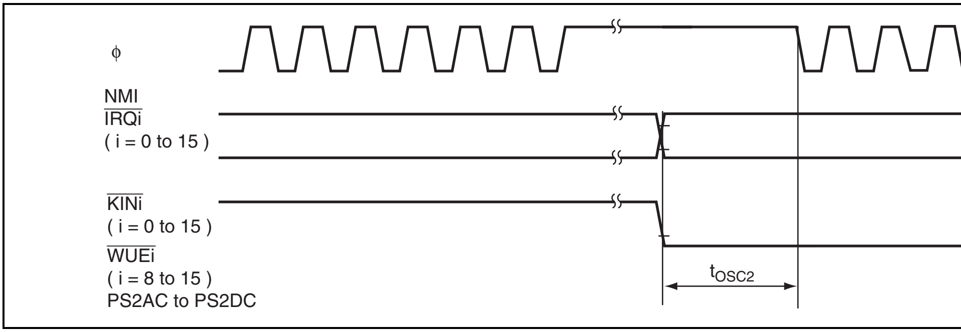


Figure 28.6 Oscillation Stabilization Timing (Exiting Software Standby Mo

Item	Symbol	Min.	Max.	Unit	Tes Con
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figu
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figu
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) (exiting software standby mode)	t_{IRQW}	200	—		

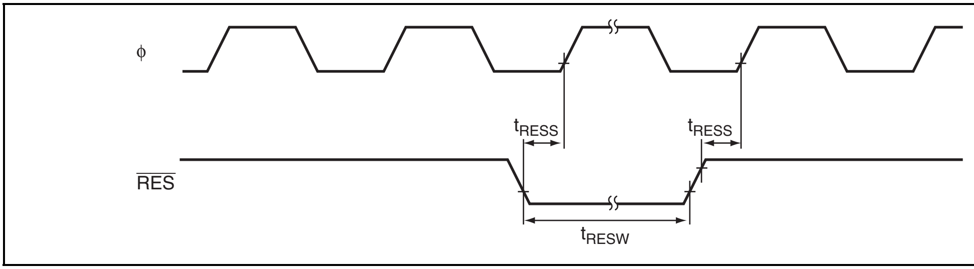


Figure 28.7 Reset Input Timing

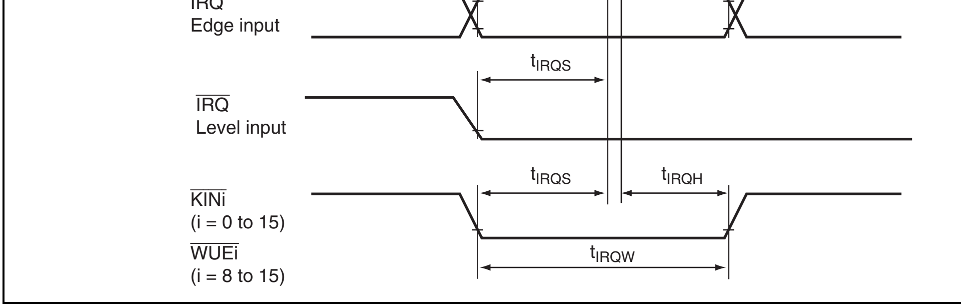


Figure 28.8 Interrupt Input Timing

(33 MHz)

Item			Symbol	Min.	Max.	Unit	T C
I/O ports	Output data delay time*[2]		t_{PWD}	—	50	ns	F
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
TPU	Timer output delay time		t_{TOCD}	—	50	ns	F
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—		F
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	50	ns	F
	Timer reset input setup time		t_{TMRS}	30	—		F
	Timer clock input setup time		t_{TMCS}	30	—		F
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
TCM	TCM input setup time		t_{TCMS}	30	—	ns	F
	TCM clock input setup time		t_{TCMCKS}	30	—		F
	TCM clock pulse width		t_{TCMCKW}	1.5	—	t_{cyc}	
TDP	TDP input setup time		t_{TDPS}	30	—	ns	F
	TDP clock input setup time		t_{TDPCKS}	30	—		F
	TDP clock pulse width		t_{TDPCKW}	1.5	—	t_{cyc}	
PWMU, PWMX	Pulse output delay time		t_{PWOD}	—	50	ns	F
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	F
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	

SS signal rise delay time	t_{SSH}	12	—
SS signal fall delay time	t_{SSL}	12	—
Transmit signal delay time	t_{TXD}	—	12
Receive signal setup time	t_{RXS}	5	—
Receive signal hold time	t_{RXH}	5	—

Notes: 1. Applied only for the peripheral modules that are available during subclock ope

2. Other than P52, P97, P86, P42, port A, port G, and port I.

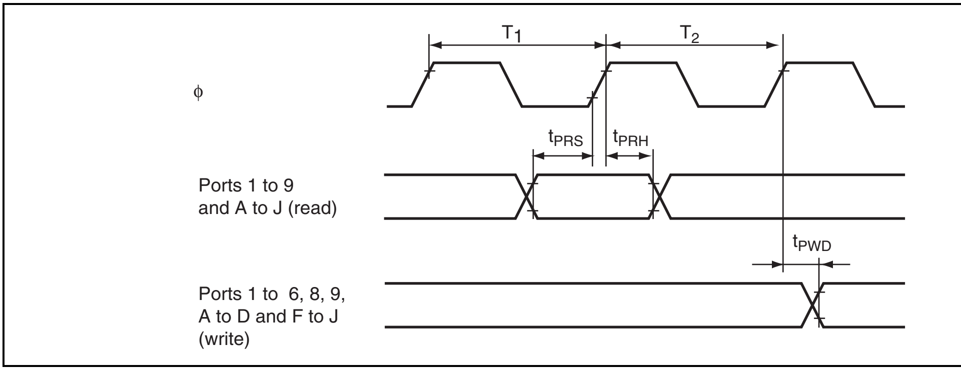


Figure 28.9 I/O Port Input/Output Timing

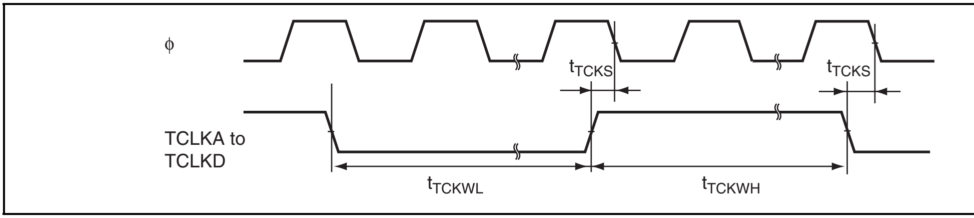


Figure 28.11 TPU Clock Input Timing

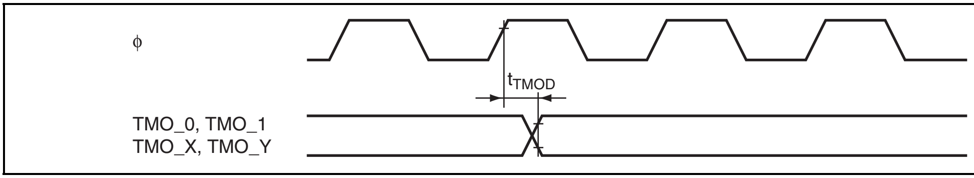


Figure 28.12 8-Bit Timer Output Timing

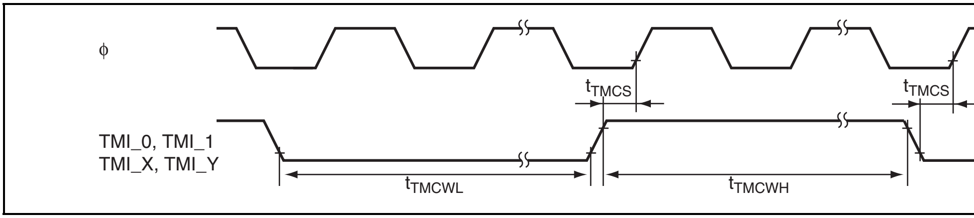


Figure 28.13 8-Bit Timer Clock Input Timing

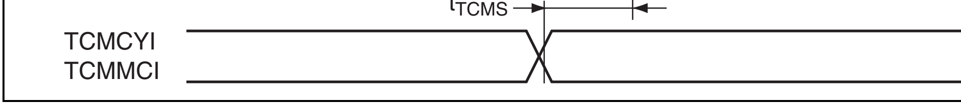


Figure 28.15 TCM Input Setup Time

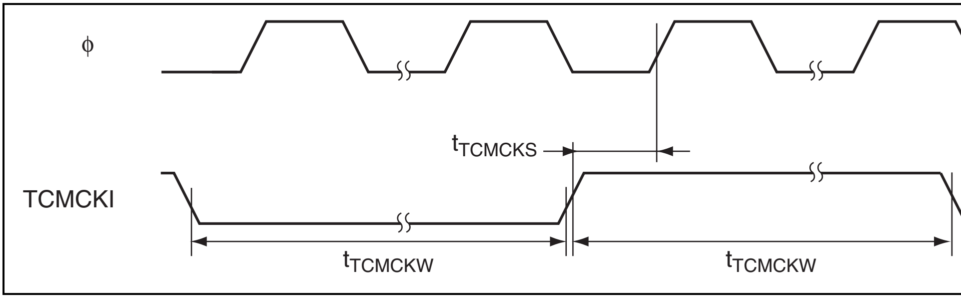


Figure 28.16 TCM Clock Input Timing

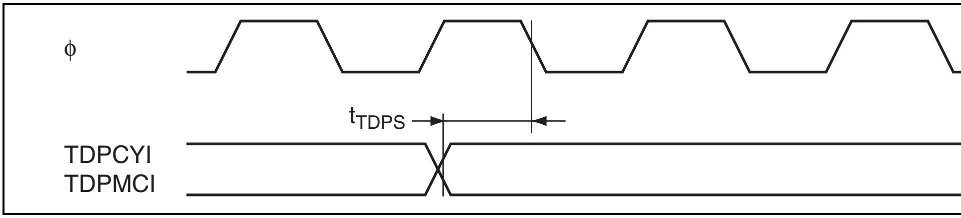


Figure 28.17 TDP Input Setup Time

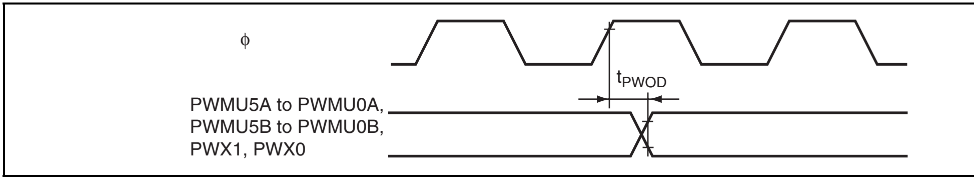


Figure 28.19 PWMU, PWMX Output Timing

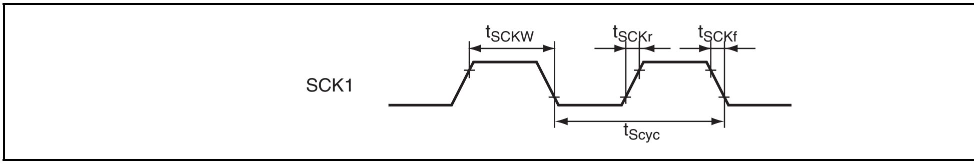


Figure 28.20 SCK Clock Input Timing

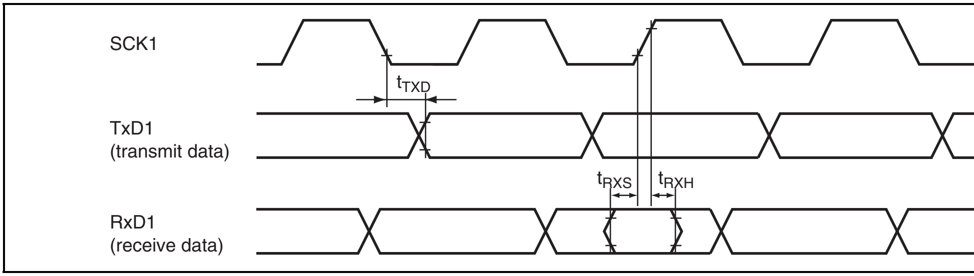


Figure 28.21 SCI Input/Output Timing (Clock Synchronous Mode)

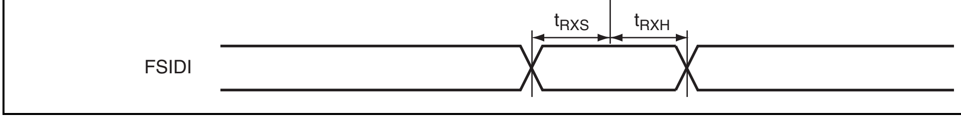


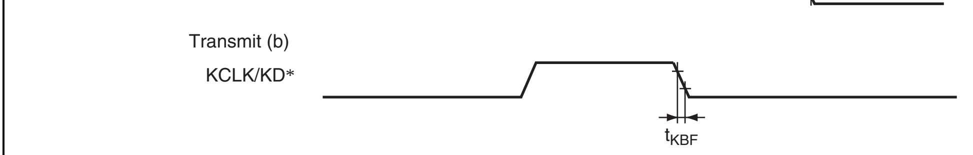
Figure 28.22 FSI Input/Output Timing

Table 28.8 PS2 Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to maximum operating frequ

		Standard Value				Test
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Conditions**
KCLK, KD output fall time	t_{KBF}	—	—	250	ns	
KCLK, KD input data hold time	t_{KBIH}	150	—	—		
KCLK, KD input data setup time	t_{KBIS}	150	—	—		
KCLK, KD output delay time	t_{KBOD}	—	—	450		
KCLK, KD capacitive load	C_b	—	—	400	pF	

Note: * When KCLK and KD are output, an external pull-up register must be connect shown in figure 28.23.

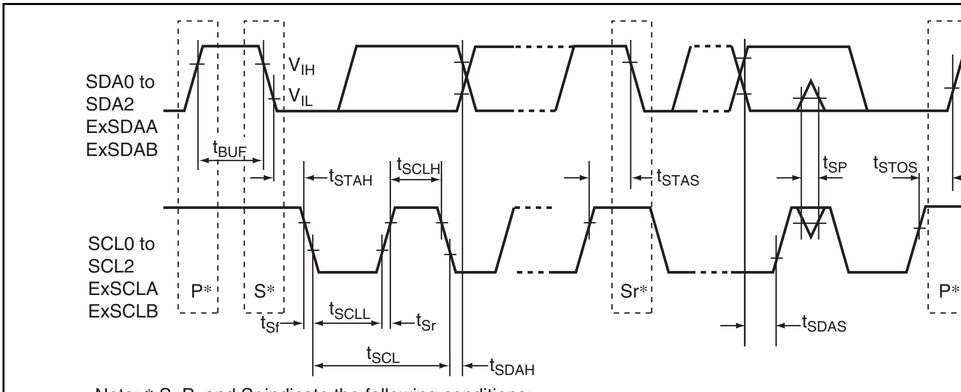


Note: * KCLK : PS2AC to PS2DC
KD : PS2AD to PS2DD

Figure 28.23 PS2 Timing

SCL, SDA input fall time	t_{Sf}			300	ns
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	1	t_{cyc}
SDA input bus free time	t_{BUF}	5	—	—	
Start condition input hold time	t_{STAH}	3	—	—	
Retransmission start condition input setup time	t_{STAS}	3	—	—	
Stop condition input setup time	t_{STOS}	3	—	—	
Data input setup time	t_{SDAS}	0.5	—	—	
Data input hold time	t_{SDAH}	0	—	—	ns
SCL, SDA capacitive load	C_b	—	—	400	pF

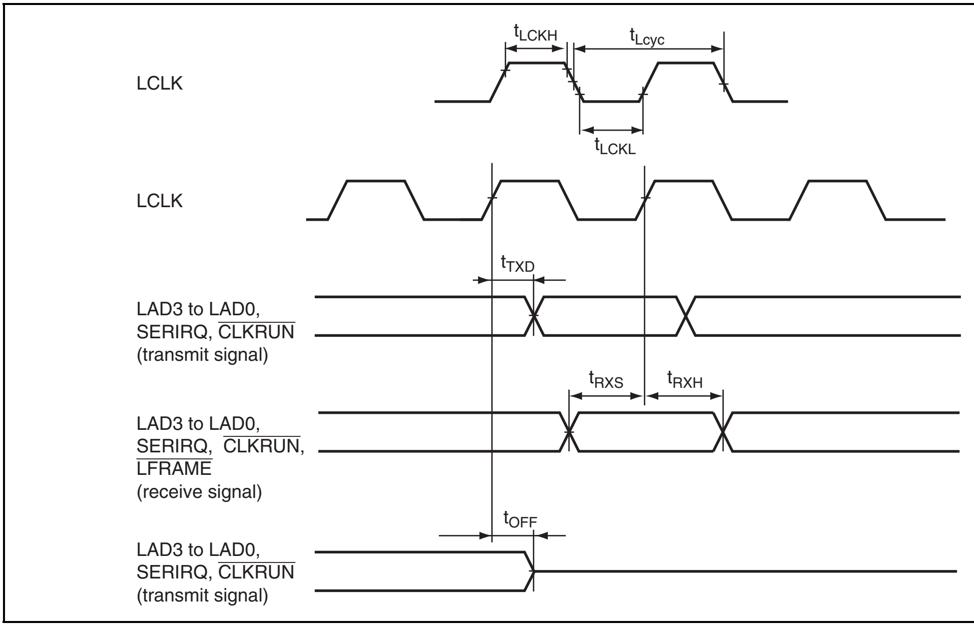
Note: * 17.5 t_{cyc} can be set according to the clock selected for use by the I^{2}C module.



Note: * S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 28.24 I^{2}C Bus Interface Input/Output Timing

Transmit signal floating delay time	t_{OFF}	—	—	28
Receive signal setup time	t_{RXS}	7	—	—
Receive signal hold time	t_{RXH}	0	—	—



t_{LCKH}
t_{Lcyc}
LCLK
t_{LCKL}
LCLK
t_{TXD}
LAD3 to LAD0, SERIRQ, $\overline{\text{CLKRUN}}$ (transmit signal)
t_{RXS}
t_{RXH}
LAD3 to LAD0, SERIRQ, $\overline{\text{CLKRUN}}$, $\overline{\text{LFRAME}}$ (receive signal)
t_{OFF}
LAD3 to LAD0, SERIRQ, $\overline{\text{CLKRUN}}$ (transmit signal)

Figure 28.25 LPC Interface Timing

Item	Symbol	Min.	Max.	Unit	T C
ETCK clock cycle time	t_{TCKcyc}	50*	125*	ns	F 2
ETCK clock high pulse width	t_{TCKH}	20	—		
ETCK clock low pulse width	t_{TCKL}	20	—		
ETCK clock rise time	t_{TCKr}	—	5		
ETCK clock fall time	t_{TCKf}	—	5		
$\overline{\text{ETRST}}$ pulse width	t_{TRSTW}	20	—	t_{cyc}	F 2
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	F 2
ETMS hold time	t_{TMSH}	20	—		
ETDI setup time	t_{TDIS}	20	—		
ETDI hold time	t_{TDIH}	20	—		
ETDO data delay time	t_{TDOD}	—	20		

Note: * When $t_{cyc} \le t_{TCKcyc}$

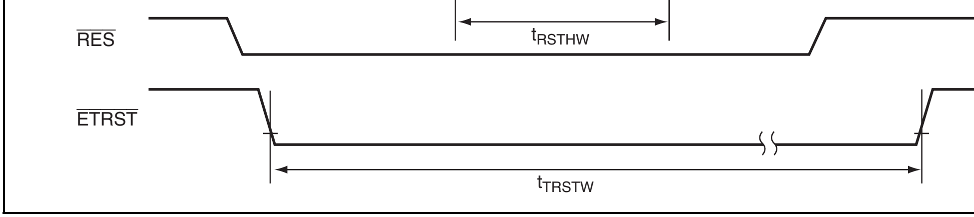


Figure 28.28 Reset Hold Timing

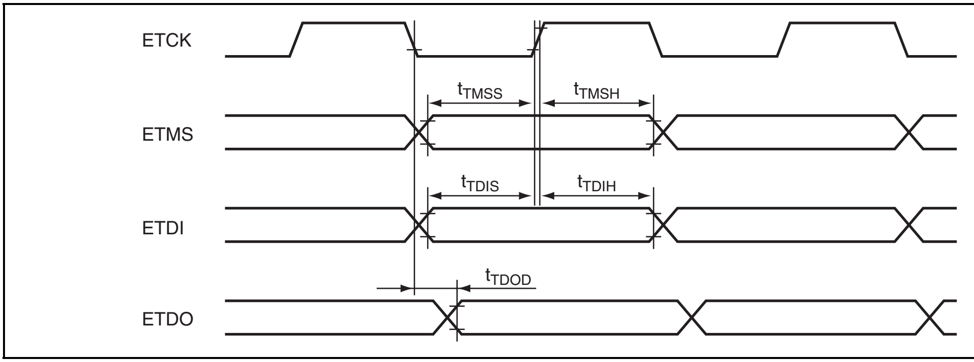


Figure 28.29 JTAG Input/Output Timing

Item	Min.	Typ.	Max.
Resolution	10		
Conversion time	—	—	4.0*
Analog input capacitance	—	—	20
Permissible signal-source impedance	—	—	5
Nonlinearity error	—	—	±7.0
Offset error	—	—	±7.5
Full-scale error	—	—	±7.5
Quantization error	—	—	±0.5
Absolute accuracy	—	—	±8.0

Note: The power supply to Avref must either be made simultaneously with or follow the supply to Avcc.

* Value when using the maximum operating frequency of 40 states (ADCLK = 1

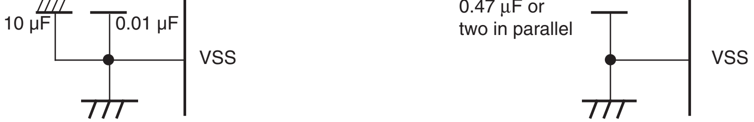
Programming time*[1]*[2]*[4]	t_P	—	1	10	ms/128 bytes	
Erase time*[1]*[2]*[4]	t_E	—	40	130	ms/4-Kbyte block	
		—	300	800	ms/32-Kbyte block	
		—	600	1500	ms/64-Kbyte block	
Programming time (total)*[1]*[2]*[4]	Σ_{tP}	—	1.4	4	s/160 Kbytes	Ta = 25
Erase time (total)*[1]*[2]*[4]	Σ_{tE}	—	1.4	4	s/160 Kbytes	Ta = 25
Programming and Erase time (total)*[1]*[2]*[4]	Σ_{tPE}	—	2.9	8	s/160 Kbytes	Ta = 25
Reprogramming count	N_{WEC}	100*[3]	1000	—	Times	
Data retention time*[4]	t_{DRP}	10	—	—	Years	

Notes: 1. Programming and erase time depends on the data.

2. Programming and erase time do not include data transfer time.

3 This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value rang from 1 to the minimum number.)

4. This value indicates the characteristics while the flash memory is reprogramme the specified range (including the minimum number).



It is recommended that a bypass capacitor be connected to the VCC pin. (The values are reference values.)
When connecting, place a bypass capacitor near the pin.

Do not connect Vcc power supply to the VCL pin.
Always connect a capacitor for internal step-dowr
stabilization. Use one or two ceramic multilayer ca
(0.1 µF / 0.47 µF: connect in parallel when using
and place it (them) near the pin.

Figure 28.30 Connection of VCC and VCL Capacitors

Port 3	T	keep	keep	keep	I/O po
Port 4	T	keep	keep	keep	I/O po
Ports 52 to 50	T	keep	keep	keep	I/O po
Port 6	T	keep	keep	keep	I/O po
Ports 7 and E4 to E1	T	T	T	T	Input
Port 8	T	keep	keep	keep	I/O po
Port 97	T	keep	keep	keep	I/O po
Port 96 ϕ, EXCL	T	[DDR = 1]H [DDR = 0]T	EXCL input/ keep	[DDR = 1] Clock output [DDR = 0]T	Clock EXCL Input
Ports 95 to 90	T	keep	keep	keep	I/O po
Ports A to D, F, G, and H5 to H0	T	keep	keep	keep	I/O po
Port E0	T	T	ExEXCL input/T	T	ExEX input
Port I	T	keep	keep	keep	I/O po
Port J	T	keep	keep	keep	I/O po

[Legend]

H: High level

L: Low level

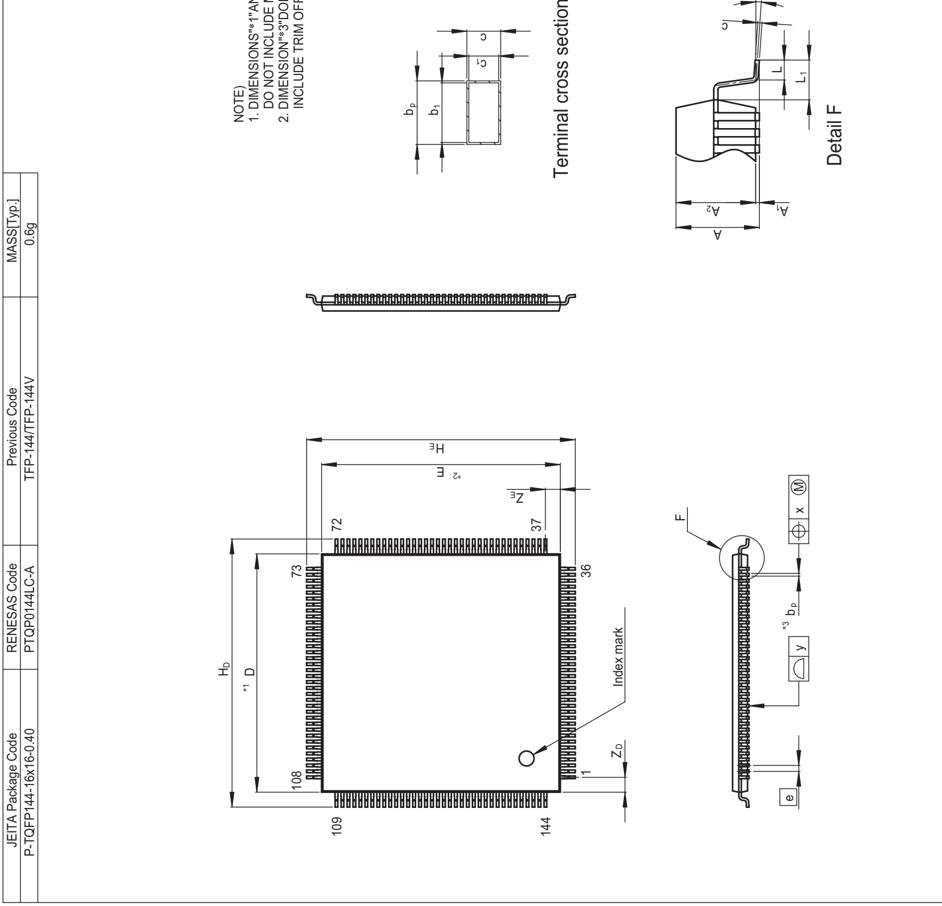
T: High impedance

keep: Input ports are in the high-impedance state (when DDR = 0 and PCR = 1, the inp MOS remains on).

Output ports maintain their previous state.

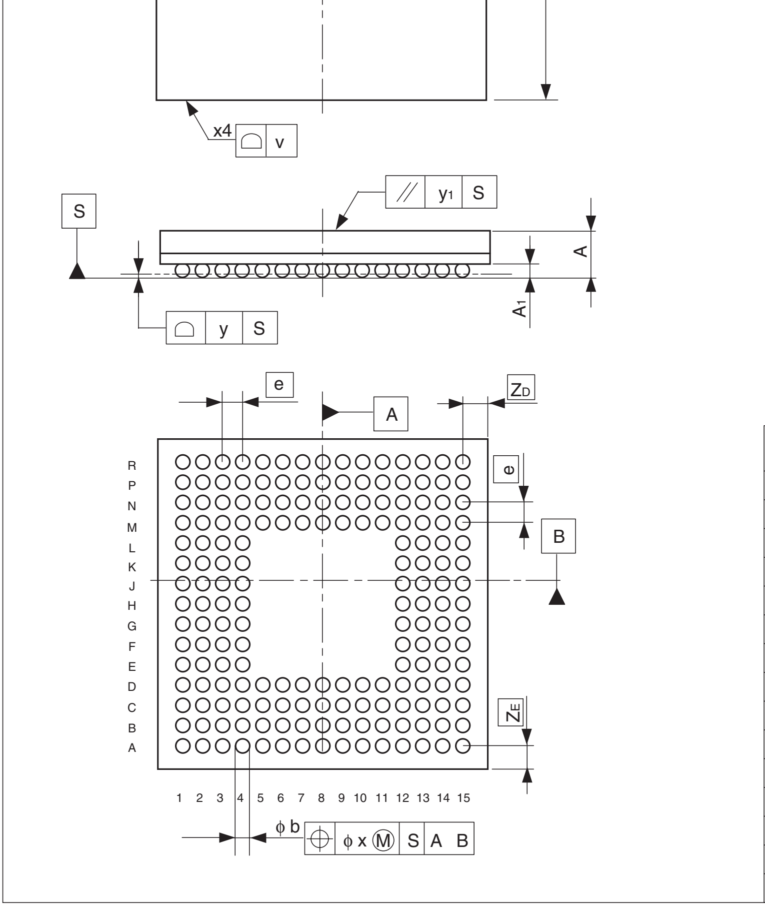
Depending on the pins, the on-chip peripheral modules may be initialized and the function determined by DDR and DR.

DDR: Data direction register



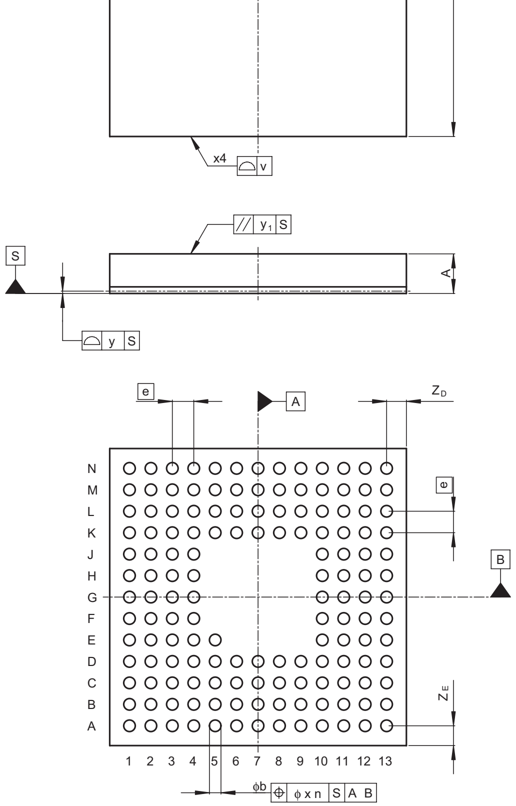
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TQFP144-16x16-0.40	PTQP0144LC-A	TFP-144/TFP-144V	0.6g

Figure C.1 Package Dimensions (TFP-144V)



Reference Symbol	Dimension i	
	Min	No
D	——	13
E	——	13
v	——	—
w	——	—
A	——	—
A_1	0.35	0.4
e	——	0.8
b	0.45	0.5
x	——	—
y	——	—
y_1	——	—
S_D	——	—
S_E	——	—
Z_D	——	0.9
Z_E	——	0.9

Figure C.2 Package Dimensions (BP-176V)



Reference Symbol	Dimension in Millimete		
	Min	Nom	Ma
D	——	9.0	——
E	——	9.0	——
v	——	——	0.1
w	——	——	0.2
A	——	——	1.2
A_1	——	——	——
e	——	0.65	——
b	0.30	0.35	0.4
x	——	——	0.0
y	——	——	0.
y_1	——	——	0.2
S_D	——	——	——
S_E	——	——	——
Z_D	——	0.6	——
Z_E	——	0.6	——

Figure C.3 Package Dimensions (TLP-145V)

NMI	• Connect to V_{CC} via a pull-up resistor
EXTAL	(Always used as a clock pin)
XTAL	(Always used as a clock pin)
Port 1 Port 2 Port 3 Port 4 Port 5 Port 6 Port 8 Port 9 Port A Port B Port C Port D Port F Port G Port H Port I Port J	• Connect each pin to V_{CC} via a pull-up resistor or to V_{SS} via a pull-dow resistor
Port 7	• Connect each pin to AV_{CC} via a pull-up resistor or to AV_{SS} via a pull-d resistor
Port E	• Connect each pin to V_{CC} via a pull-up resistor

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2117R Group

Publication Date:	Rev.1.00, April 28, 2008
	Rev.2.00, September 28, 2009
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, J

RENESAS

RENESAS SALES OFFICES

http://www.ren

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) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Ma
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2117R Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan