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

M16C/6N Group (M16C/6NL, M16C/6NN)

Hardware Manual

RENESAS MCU
M16C FAMILY / M16C/60 SERIES

All information contained in this material, including products and product specifications at the time of publication of this material, 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

Renesas Electronics
www.renesas.com

Rev.2.10 Apr 2006

Keep safety first in your circuit designs!

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

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
- Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.

  The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.

  Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.

  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should 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 register 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, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power 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 functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

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

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

5. Differences between Products

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

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

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual. The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the M16C/6N Group (M16C/6NL, M16C/6NN). Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Technology Web site.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	M16C/6N Group (M16C/6NL, M16C/6NN) Datasheet	REJ03B0061
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	M16C/6N Group (M16C/6NL. M16C/6NN) Hardware Manual	This hardware manual (REJ09B0126)
Software manual	Description of CPU instruction set	M16C/60, M16C/20, M16C/Tiny Series Software Manual	REJ09B0137
Application note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Technology web site	
Renesas technical update	Product specifications, updates on documents, etc.		

2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word “register,” “bit,” or “pin” to distinguish the three categories.

Examples the PM03 bit in the PM0 register
P3_5 pin, VCC pin

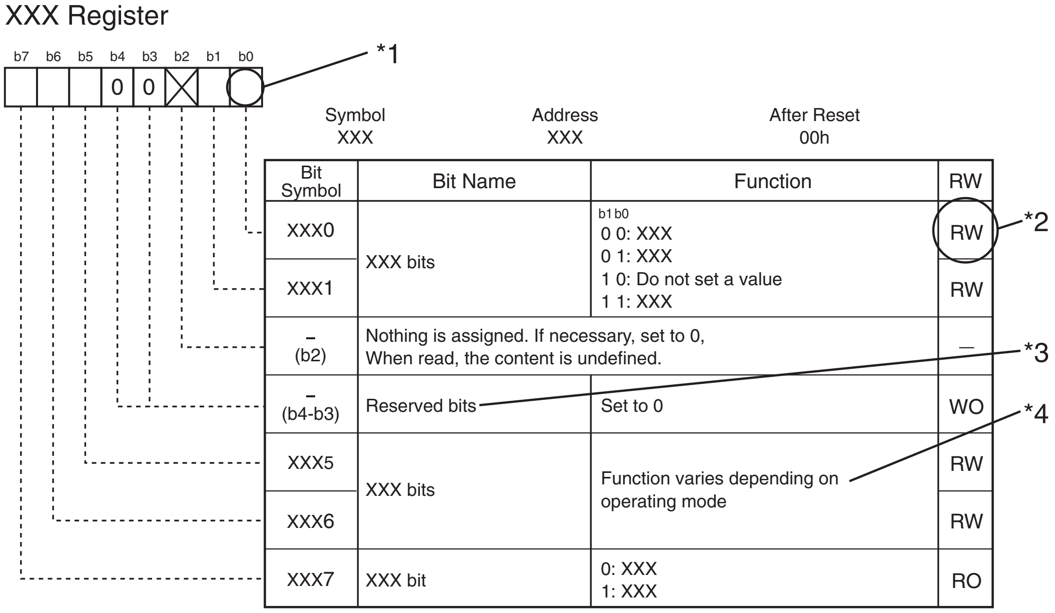
(2) Notation of Numbers

The indication “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b
Hexadecimal: EFA0h
Decimal: 1234

3. Register Notation

The symbols and terms used in register diagrams are described below.



Symbol XXX   Address XXX   After Reset 00h

Bit Symbol	Bit Name	Function	RW
XXX0	XXX bits	b1 b0 0 0: XXX 0 1: XXX 1 0: Do not set a value 1 1: XXX	RW
XXX1			RW
– (b2)	Nothing is assigned. If necessary, set to 0, When read, the content is undefined.		–
– (b4-b3)	Reserved bits	Set to 0	WO
XXX5	XXX bits	Function varies depending on operating mode	RW
XXX6			RW
XXX7	XXX bit	0: XXX 1: XXX	RO

*1

Blank: Set to 0 or 1 according to the application
0 : Set to 0
1 : Set to 1
X : Nothing is assigned

*2

RW : Read and write
RO : Read only
WO : Write only
– : Nothing is assigned

*3

- Reserved bit
  Reserved bit. Set to specified value.

*4

- Nothing is assigned
  Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.
- Do not set a value
  Operation is not guaranteed when a value is set.
- Function varies depending on operating mode
  The function of the bit varies with the peripheral function mode.
  Refer to the register diagram for information on the individual modes.

4. List of Abbreviations and Acronyms

Abbreviation	Full Form
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
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connection
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

Table of Contents

Specifications written in this manual are believed to be accurate, but are not guaranteed to be entirely free of error. Specifications in this manual may be changed for functional or performance improvements. Please make sure your manual is the latest edition.

SFR Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	35
0005h	Processor Mode Register 1	PM1	36
0006h	System Clock Control Register 0	CM0	53
0007h	System Clock Control Register 1	CM1	54
0008h	Chip Select Control Register	CSR	41
0009h	Address Match Interrupt Enable Register	AIER	95
000Ah	Protect Register	PRCR	75
000Bh			
000Ch	Oscillation Stop Detection Register	CM2	55
000Dh			
000Eh	Watchdog Timer Start Register	WDTS	97
000Fh	Watchdog Timer Control Register	WDC	97
0010h	Address Match Interrupt Register 0	RMAD0	95
0011h			
0012h			
0013h			
0014h	Address Match Interrupt Register 1	RMAD1	95
0015h			
0016h			
0017h			
0018h			
0019h			
001Ah			
001Bh	Chip Select Expansion Control Register	CSE	47
001Ch	PLL Control Register 0	PLC0	58
001Dh			
001Eh	Processor Mode Register 2	PM2	57
001Fh			
0020h	DMA0 Source Pointer	SAR0	102
0021h			
0022h			
0023h			
0024h	DMA0 Destination Pointer	DAR0	102
0025h			
0026h			
0027h			
0028h	DMA0 Transfer Counter	TCR0	102
0029h			
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	101
002Dh			
002Eh			
002Fh			
0030h	DMA1 Source Pointer	SAR1	102
0031h			
0032h			
0033h			
0034h	DMA1 Destination Pointer	DAR1	102
0035h			
0036h			
0037h			
0038h	DMA1 Transfer Counter	TCR1	102
0039h			
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	101
003Dh			
003Eh			
003Fh			
0040h			
0041h	CAN0 Wake-up Interrupt Control Register	C01WKIC	81
0042h	CAN0 Successful Reception Interrupt Control Register	C0RECIC	81
0043h	CAN0 Successful Transmission Interrupt Control Register	C0TRMIC	81
0044h	INT3 Interrupt Control Register	INT3IC	82
0045h	Timer B5 Interrupt Control Register	TB5IC	81
	SI/O5 Interrupt Control Register	S5IC	81
0046h	Timer B4 Interrupt Control Register	TB4IC	81
	UART1 Bus Collision Detection Interrupt Control Register	U1BCNIC	81
0047h	Timer B3 Interrupt Control Register	TB3IC	81
	UART0 Bus Collision Detection Interrupt Control Register	U0BCNIC	81
0048h	SI/O4 Interrupt Control Register	S4IC	82
	INT5 Interrupt Control Register	INT5IC	82
0049h	SI/O3 Interrupt Control Register	S3IC	82
	INT4 Interrupt Control Register	INT4IC	82
004Ah	UART2 Bus Collision Detection Interrupt Control Register	U2BCNIC	81
004Bh	DMA0 Interrupt Control Register	DM0IC	81
004Ch	DMA1 Interrupt Control Register	DM1IC	81
004Dh	CAN0 Error Interrupt Control Register	C01ERRIC	81
004Eh	A/D Conversion Interrupt Control Register	ADIC	81
	Key Input Interrupt Control Register	KUPIC	81
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	81
0050h	UART2 Receive Interrupt Control Register	S2RIC	81
0051h	UART0 Transmit Interrupt Control Register	S0TIC	81
0052h	UART0 Receive Interrupt Control Register	S0RIC	81
0053h	UART1 Transmit Interrupt Control Register	S1TIC	81
0054h	UART1 Receive Interrupt Control Register	S1RIC	81
0055h	Timer A0 Interrupt Control Register	TA0IC	81
0056h	Timer A1 Interrupt Control Register	TA1IC	81
0057h	Timer A2 Interrupt Control Register	TA2IC	82
	INT7 Interrupt Control Register	INT7IC	82
0058h	Timer A3 Interrupt Control Register	TA3IC	82
	INT6 Interrupt Control Register	INT6IC	82
0059h	Timer A4 Interrupt Control Register	TA4IC	81
005Ah	Timer B0 Interrupt Control Register	TB0IC	81
	SI/O6 Interrupt Control Register	S6IC	81
005Bh	Timer B1 Interrupt Control Register	TB1IC	82
	INT8 Interrupt Control Register	INT8IC	82
005Ch	Timer B2 Interrupt Control Register	TB2IC	81
005Dh	INT0 Interrupt Control Register	INT0IC	82
005Eh	INT1 Interrupt Control Register	INT1IC	82
005Fh	INT2 Interrupt Control Register	INT2IC	82
0060h	CAN0 Message Box 0: Identifier / DLC		220 221
0061h			
0062h			
0063h			
0064h			
0065h			
0066h	CAN0 Message Box 0: Data Field		
0067h			
0068h			
0069h			
006Ah			
006Bh			
006Ch			
006Dh			
006Eh	CAN0 Message Box 0: Time Stamp		
006Fh			
0070h	CAN0 Message Box 1: Identifier / DLC		
0071h			
0072h			
0073h			
0074h			
0075h			
0076h	CAN0 Message Box 1: Data Field		
0077h			
0078h			
0079h			
007Ah			
007Bh			
007Ch			
007Dh			
007Eh	CAN0 Message Box 1: Time Stamp		
007Fh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0080h	CAN0 Message Box 2: Identifier / DLC		220 221
0081h			
0082h			
0083h			
0084h			
0085h			
0086h	CAN0 Message Box 2: Data Field		
0087h			
0088h			
0089h			
008Ah			
008Bh			
008Ch			
008Dh			
008Eh	CAN0 Message Box 2: Time Stamp		
008Fh			
0090h	CAN0 Message Box 3: Identifier / DLC		
0091h			
0092h			
0093h			
0094h			
0095h			
0096h	CAN0 Message Box 3: Data Field		
0097h			
0098h			
0099h			
009Ah			
009Bh			
009Ch			
009Dh			
009Eh	CAN0 Message Box 3: Time Stamp		
009Fh			
00A0h	CAN0 Message Box 4: Identifier / DLC		
00A1h			
00A2h			
00A3h			
00A4h			
00A5h			
00A6h	CAN0 Message Box 4: Data Field		
00A7h			
00A8h			
00A9h			
00AAh			
00ABh			
00ACh			
00ADh			
00AEh	CAN0 Message Box 4: Time Stamp		
00AFh			
00B0h	CAN0 Message Box 5: Identifier / DLC		
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h	CAN0 Message Box 5: Data Field		
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh	CAN0 Message Box 5: Time Stamp		
00BFh			

Address	Register	Symbol	Page
00C0h	CAN0 Message Box 6: Identifier / DLC		220 221
00C1h			
00C2h			
00C3h			
00C4h			
00C5h			
00C6h	CAN0 Message Box 6: Data Field		
00C7h			
00C8h			
00C9h			
00CAh			
00CBh			
00CCh			
00CDh			
00CEh	CAN0 Message Box 6: Time Stamp		
00CFh			
00D0h	CAN0 Message Box 7: Identifier / DLC		
00D1h			
00D2h			
00D3h			
00D4h			
00D5h			
00D6h	CAN0 Message Box 7: Data Field		
00D7h			
00D8h			
00D9h			
00DAh			
00DBh			
00DCh			
00DDh			
00DEh	CAN0 Message Box 7: Time Stamp		
00DFh			
00E0h	CAN0 Message Box 8: Identifier / DLC		
00E1h			
00E2h			
00E3h			
00E4h			
00E5h			
00E6h	CAN0 Message Box 8: Data Field		
00E7h			
00E8h			
00E9h			
00EAh			
00EBh			
00ECh			
00EDh			
00EEh	CAN0 Message Box 8: Time Stamp		
00EFh			
00F0h	CAN0 Message Box 9: Identifier / DLC		
00F1h			
00F2h			
00F3h			
00F4h			
00F5h			
00F6h	CAN0 Message Box 9: Data Field		
00F7h			
00F8h			
00F9h			
00FAh			
00FBh			
00FCh			
00FDh			
00FEh	CAN0 Message Box 9: Time Stamp		
00FFh			

Address	Register	Symbol	Page
0100h	CAN0 Message Box 10: Identifier / DLC		220 221
0101h			
0102h			
0103h			
0104h			
0105h			
0106h	CAN0 Message Box 10: Data Field		
0107h			
0108h			
0109h			
010Ah			
010Bh			
010Ch			
010Dh			
010Eh	CAN0 Message Box 10: Time Stamp		
010Fh			
0110h	CAN0 Message Box 11: Identifier / DLC		
0111h			
0112h			
0113h			
0114h			
0115h			
0116h	CAN0 Message Box 11: Data Field		
0117h			
0118h			
0119h			
011Ah			
011Bh			
011Ch			
011Dh			
011Eh	CAN0 Message Box 11: Time Stamp		
011Fh			
0120h	CAN0 Message Box 12: Identifier / DLC		
0121h			
0122h			
0123h			
0124h			
0125h			
0126h	CAN0 Message Box 12: Data Field		
0127h			
0128h			
0129h			
012Ah			
012Bh			
012Ch			
012Dh			
012Eh	CAN0 Message Box 12: Time Stamp		
012Fh			
0130h	CAN0 Message Box 13: Identifier / DLC		
0131h			
0132h			
0133h			
0134h			
0135h			
0136h	CAN0 Message Box 13: Data Field		
0137h			
0138h			
0139h			
013Ah			
013Bh			
013Ch			
013Dh			
013Eh	CAN0 Message Box 13: Time Stamp		
013Fh			

Address	Register	Symbol	Page
0140h	CAN0 Message Box 14: Identifier /DLC		220 221
0141h			
0142h			
0143h			
0144h			
0145h			
0146h	CAN0 Message Box 14: Data Field		
0147h			
0148h			
0149h			
014Ah			
014Bh			
014Ch			
014Dh			
014Eh	CAN0 Message Box 14: Time Stamp		
014Fh			
0150h	CAN0 Message Box 15: Identifier /DLC		
0151h			
0152h			
0153h			
0154h			
0155h			
0156h	CAN0 Message Box 15: Data Field		
0157h			
0158h			
0159h			
015Ah			
015Bh			
015Ch			
015Dh			
015Eh	CAN0 Message Box 15: Time Stamp		
015Fh			
0160h	CAN0 Global Mask Register	C0GMR	222
0161h			
0162h			
0163h			
0164h			
0165h			
0166h	CAN0 Local Mask A Register	C0LMAR	222
0167h			
0168h			
0169h			
016Ah			
016Bh			
016Ch	CAN0 Local Mask B Register	C0LMBR	222
016Dh			
016Eh			
016Fh			
0170h			
0171h			
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0180h			
0181h			
0182h			
0183h			
0184h			
0185h			
0186h			
0187h			
0188h			
0189h			
018Ah			
018Bh			
018Ch			
018Dh			
018Eh			
018Fh			
0190h			
0191h			
0192h			
0193h			
0194h			
0195h			
0196h			
0197h			
0198h			
0199h			
019Ah			
019Bh			
019Ch			
019Dh			
019Eh			
019Fh			
01A0h			
01A1h			
01A2h			
01A3h			
01A4h			
01A5h			
01A6h			
01A7h			
01A8h			
01A9h			
01AAh			
01ABh			
01ACh			
01ADh			
01AEh			
01AFh			
01B0h			
01B1h			
01B2h			
01B3h			
01B4h			
01B5h	Flash Memory Control Register 1	FMR1	262
01B6h			
01B7h	Flash Memory Control Register 0	FMR0	262
01B8h	Address Match Interrupt Register 2	RMAD2	95
01B9h			
01BAh			
01BBh	Address Match Interrupt Enable Register 2	AIER2	95
01BCh	Address Match Interrupt Register 3	RMAD3	95
01BDh			
01BEh			
01BFh			

Address	Register	Symbol	Page
01C0h	Timer B3, B4, B5 Count Start Flag	TBSR	127
01C1h			
01C2h	Timer A1-1 Register	TA11	138
01C3h			
01C4h	Timer A2-1 Register	TA21	138
01C5h			
01C6h	Timer A4-1 Register	TA41	138
01C7h			
01C8h	Three-Phase PWM Control Register 0	INVC0	135
01C9h	Three-Phase PWM Control Register 1	INVC1	136
01CAh	Three-Phase Output Buffer Register 0	IDB0	137
01CBh	Three-Phase Output Buffer Register 1	IDB1	137
01CCh	Dead Time Timer	DTT	137
01CDh	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	139
01CEh			
01CFh	Interrupt Cause Select Register 2	IFSR2	92
01D0h	Timer B3 Register	TB3	136
01D1h			
01D2h	Timer B4 Register	TB4	126
01D3h			
01D4h	Timer B5 Register	TB5	126
01D5h			
01D6h	SI/O6 Transmit/Receive Register	S6TRR	192
01D7h			
01D8h	SI/O6 Control Register	S6C	192
01D9h	SI/O6 Bit Rate Register	S6BRG	192
01DAh	SI/O3, 4, 5, 6 Transmit/Receive Register	S3456TRR	193
01DBh	Timer B3 Mode Register	TB3MR	126 128 129 131
01DCh	Timer B4 Mode Register	TB4MR	
01DDh	Timer B5 Mode Register	TB5MR	
01DEh	Interrupt Source Select Register 0	IFSR0	90
01DFh	Interrupt Source Select Register 1	IFSR1	91
01E0h	SI/O3 Transmit/Receive Register	S3TRR	192
01E1h			
01E2h	SI/O3 Control Register	S3C	192
01E3h	SI/O3 Bit Rate Register	S3BRG	192
01E4h	SI/O4 Transmit/Receive Register	S4TRR	192
01E5h			
01E6h	SI/O4 Control Register	S4C	192
01E7h	SI/O4 Bit Rate Register	S4BRG	192
01E8h	SI/O5 Transmit/Receive Register	S5TRR	192
01E9h			
01EAh	SI/O5 Control Register	S5C	192
01EBh	SI/O5 Bit Rate Register	S5BRG	192
01ECh	UART0 Special Mode Register 4	U0SMR4	153
01EDh	UART0 Special Mode Register 3	U0SMR3	152
01EEh	UART0 Special Mode Register 2	U0SMR2	152
01EFh	UART0 Special Mode Register	U0SMR	151
01F0h	UART1 Special Mode Register 4	U1SMR4	153
01F1h	UART1 Special Mode Register 3	U1SMR3	152
01F2h	UART1 Special Mode Register 2	U1SMR2	152
01F3h	UART1 Special Mode Register	U1SMR	151
01F4h	UART2 Special Mode Register 4	U2SMR4	153
01F5h	UART2 Special Mode Register 3	U2SMR3	152
01F6h	UART2 Special Mode Register 2	U2SMR2	152
01F7h	UART2 Special Mode Register	U2SMR	151
01F8h	UART2 Transmit/Receive Mode Register	U2MR	149
01F9h	UART2 Bit Rate Register	U2BRG	148
01FAh	UART2 Transmit Buffer Register	U2TB	148
01FBh			
01FCh	UART2 Transmit/Receive Control Register 0	U2C0	149
01FDh	UART2 Transmit/Receive Control Register 1	U2C1	150
01FEh	UART2 Receive Buffer Register	U2RB	148
01FFh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0200h	CAN0 Message Control Register 0	C0MCTL0	223
0201h	CAN0 Message Control Register 1	C0MCTL1	
0202h	CAN0 Message Control Register 2	C0MCTL2	
0203h	CAN0 Message Control Register 3	C0MCTL3	
0204h	CAN0 Message Control Register 4	C0MCTL4	
0205h	CAN0 Message Control Register 5	C0MCTL5	
0206h	CAN0 Message Control Register 6	C0MCTL6	
0207h	CAN0 Message Control Register 7	C0MCTL7	
0208h	CAN0 Message Control Register 8	C0MCTL8	
0209h	CAN0 Message Control Register 9	C0MCTL9	
020Ah	CAN0 Message Control Register 10	C0MCTL10	
020Bh	CAN0 Message Control Register 11	C0MCTL11	
020Ch	CAN0 Message Control Register 12	C0MCTL12	
020Dh	CAN0 Message Control Register 13	C0MCTL13	
020Eh	CAN0 Message Control Register 14	C0MCTL14	
020Fh	CAN0 Message Control Register 15	C0MCTL15	
0210h	CAN0 Control Register	C0CTLR	224
0211h			
0212h	CAN0 Status Register	C0STR	226
0213h			
0214h	CAN0 Slot Status Register	C0SSTR	227
0215h			
0216h	CAN0 Interrupt Control Register	C0ICR	227
0217h			
0218h	CAN0 Extended ID Register	C0IDR	227
0219h			
021Ah	CAN0 Configuration Register	C0CONR	228
021Bh			
021Ch	CAN0 Receive Error Count Register	C0RECR	229
021Dh	CAN0 Transmit Error Count Register	C0TECR	229
021Eh	CAN0 Time Stamp Register	C0TSR	229
021Fh			
0220h			
0221h			
0222h			
0223h			
0224h			
0225h			
0226h			
0227h			
0228h			
0229h			
022Ah			
022Bh			
022Ch			
022Dh			
022Eh			
022Fh			
0230h	CAN1 Control Register	C1CTLR	225
0231h			
0232h			
0233h			
0234h			
0235h			
0236h			
0237h			
0238h			
0239h			
023Ah			
023Bh			
023Ch			
023Dh			
023Eh			
023Fh			

Address	Register	Symbol	Page
0240h			
0241h			
0242h	CAN0 Acceptance Filter Support Register	C0AFS	229
0243h			
0244h			
0245h			
0246h			
0247h			
0248h			
0249h			
024Ah			
024Bh			
024Ch			
024Dh			
024Eh			
024Fh			
0250h			
0251h			
0252h			
0253h			
0254h			
0255h			
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh	Peripheral Clock Select Register	PCLKR	56
025Fh	CAN0 Clock Select Register	CCLKR	57
0260h			
0261h			
0262h			
0263h			
0264h			
0265h			
0266h			
0267h			
0268h			
0269h			
026Ah			
026Bh			
026Ch			
026Dh			
026Eh			
026Fh			
0270h to 0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h			
0379h			
037Ah			
037Bh			
037Ch			
037Dh			
037Eh			
037Fh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0380h	Count Start Flag	TABSR	112,127,140
0381h	Clock Prescaler Reset Flag	CPSRF	113,127
0382h	One-Shot Start Flag	ONSF	113
0383h	Trigger Select Register	TRGSR	113,140
0384h	Up/Down Flag	UDF	112
0385h			
0386h	Timer A0 Register	TA0	111
0387h			
0388h	Timer A1 Register	TA1	111
0389h			138
038Ah	Timer A2 Register	TA2	111
038Bh			138
038Ch	Timer A3 Register	TA3	111
038Dh			
038Eh	Timer A4 Register	TA4	111
038Fh			138
0390h	Timer B0 Register	TB0	126
0391h			
0392h	Timer B1 Register	TB1	126
0393h			
0394h	Timer B2 Register	TB2	126
0395h			138
0396h	Timer A0 Mode Register	TA0MR	111
0397h	Timer A1 Mode Register	TA1MR	114 141
0398h	Timer A2 Mode Register	TA2MR	116 118,141
0399h	Timer A3 Mode Register	TA3MR	121 118
039Ah	Timer A4 Mode Register	TA4MR	123 118,141
039Bh	Timer B0 Mode Register	TB0MR	126,128
039Ch	Timer B1 Mode Register	TB1MR	129,131
039Dh	Timer B2 Mode Register	TB2MR	141
039Eh	Timer B2 Special Mode Register	TB2SC	139
039Fh			
03A0h	UART0 Transmit/Receive Mode Register	U0MR	149
03A1h	UART0 Bit Rate Register	U0BRG	148
03A2h	UART0 Transmit Buffer Register	U0TB	148
03A3h			
03A4h	UART0 Transmit/Receive Control Register 0	U0C0	149
03A5h	UART0 Transmit/Receive Control Register 1	U0C1	150
03A6h	UART0 Receive Buffer Register	U0RB	148
03A7h			
03A8h	UART1 Transmit/Receive Mode Register	U1MR	149
03A9h	UART1 Bit Rate Register	U1BRG	148
03AAh	UART1 Transmit Buffer Register	U1TB	148
03ABh			
03ACh	UART1 Transmit/Receive Control Register 0	U1C0	149
03ADh	UART1 Transmit/Receive Control Register 1	U1C1	150
03AEh	UART1 Receive Buffer Register	U1RB	148
03AFh			
03B0h	UART Transmit/Receive Control Register 2	UCON	151
03B1h			
03B2h			
03B3h			
03B4h			
03B5h			
03B6h			
03B7h			
03B8h	DMA0 Request Source Select Register	DM0SL	100
03B9h			
03BAh	DMA1 Request source Select Register	DM1SL	101
03BBh			
03BCh	CRC Data Register	CRCD	216
03BDh			
03BEh	CRC Input Register	CRCIN	216
03BFh			

Address	Register	Symbol	Page
03C0h	A/D Register 0	AD0	200
03C1h			
03C2h	A/D Register 1	AD1	
03C3h			
03C4h	A/D Register 2	AD2	
03C5h			
03C6h	A/D Register 3	AD3	
03C7h			
03C8h	A/D Register 4	AD4	
03C9h			
03CAh	A/D Register 5	AD5	
03CBh			
03CCh	A/D Register 6	AD6	
03CDh			
03CEh	A/D Register 7	AD7	
03CFh			
03D0h			
03D1h			
03D2h			
03D3h			
03D4h	A/D Control Register 2	ADCON2	200
03D5h			
03D6h	A/D Control Register 0	ADCON0	199,202,204
03D7h	A/D Control Register 1	ADCON1	206,208,210
03D8h	D/A Register 0	DA0	215
03D9h			
03DAh	D/A Register 1	DA1	215
03DBh			
03DCh	D/A Control Register	DACON	215
03DDh			
03DEh	Port P14 Control Register	PC14	251
03DFh	Pull-Up Control Register 3	PUR3	253
03E0h	Port P0 Register	P0	251
03E1h	Port P1 Register	P1	251
03E2h	Port P0 Direction Register	PD0	250
03E3h	Port P1 Direction Register	PD1	250
03E4h	Port P2 Register	P2	251
03E5h	Port P3 Register	P3	251
03E6h	Port P2 Direction Register	PD2	250
03E7h	Port P3 Direction Register	PD3	250
03E8h	Port P4 Register	P4	251
03E9h	Port P5 Register	P5	251
03EAh	Port P4 Direction Register	PD4	250
03EBh	Port P5 Direction Register	PD5	250
03ECh	Port P6 Register	P6	251
03EDh	Port P7 Register	P7	251
03EEh	Port P6 Direction Register	PD6	250
03EFh	Port P7 Direction Register	PD7	250
03F0h	Port P8 Register	P8	251
03F1h	Port P9 Register	P9	251
03F2h	Port P8 Direction Register	PD8	250
03F3h	Port P9 Direction Register	PD9	250
03F4h	Port P10 Register	P10	251
03F5h	Port P11 Register	P11	251
03F6h	Port P10 Direction Register	PD10	250
03F7h	Port P11 Direction Register	PD11	250
03F8h	Port P12 Register	P12	251
03F9h	Port P13 Register	P13	251
03FAh	Port P12 Direction Register	PD12	250
03FBh	Port P13 Direction Register	PD13	250
03FCh	Pull-up Control Register 0	PUR0	252
03FDh	Pull-up Control Register 1	PUR1	252
03FEh	Pull-up Control Register 2	PUR2	252
03FFh	Port Control Register	PCR	253

Blank spaces are reserved. No access is allowed.

M16C/6N Group (M16C/6NL, M16C/6NN)

Renesas MCU

1. Overview

The M16C/6N Group (M16C/6NL, M16C/6NN) of MCUs are built using the high-performance silicon gate CMOS process using the M16C/60 Series CPU core and are packaged in 100-pin and 128-pin plastic molded LQFP. These MCUs operate using sophisticated instructions featuring a high level of instruction efficiency. With 1 Mbyte of address space, they are capable of executing instructions at high speed. Being equipped with one CAN (Controller Area Network) module in the M16C/6N Group (M16C/6NL, M16C/6NN), the MCU is suited to drive automotive and industrial control systems. The CAN module complies with the 2.0B specification. In addition, this MCU contains a multiplier and DMAC which combined with fast instruction processing capability, makes it suitable for control of various OA, communication equipment which requires high-speed arithmetic/logic operations.

1.1 Applications

- Car audio and industrial control systems, other

1.2 Performance Overview

Tables 1.1 and 1.2 list the Functions and Specifications for M16C/6N Group (M16C/6NL, M16C/6NN).

Table 1.1 Functions and Specifications for M16C/6N Group (100-pin Version: M16C/6NL)

Item			Performance
CPU	Number of fundamental instructions		91 instructions
	Minimum instruction execution time		41.7ns (f(BCLK) = 24MHz, 1/1 prescaler, without software wait)
	Operating mode		Single-chip, memory expansion and microprocessor modes
	Address space		1 Mbyte
	Memory capacity		See **Table 1.3 Product List**
Peripheral Function	Ports		Input/Output: 87 pins, Input: 1 pin
	Multifunction timers		Timer A: 16 bits × 5 channels Timer B: 16 bits × 6 channels Three-phase motor control circuit
	Serial interfaces		3 channels Clock synchronous, UART, I^{2}C-bus [1], IEBus [2] 2 channels Clock synchronous
	A/D converter		10-bit A/D converter: 1 circuit, 26 channels
	D/A converter		8 bits × 2 channels
	DMAC		2 channels
	CRC calculation circuit		CRC-CCITT
	CAN module		1 channel with 2.0B specification
	Watchdog timer		15 bits × 1 channel (with prescaler)
	Interrupts		Internal: 30 sources, External: 9 sources Software: 4 sources, Priority level: 7 levels
	Clock generation circuits		4 circuits • Main clock oscillation circuit (*) • Sub clock oscillation circuit (*) • On-chip oscillator • PLL frequency synthesizer (*) Equipped with a built-in feedback resistor
	Oscillation-stopped detector		Main clock oscillation stop and re-oscillation detection function
Electrical Characteristics	Supply Voltage		VCC = 3.0 to 5.5V (f(BCLK) = 24MHz, 1/1 prescaler, without software wait)
	Consumption current	Mask ROM	19mA (f(BCLK) = 24MHz, PLL operation, no division)
		Flash memory	21mA (f(BCLK) = 24MHz, PLL operation, no division)
		Mask ROM Flash memory	3µA (f(BCLK) = 32kHz, Wait mode, Oscillation capacity Low)
			0.8µA (Stop mode, Topr = 25°C)
Flash Memory Version	Programming and erasure voltage		3.3 ± 0.3V or 5.0 ± 0.5V
	Programming and erasure endurance		100 times
I/O Characteristics	I/O withstand voltage		5.0V
	Output current		5mA
Operating Ambient Temperature			-40 to 85°C
Device Configuration			CMOS high performance silicon gate
Package			100-pin molded-plastic LQFP

NOTES:

1. I^{2}C-bus is a registered trademark of Koninklijke Philips Electronics N.V.
2. IEBus is a registered trademark of NEC Electronics Corporation.

Table 1.2 Functions and Specifications for M16C/6N Group (128-pin Version: M16C/6NN)

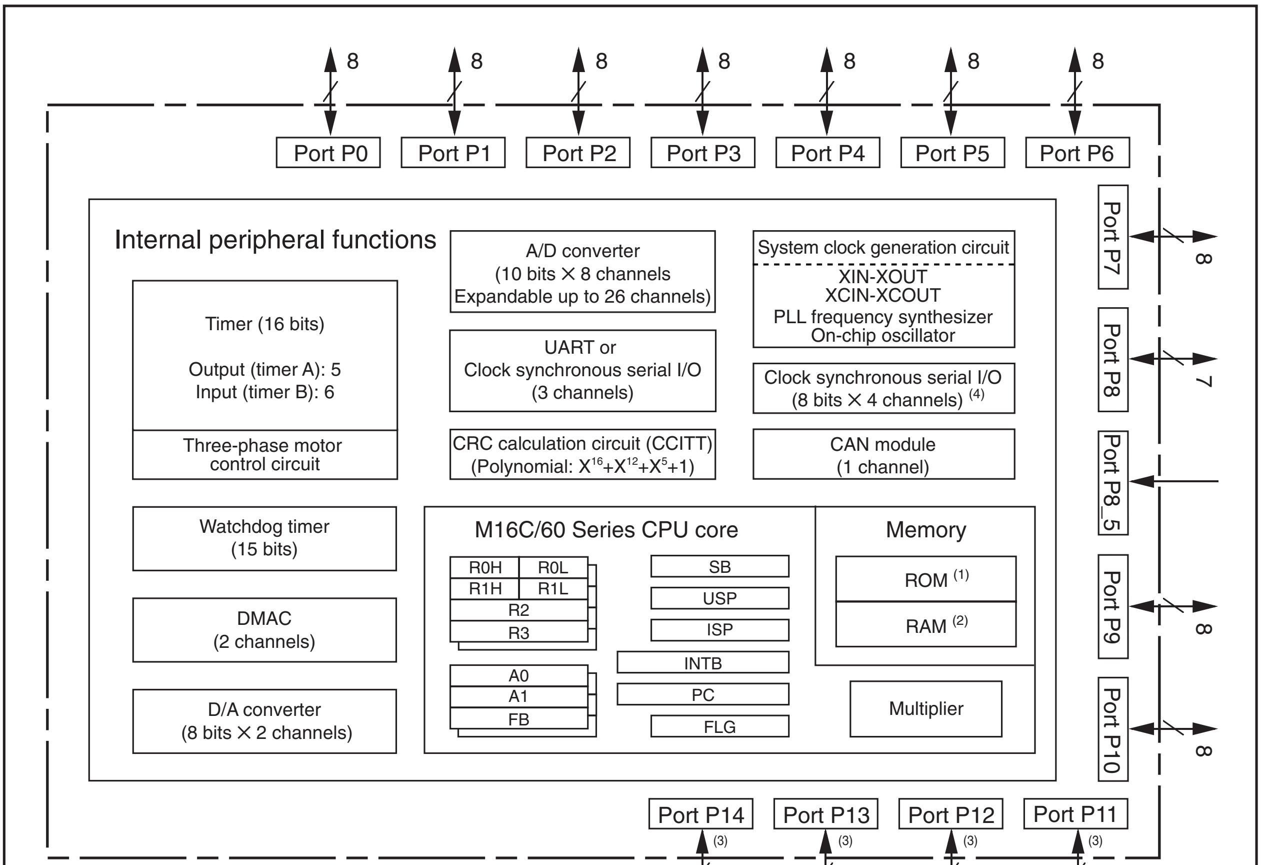
Item			Performance
CPU	Number of fundamental instructions		91 instructions
	Minimum instruction execution time		41.7ns (f(BCLK) = 24MHz, 1/1 prescaler, without software wait)
	Operating mode		Single-chip, memory expansion and microprocessor modes
	Address space		1 Mbyte
	Memory capacity		See **Table 1.3 Product List**
Peripheral Function	Ports		Input/Output: 113 pins, Input: 1 pin
	Multifunction timers		Timer A: 16 bits × 5 channels Timer B: 16 bits × 6 channels Three-phase motor control circuit
	Serial interfaces		3 channels Clock synchronous, UART, I^{2}C-bus [1], IEBus [2] 4 channels Clock synchronous
	A/D converter		10-bit A/D converter: 1 circuit, 26 channels
	D/A converter		8 bits × 2 channels
	DMAC		2 channels
	CRC calculation circuit		CRC-CCITT
	CAN module		1 channel with 2.0B specification
	Watchdog timer		15 bits × 1 channel (with prescaler)
	Interrupts		Internal: 32 sources, External: 12 sources Software: 4 sources, Priority level: 7 levels
	Clock generation circuits		4 circuits • Main clock oscillation circuit (*) • Sub clock oscillation circuit (*) • On-chip oscillator • PLL frequency synthesizer (*) Equipped with a built-in feedback resistor
	Oscillation-stopped detector		Main clock oscillation stop and re-oscillation detection function
Electrical Characteristics	Supply Voltage		VCC = 3.0 to 5.5V (f(BCLK) = 24MHz, 1/1 prescaler, without software wait)
	Consumption current	Mask ROM	19mA (f(BCLK) = 24MHz, PLL operation, no division)
		Flash memory	21mA (f(BCLK) = 24MHz, PLL operation, no division)
		Mask ROM	3µA (f(BCLK) = 32kHz, Wait mode, Oscillation capacity Low)
		Flash memory	0.8µA (Stop mode, Topr = 25°C)
Flash Memory Version	Programming and erasure voltage		3.3 ± 0.3V or 5.0 ± 0.5V
	Programming and erasure endurance		100 times
I/O Characteristics	I/O withstand voltage		5.0V
	Output current		5mA
Operating Ambient Temperature			-40 to 85°C
Device Configuration			CMOS high performance silicon gate
Package			128-pin molded-plastic LQFP

NOTES:

1. I^{2}C-bus is a registered trademark of Koninklijke Philips Electronics N.V.
2. IEBus is a registered trademark of NEC Electronics Corporation.

1.3 Block Diagram

Figure 1.1 shows a Block Diagram.



NOTES:

1: ROM size depends on MCU type.
2: RAM size depends on MCU type.
3: Ports P11 to P14 are only in the 128-pin version.
4: 8 bits × 2 channels in the 100-pin version.

Figure 1.1 Block Diagram

1.4 Product Information

Table 1.3 lists the Product Information and Figure 1.2 shows the Type Number, Memory Size, and Packages.

Table 1.3 Product Information As of Apr. 2006

Type No.	ROM Capacity	RAM Capacity	Package Type [2]	Remarks
M306NLFHGP	384 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A	Flash memory version [1]
M306NNFHGP			PLQP0128KB-A	
M306NLFJGP (D)	512 K + 4 Kbytes	31 Kbytes	PLQP0100KB-A	
M306NNFJGP			PLQP0128KB-A	
M306NLME-XXXGP	192 Kbytes	16 Kbytes	PLQP0100KB-A	Mask ROM version
M306NNME-XXXGP			PLQP0128KB-A	
M306NLMG-XXXGP	256 Kbytes	20 Kbytes	PLQP0100KB-A	
M306NNMG-XXXGP			PLQP0128KB-A	

(D): Under development

NOTES:

1. Data flash memory provides an additional 4 Kbytes of ROM capacity (block A).
2. The correspondence between new and old package types is as follows.
   PLQP0100KB-A: 100P6Q-A
   PLQP0128KB-A: 128P6Q-A

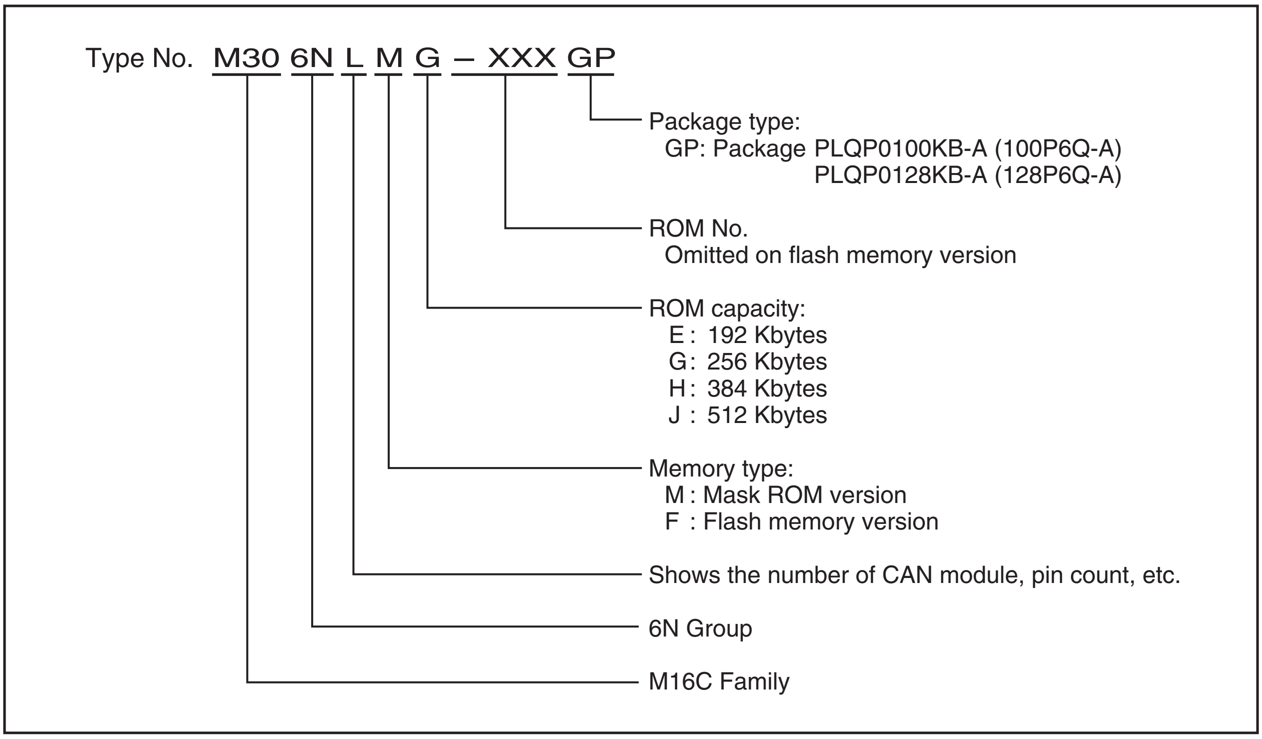
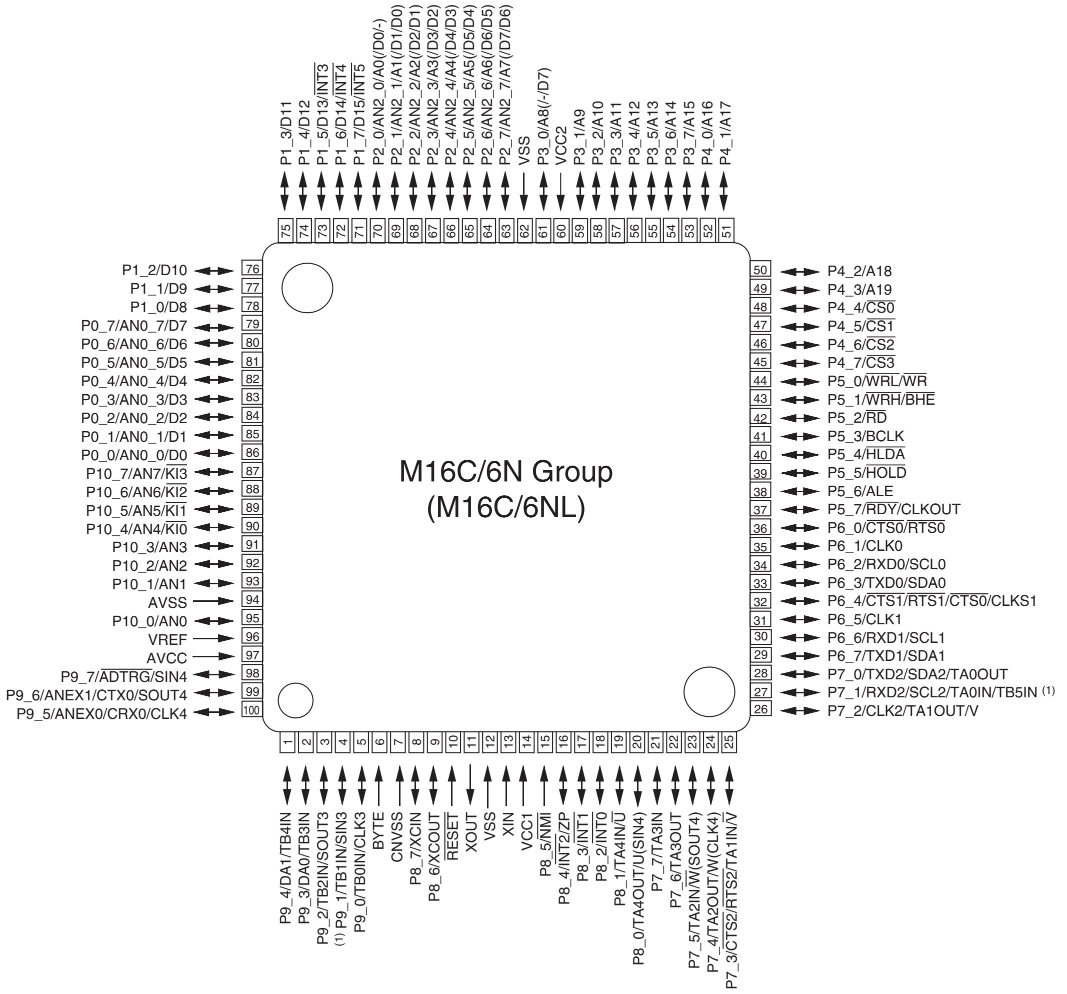


Figure 1.2 Type Number, Memory Size, and Package

1.5 Pin Assignments

Figures 1.3 and 1.4 show the Pin Assignment (Top View). Tables 1.4 and 1.5 list the List of Pin Names.



NOTE:

1. P7_1 and P9_1 are N channel open-drain pins.

Package: PLQP0100KB-A (100P6Q-A)

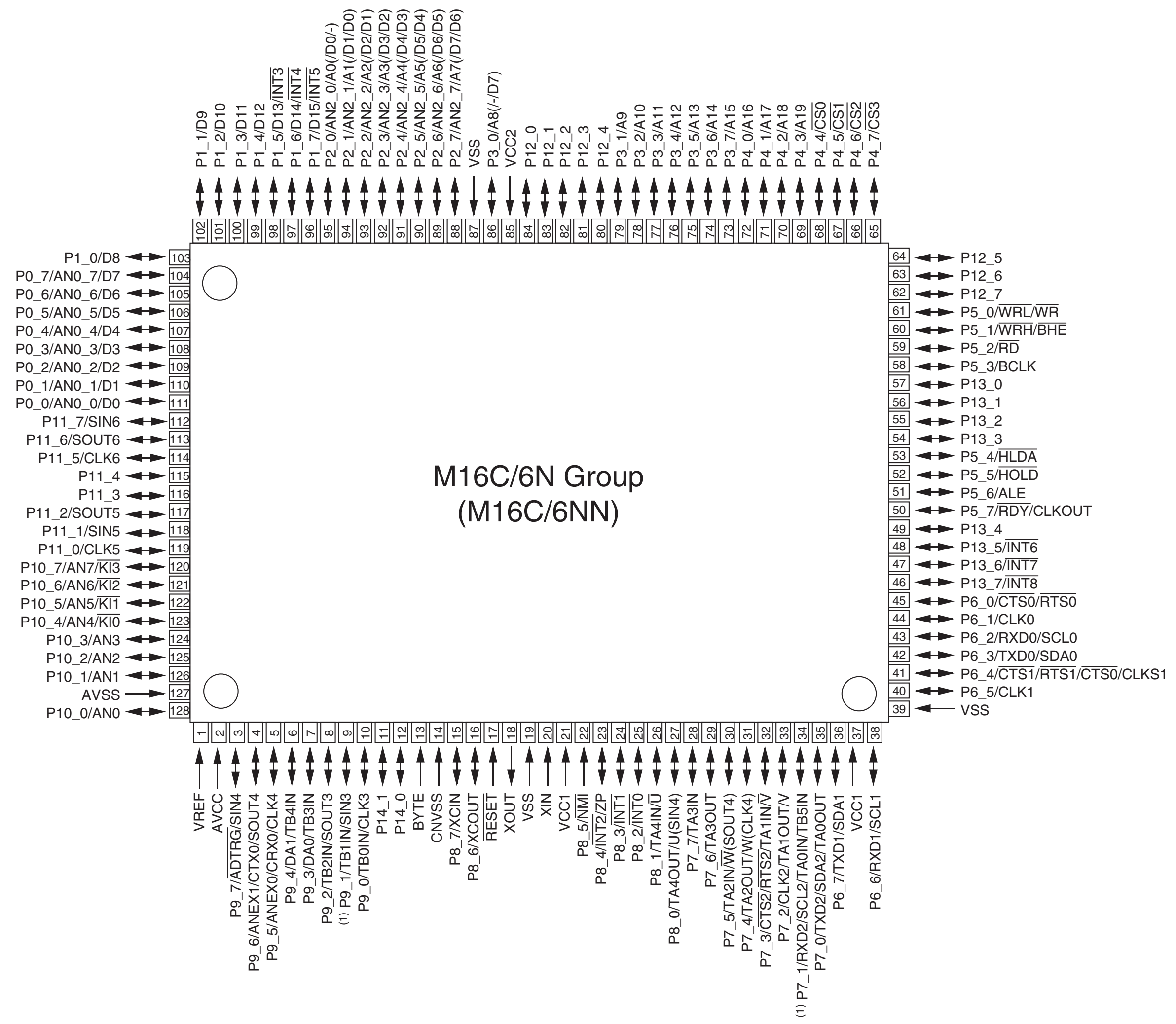
Figure 1.3 Pin Assignments (Top View) (1)

Table 1.4 List of Pin Names for 100-Pin Package (1)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	CAN Module Pin	Bus Control Pin
1		P9_4		TB4IN		DA1		
2		P9_3		TB3IN		DA0		
3		P9_2		TB2IN	SOUT3			
4		P9_1		TB1IN	SIN3			
5		P9_0		TB0IN	CLK3			
6	BYTE							
7	CNVSS							
8	XCIN	P8_7						
9	XCOUT	P8_6						
10	$\overline{\text{RESET}}$							
11	XOUT							
12	VSS							
13	XIN							
14	VCC1							
15		P8_5	$\overline{\text{NMI}}$					
16		P8_4	$\overline{\text{INT2}}$	ZP				
17		P8_3	$\overline{\text{INT1}}$					
18		P8_2	$\overline{\text{INT0}}$					
19		P8_1		TA4IN/$\overline{\text{U}}$				
20		P8_0		TA4OUT/U	(SIN4)			
21		P7_7		TA3IN				
22		P7_6		TA3OUT				
23		P7_5		TA2IN/$\overline{\text{W}}$	(SOUT4)			
24		P7_4		TA2OUT/W	(CLK4)			
25		P7_3		TA1IN/$\overline{\text{V}}$	$\overline{\text{CTS2}}$/$\overline{\text{RTS2}}$			
26		P7_2		TA1OUT/V	CLK2			
27		P7_1		TA0IN/TB5IN	RXD2/SCL2			
28		P7_0		TA0OUT	TXD2/SDA2			
29		P6_7			TXD1/SDA1			
30		P6_6			RXD1/SCL1			
31		P6_5			CLK1			
32		P6_4			$\overline{\text{CTS1}}$/$\overline{\text{RTS1}}$/$\overline{\text{CTS0}}$/CLKS1			
33		P6_3			TXD0/SDA0			
34		P6_2			RXD0/SCL0			
35		P6_1			CLK0			
36		P6_0			$\overline{\text{CTS0}}$/$\overline{\text{RTS0}}$			
37		P5_7						$\overline{\text{RDY}}$/CLKOUT
38		P5_6						ALE
39		P5_5						$\overline{\text{HOLD}}$
40		P5_4						$\overline{\text{HLDA}}$
41		P5_3						BCLK
42		P5_2						$\overline{\text{RD}}$
43		P5_1						$\overline{\text{WRH}}$/$\overline{\text{BHE}}$
44		P5_0						$\overline{\text{WRL}}$/$\overline{\text{WR}}$
45		P4_7						$\overline{\text{CS3}}$
46		P4_6						$\overline{\text{CS2}}$
47		P4_5						$\overline{\text{CS1}}$
48		P4_4						$\overline{\text{CS0}}$
49		P4_3						A19
50		P4_2						A18

Table 1.5 List of Pin Names for 100-Pin Package (2)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	CAN Module Pin	Bus Control Pin
51		P4_1						A17
52		P4_0						A16
53		P3_7						A15
54		P3_6						A14
55		P3_5						A13
56		P3_4						A12
57		P3_3						A11
58		P3_2						A10
59		P3_1						A9
60	VCC2							
61		P3_0						A8(/-/D7)
62	VSS							
63		P2_7				AN2_7		A7(/D7/D6)
64		P2_6				AN2_6		A6(/D6/D5)
65		P2_5				AN2_5		A5(/D5/D4)
66		P2_4				AN2_4		A4(/D4/D3)
67		P2_3				AN2_3		A3(/D3/D2)
68		P2_2				AN2_2		A2(/D2/D1)
69		P2_1				AN2_1		A1(/D1/D0)
70		P2_0				AN2_0		A0(/D0/-)
71		P1_7	$\overline{\mathrm{INT5}}$					D15
72		P1_6	$\overline{\mathrm{INT4}}$					D14
73		P1_5	$\overline{\mathrm{INT3}}$					D13
74		P1_4						D12
75		P1_3						D11
76		P1_2						D10
77		P1_1						D9
78		P1_0						D8
79		P0_7				AN0_7		D7
80		P0_6				AN0_6		D6
81		P0_5				AN0_5		D5
82		P0_4				AN0_4		D4
83		P0_3				AN0_3		D3
84		P0_2				AN0_2		D2
85		P0_1				AN0_1		D1
86		P0_0				AN0_0		D0
87		P10_7	$\overline{\mathrm{KI3}}$			AN7		
88		P10_6	$\overline{\mathrm{KI2}}$			AN6		
89		P10_5	$\overline{\mathrm{KI1}}$			AN5		
90		P10_4	$\overline{\mathrm{KI0}}$			AN4		
91		P10_3				AN3		
92		P10_2				AN2		
93		P10_1				AN1		
94	AVSS							
95		P10_0				AN0		
96	VREF							
97	AVCC							
98		P9_7			SIN4	$\overline{\mathrm{ADTRG}}$		
99		P9_6			SOUT4	ANEX1	CTX0	
100		P9_5			CLK4	ANEX0	CRX0	



NOTE:

1. P7_1 and P9_1 are N channel open-drain pins.

Package: PLQP0128KB-A (128P6Q-A)

Figure 1.4 Pin Assignments (Top View) (2)

Table 1.6 List of Pin Names for 128-Pin Package (1)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	CAN Module Pin	Bus Control Pin
1	VREF							
2	AVCC							
3		P9_7			SIN4	$\overline{\mathrm{ADTRG}}$		
4		P9_6			SOUT4	ANEX1	CTX0	
5		P9_5			CLK4	ANEX0	CRX0	
6		P9_4		TB4IN		DA1		
7		P9_3		TB3IN		DA0		
8		P9_2		TB2IN	SOUT3			
9		P9_1		TB1IN	SIN3			
10		P9_0		TB0IN	CLK3			
11		P14_1						
12		P14_0						
13	BYTE							
14	CNVSS							
15	XCIN	P8_7						
16	XCOUT	P8_6						
17	$\overline{\mathrm{RESET}}$							
18	XOUT							
19	VSS							
20	XIN							
21	VCC1							
22		P8_5	$\overline{\mathrm{NMI}}$					
23		P8_4	$\overline{\mathrm{INT2}}$	ZP				
24		P8_3	$\overline{\mathrm{INT1}}$					
25		P8_2	$\overline{\mathrm{INT0}}$					
26		P8_1		TA4IN/$\overline{\mathrm{U}}$				
27		P8_0		TA4OUT/U	(SIN4)			
28		P7_7		TA3IN				
29		P7_6		TA3OUT				
30		P7_5		TA2IN/$\overline{\mathrm{W}}$	(SOUT4)			
31		P7_4		TA2OUT/W	(CLK4)			
32		P7_3		TA1IN/$\overline{\mathrm{V}}$	$\overline{\mathrm{CTS2}}$/$\overline{\mathrm{RTS2}}$			
33		P7_2		TA1OUT/V	CLK2			
34		P7_1		TA0IN/TB5IN	RXD2/SCL2			
35		P7_0		TA0OUT	TXD2/SDA2			
36		P6_7			TXD1/SDA1			
37	VCC1							
38		P6_6			RXD1/SCL1			
39	VSS							
40		P6_5			CLK1			
41		P6_4			$\overline{\mathrm{CTS1}}$/$\overline{\mathrm{RTS1}}$/$\overline{\mathrm{CTS0}}$/CLKS1			
42		P6_3			TXD0/SDA0			
43		P6_2			RXD0/SCL0			
44		P6_1			CLK0			
45		P6_0			$\overline{\mathrm{CTS0}}$/$\overline{\mathrm{RTS0}}$			
46		P13_7	$\overline{\mathrm{INT8}}$					
47		P13_6	$\overline{\mathrm{INT7}}$					
48		P13_5	$\overline{\mathrm{INT6}}$					
49		P13_4						
50		P5_7						$\overline{\mathrm{RDY}}$/CLKOUT

Table 1.7 List of Pin Names for 128-Pin Package (2)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	CAN Module Pin	Bus Control Pin
51		P5_6						ALE
52		P5_5						$\overline{\text{HOLD}}$
53		P5_4						$\overline{\text{HLDA}}$
54		P13_3						
55		P13_2						
56		P13_1						
57		P13_0						
58		P5_3						BCLK
59		P5_2						$\overline{\text{RD}}$
60		P5_1						$\overline{\text{WRH}}/\overline{\text{BHE}}$
61		P5_0						$\overline{\text{WRL}}/\overline{\text{WR}}$
62		P12_7						
63		P12_6						
64		P12_5						
65		P4_7						$\overline{\text{CS3}}$
66		P4_6						$\overline{\text{CS2}}$
67		P4_5						$\overline{\text{CS1}}$
68		P4_4						$\overline{\text{CS0}}$
69		P4_3						A19
70		P4_2						A18
71		P4_1						A17
72		P4_0						A16
73		P3_7						A15
74		P3_6						A14
75		P3_5						A13
76		P3_4						A12
77		P3_3						A11
78		P3_2						A10
79		P3_1						A9
80		P12_4						
81		P12_3						
82		P12_2						
83		P12_1						
84		P12_0						
85	VCC2							
86		P3_0						A8(/-/D7)
87	VSS							
88		P2_7				AN2_7		A7(/D7/D6)
89		P2_6				AN2_6		A6(/D6/D5)
90		P2_5				AN2_5		A5(/D5/D4)
91		P2_4				AN2_4		A4(/D4/D3)
92		P2_3				AN2_3		A3(/D3/D2)
93		P2_2				AN2_2		A2(/D2/D1)
94		P2_1				AN2_1		A1(/D1/D0)
95		P2_0				AN2_0		A0(/D0/-)
96		P1_7	$\overline{\text{INT5}}$					D15
97		P1_6	$\overline{\text{INT4}}$					D14
98		P1_5	$\overline{\text{INT3}}$					D13
99		P1_4						D12
100		P1_3						D11

Table 1.8 List of Pin Names for 128-Pin Package (3)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	CAN Module Pin	Bus Control Pin
101		P1_2						D10
102		P1_1						D9
103		P1_0						D8
104		P0_7				AN0_7		D7
105		P0_6				AN0_6		D6
106		P0_5				AN0_5		D5
107		P0_4				AN0_4		D4
108		P0_3				AN0_3		D3
109		P0_2				AN0_2		D2
110		P0_1				AN0_1		D1
111		P0_0				AN0_0		D0
112		P11_7			SIN6			
113		P11_6			SOUT6			
114		P11_5			CLK6			
115		P11_4						
116		P11_3						
117		P11_2			SOUT5			
118		P11_1			SIN5			
119		P11_0			CLK5			
120		P10_7	$\overline{\text{KI3}}$			AN7		
121		P10_6	$\overline{\text{KI2}}$			AN6		
122		P10_5	$\overline{\text{KI1}}$			AN5		
123		P10_4	$\overline{\text{KI0}}$			AN4		
124		P10_3				AN3		
125		P10_2				AN2		
126		P10_1				AN1		
127	AVSS							
128		P10_0				AN0		

1.6 Pin Functions

Tables 1.9 to 1.11 list the Pin Functions.

Table 1.9 Pin Functions (100-pin and 128-pin Versions) (1)

Signal Name	Pin Name	I/O Type	Description
Power supply input	VCC1, VCC2, VSS	I	Apply 3.0 to 5.5 V to the VCC1 and VCC2 pins and 0 V to the VSS pin. The VCC apply condition is that VCC2 = VCC1 [1].
Analog power supply input	AVCC, AVSS	I	Applies the power supply for the A/D converter. Connect the AVCC pin to VCC1. Connect the AVSS pin to VSS.
Reset input	$\overline{\text{RESET}}$	I	The MCU is in a reset state when applying "L" to the this pin.
CNVSS	CNVSS	I	Switches processor mode. Connect this pin to VSS to when after a reset to start up in single-chip mode. Connect this pin to VCC1 to start up in microprocessor mode.
External data bus width select input	BYTE	I	Switches the data bus in external memory space. The data bus is 16-bit long when the this pin is held "L" and 8-bit long when the this pin is held "H". Set it to either one. Connect this pin to VSS when single-chip mode.
Bus control pins	D0 to D7	I/O	Inputs and outputs data (D0 to D7) when these pins are set as the separate bus.
	D8 to D15	I/O	Inputs and outputs data (D8 to D15) when external 16-bit data bus is set as the separate bus.
	A0 to A19	O	Output address bits (A0 to A19).
	A0/D0 to A7/D7	I/O	Input and output data (D0 to D7) and output address bits (A0 to A7) by time-sharing when external 8-bit data bus are set as the multiplexed bus.
	A1/D0 to A8/D7	I/O	Input and output data (D0 to D7) and output address bits (A1 to A8) by time-sharing when external 16-bit data bus are set as the multiplexed bus.
	$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	O	Output $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ signals. $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ are chip-select signals to specify an external space.
	$\overline{\text{WRL}}$/$\overline{\text{WR}}$ $\overline{\text{WRH}}$/$\overline{\text{BHE}}$ $\overline{\text{RD}}$	O	Output $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, ($\overline{\text{WR}}$, $\overline{\text{BHE}}$), $\overline{\text{RD}}$ signals. $\overline{\text{WRL}}$ and $\overline{\text{WRH}}$ or $\overline{\text{BHE}}$, and $\overline{\text{WR}}$ can be switched by program. • $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, and $\overline{\text{RD}}$ are selected The $\overline{\text{WRL}}$ signal becomes "L" by writing data to an even address in an external memory space. The $\overline{\text{WRH}}$ signal becomes "L" by writing data to an odd address in an external memory space. The $\overline{\text{RD}}$ pin signal becomes "L" by reading data in an external memory space. • $\overline{\text{WR}}$, $\overline{\text{BHE}}$, and $\overline{\text{RD}}$ are selected The $\overline{\text{WR}}$ signal becomes "L" by writing data in an external memory space. The $\overline{\text{RD}}$ signal becomes "L" by reading data in an external memory space. The $\overline{\text{BHE}}$ signal becomes "L" by accessing an odd address. Select $\overline{\text{WR}}$, $\overline{\text{BHE}}$, and $\overline{\text{RD}}$ for an external 8-bit data bus.
	ALE	O	ALE is a signal to latch the address.
	$\overline{\text{HOLD}}$	I	While the $\overline{\text{HOLD}}$ pin is held "L", the MCU is placed in a hold state.
	$\overline{\text{HLDA}}$	O	In a hold state, $\overline{\text{HLDA}}$ outputs a "L" signal.
	$\overline{\text{RDY}}$	I	While applying a "L" signal to the $\overline{\text{RDY}}$ pin, the MCU is placed in a wait state.

I: Input O: Output I/O: Input/Output

NOTE:

1. In this manual, hereafter, VCC refers to VCC1 unless otherwise noted.

Table 1.10 Pin Functions (100-pin and 128-pin Versions) (2)

Signal Name	Pin Name	I/O Type	Description
Main clock input	XIN	I	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT (1). To use the external clock, input the clock from XIN and leave XOUT open.
Main clock output	XOUT	O	
Sub clock input	XCIN	I	I/O pins for a sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUT (1). To use the external clock, input the clock from XCIN and leave XCOUT open.
Sub clock output	XCOUT	O	
BCLK output	BCLK	O	Outputs the BCLK signal.
Clock output	CLKOUT	O	The clock of the same cycle as fC, f8, or f32 is output.
INT interrupt input	NT0 to INT8 (2)	I	Input pins for the $\overline{\text{INT}}$ interrupt.
$\overline{\text{NMI}}$ interrupt input	$\overline{\text{NMI}}$	I	Input pin for the $\overline{\text{NMI}}$ interrupt.
Key input interrupt input	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	Input pins for the key input interrupt.
Timer A	TA0OUT to TA4OUT	I/O	These are timer A0 to timer A4 I/O pins.
	TA0IN to TA4IN	I	These are timer A0 to timer A4 input pins.
	ZP	I	Input pin for the Z-phase.
Timer B	TB0IN to TB5IN	I	These are timer B0 to timer B5 input pins.
Three-phase motor control output	U, $\overline{\text{U}}$, V, $\overline{\text{V}}$, W, $\overline{\text{W}}$	O	These are Three-phase motor control output pins.
Serial interface	$\overline{\text{CTS0}}$ to $\overline{\text{CTS2}}$	I	These are transmit control input pins.
	$\overline{\text{RTS0}}$ to $\overline{\text{RTS2}}$	O	These are receive control output pins.
	CLK0 to CLK6 (2)	I/O	These are transfer clock I/O pins.
	RXD0 to RXD2	I	These are serial data input pins.
	SIN3 to SIN6 (2)	I	These are serial data input pins.
	TXD0 to TXD2	O	These are serial data output pins.
	SOUT3 to SOUT6 (2)	O	These are serial data output pins.
	CLKS1	O	This is output pin for transfer clock output from multiple pins function.
I^2C mode	SDA0 to SDA2	I/O	These are serial data I/O pins.
	SCL0 to SCL2	I/O	These are transfer clock I/O pins. (however, SCL2 for the N-channel open drain output.)
Reference voltage input	VREF	I	Applies the reference voltage for the A/D converter and D/A converter.
A/D converter	AN0 to AN7 AN0_0 to AN0_7 AN2_0 to AN2_7	I	Analog input pins for the A/D converter.
	$\overline{\text{ADTRG}}$	I	This is an A/D trigger input pin.
	ANEX0	I/O	This is the extended analog input pin for the A/D converter, and is the output in external op-amp connection mode.
	ANEX1	I	This is the extended analog input pin for the A/D converter.
D/A converter	DA0, DA1	O	These are the output pins for the D/A converter.
CAN module	CRX0	I	This is the input pin for the CAN module.
	CTX0	O	This is the output pin for the CAN module.

I: Input O: Output I/O: Input/Output

NOTES:

1. Ask the oscillator maker the oscillation characteristic.
2. INT6 to INT8, CLK5, CLK6, SIN5, SIN6, SOUT5, SOUT6 are only in the 128-pin version.

Table 1.11 Pin Functions (100-pin and 128-pin Versions) (3)

Signal Name	Pin Name	I/O Type	Description
I/O port	P0_0 to P0_7 P1_0 to P1_7 P2_0 to P2_7 P3_0 to P3_7 P4_0 to P4_7 P5_0 to P5_7 P6_0 to P6_7 P7_0 to P7_7 P8_0 to P8_4 P8_6, P8_7 P9_0 to P9_7 P10_0 to P10_7 P11_0 to P11_7 (1) P12_0 to P12_7 (1) P13_0 to P13_7 (1) P14_0, P14_1 (1)	I/O	8-bit I/O ports in CMOS, having a direction register to select an input or output. Each pin is set as an input port or output port. An input port can be set for a pull-up or for no pull-up in 4-bit unit by program. (however P7_1 and P9_1 for the N-channel open drain output.)
Input port	P8_5	I	Input pin for the $\overline{\text{NMI}}$ interrupt. Pin states can be read by the P8_5 bit in the P8 register.

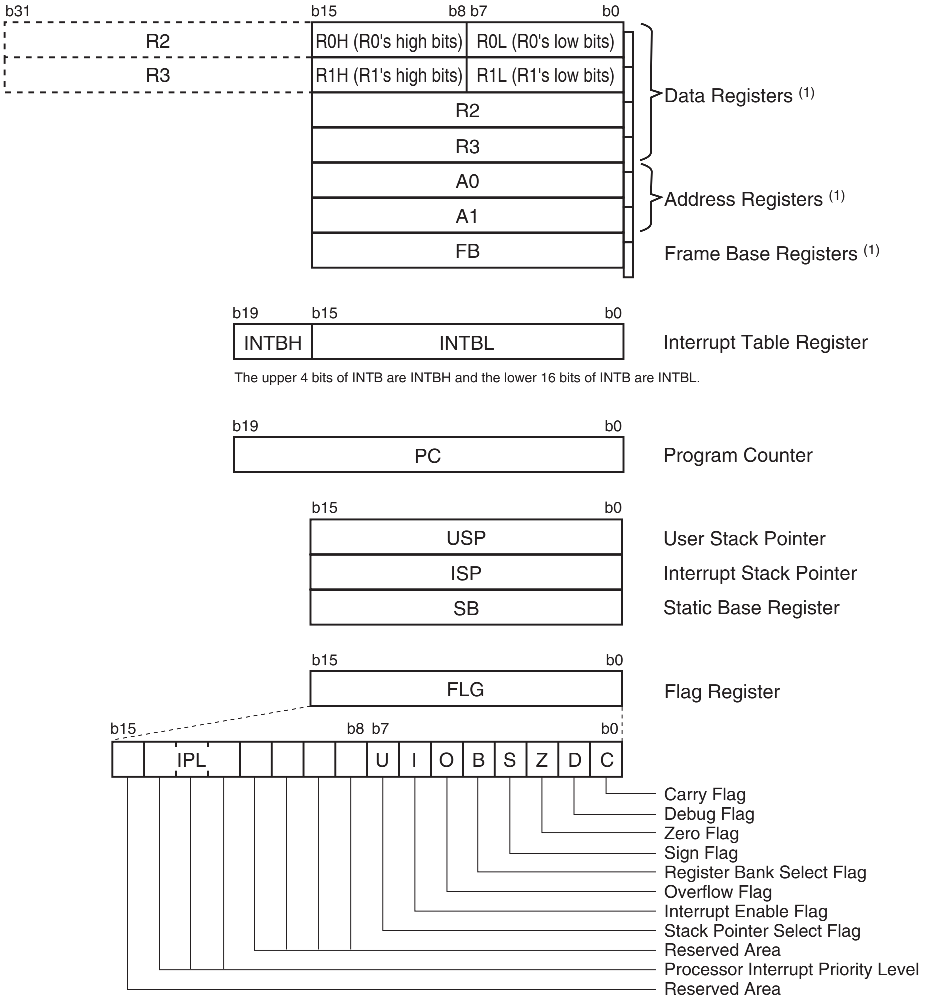
I: Input O: Output I/O: Input/Output

NOTE:

1. Ports P11 to P14 are only in the 128-pin version.

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU Registers. The CPU has 13 registers. Of these, R0, R1, R2, R3, A0, A1, and FB configure a register bank. There are two register banks.



NOTE:
1. These registers comprise a register bank. There are two register banks.

Figure 2.1 CPU Registers

2.1 Data Registers (R0, R1, R2, and R3)

The R0 register consists of 16 bits, and is used mainly for transfers and arithmetic/logic operations. R1 to R3 are the same as R0.

The R0 register can be separated between high (R0H) and low (R0L) for use as two 8-bit data registers. R1H and R1L are the same as R0H and R0L. Conversely R2 and R0 can be combined for use as a 32-bit data register (R2R0). R3R1 is analogous to R2R0.

2.2 Address Registers (A0 and A1)

The A0 register consists of 16 bits, and is used for address register indirect addressing and address register relative addressing. They also are used for transfers and arithmetic/logic operations. A1 is the same as A0.

In some instructions, A1 and A0 can be combined for use as a 32-bit address register (A1A0).

2.3 Frame Base Register (FB)

FB is configured with 16 bits, and is used for FB relative addressing.

2.4 Interrupt Table Register (INTB)

INTB is configured with 20 bits, indicating the start address of an interrupt vector table.

2.5 Program Counter (PC)

PC is configured with 20 bits, indicating the address of an instruction to be executed.

2.6 User Stack Pointer (USP), Interrupt Stack Pointer (ISP)

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.
Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

2.7 Static Base Register (SB)

SB is configured with 16 bits, and is used for SB relative addressing.

2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

2.8.1 Carry Flag (C Flag)

This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.

2.8.2 Debug Flag (D Flag)

This flag is used exclusively for debugging purpose. During normal use, set to 0.

2.8.3 Zero Flag (Z Flag)

This flag is set to 1 when an arithmetic operation resulted in 0; otherwise, it is 0.

2.8.4 Sign Flag (S Flag)

This flag is set to 1 when an arithmetic operation resulted in a negative value; otherwise, it is 0.

2.8.5 Register Bank Select Flag (B Flag)

Register bank 0 is selected when this flag is 0; register bank 1 is selected when this flag is 1.

2.8.6 Overflow Flag (O Flag)

This flag is set to 1 when the operation resulted in an overflow; otherwise, it is 0.

2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.
Maskable interrupts are disabled when the I flag is 0, and are enabled when the I flag is 1. The I flag is set to 0 when the interrupt request is accepted.

2.8.8 Stack Pointer Select Flag (U Flag)

ISP is selected when the U flag is 0; USP is selected when the U flag is 1.
The U flag is set to 0 when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.
If a requested interrupt has priority greater than IPL, the interrupt request is enabled.

2.8.10 Reserved Area

When white to this bit, write 0. When read, its content is undefined.

3. Memory

Figure 3.1 shows a Memory Map. The address space extends the 1 Mbyte from address 00000h to FFFFFh. The internal ROM is allocated in a lower address direction beginning with address FFFFFh. For example, a 512-Kbyte internal ROM is allocated to the addresses from 80000h to FFFFFh.

As for the flash memory version, 4-Kbyte space (block A) exists in 0F000h to 0FFFFh. 4-Kbyte space is mainly for storing data. In addition to storing data, 4-Kbyte space also can store programs.

The fixed interrupt vector table is allocated to the addresses from FFFDCh to FFFFFh. Therefore, store the start address of each interrupt routine here.

The internal RAM is allocated in an upper address direction beginning with address 00400h. For example, a 31-Kbyte internal RAM is allocated to the addresses from 00400h to 07FFFh. In addition to storing data, the internal RAM also stores the stack used when calling subroutines and when interrupts are generated.

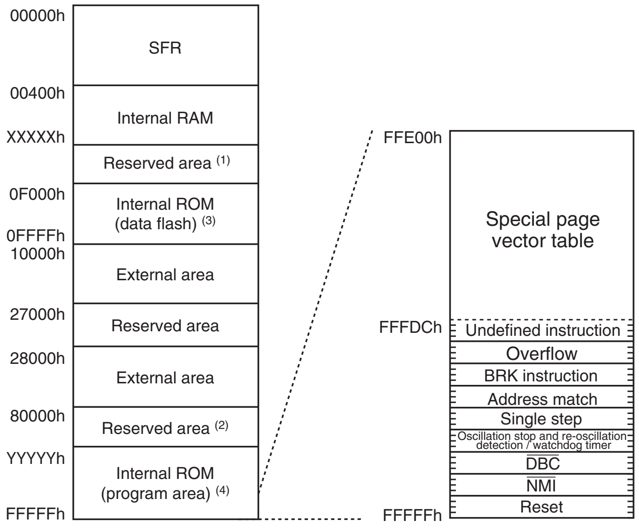
The Special Function Registers (SFRs) are allocated to the addresses from 00000h to 003FFh. Peripheral function control registers are located here. Of the SFR, any area which has no functions allocated is reserved for future use and cannot be accessed by user.

The special page vector table is allocated to the addresses from FFE00h to FFFDBh. This vector is used by the JMPS or JSRS instruction. For details, refer to **M16C/60, M16C/20, M16C/Tiny Series Software Manual**.

In memory expansion and microprocessor modes, some areas are reserved for future use and cannot be used by users.

Internal RAM	
Capacity	Address XXXXXh
16 Kbytes	043FFh
20 Kbytes	053FFh
31 Kbytes	07FFFh

Internal ROM (4)	
Capacity	Address YYYYYh
192 Kbytes	D0000h
256 Kbytes	C0000h
384 Kbytes	A0000h
512 Kbytes	80000h



NOTES:

1. During memory expansion mode or microprocessor mode, cannot be used.
2. In memory expansion mode, cannot be used.
3. As for the flash memory version, 4-Kbyte space (block A) exists.
4. When using the masked ROM version, write nothing to internal ROM area.
5. Shown here is a memory map for the case where the PM10 bit in the PM1 register is 1 (block A enabled, addresses 10000h to 26FFFh for $\overline{CS2}$ area) and the PM13 bit in the PM1 register is 1 (internal RAM area is expanded over 192 Kbytes).

Figure 3.1 Memory Map

4. Special Function Registers (SFRs)

An SFR (Special Function Register) is a control register for a peripheral function.
Tables 4.1 to 4.12 list the SFR Information.

Table 4.1 SFR Information (1) [3]

Address	Register	Symbol	After Reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0 [1]	PM0	00000000b (CNVSS pin is "L") 00000011b (CNVSS pin is "H")
0005h	Processor Mode Register 1	PM1	00001000b
0006h	System Clock Control Register 0	CM0	01001000b
0007h	System Clock Control Register 1	CM1	00100000b
0008h	Chip Select Control Register	CSR	00000001b
0009h	Address Match Interrupt Enable Register	AIER	XXXXXX00b
000Ah	Protect Register	PRCR	XX000000b
000Bh			
000Ch	Oscillation Stop Detection Register [2]	CM2	0X000000b
000Dh			
000Eh	Watchdog Timer Start Register	WDTS	XXh
000Fh	Watchdog Timer Control Register	WDC	00XXXXXXb
0010h	Address Match Interrupt Register 0	RMAD0	00h
0011h			00h
0012h			X0h
0013h			
0014h	Address Match Interrupt Register 1	RMAD1	00h
0015h			00h
0016h			X0h
0017h			
0018h			
0019h			
001Ah			
001Bh	Chip Select Expansion Control Register	CSE	00h
001Ch	PLL Control Register 0	PLC0	0001X010b
001Dh			
001Eh	Processor Mode Register 2	PM2	XXX00000b
001Fh			
0020h	DMA0 Source Pointer	SAR0	XXh
0021h			XXh
0022h			XXh
0023h			
0024h	DMA0 Destination Pointer	DAR0	XXh
0025h			XXh
0026h			XXh
0027h			
0028h	DMA0 Transfer Counter	TCR0	XXh
0029h			XXh
002Ah			
002Bh			
002Ch	DMA0 Control Register	DM0CON	00000X00b
002Dh			
002Eh			
002Fh			
0030h	DMA1 Source Pointer	SAR1	XXh
0031h			XXh
0032h			XXh
0033h			
0034h	DMA1 Destination Pointer	DAR1	XXh
0035h			XXh
0036h			XXh
0037h			
0038h	DMA1 Transfer Counter	TCR1	XXh
0039h			XXh
003Ah			
003Bh			
003Ch	DMA1 Control Register	DM1CON	00000X00b
003Dh			
003Eh			
003Fh			

X: Undefined

NOTES:
1. Bits PM00 and PM01 in the PM0 register do not change at software reset, watchdog timer reset and oscillation stop detection reset.
2. Bits CM20, CM21, and CM27 in the CM2 register do not change at oscillation stop detection reset.
3. Blank spaces are reserved. No access is allowed.

Table 4.2 SFR Information (2) [2]

Address	Register	Symbol	After Reset
0040h			
0041h	CAN0 Wake-up Interrupt Control Register	C01WKIC	XXXXX000b
0042h	CAN0 Successful Reception Interrupt Control Register	C0RECIC	XXXXX000b
0043h	CAN0 Successful Transmission Interrupt Control Register	C0TRMIC	XXXXX000b
0044h	INT3 Interrupt Control Register	INT3IC	XX00X000b
0045h	Timer B5 Interrupt Control Register	TB5IC	XXXXX000b
	SI/O5 Interrupt Control Register [1]	S5IC	
0046h	Timer B4 Interrupt Control Register	TB4IC	XXXXX000b
	UART1 Bus Collision Detection Interrupt Control Register	U1BCNIC	
0047h	Timer B3 Interrupt Control Register	TB3IC	XXXXX000b
	UART0 Bus Collision Detection Interrupt Control Register	U0BCNIC	
0048h	SI/O4 Interrupt Control Register	S4IC	XX00X000b
	INT5 Interrupt Control Register	INT5IC	
0049h	SI/O3 Interrupt Control Register	S3IC	XX00X000b
	INT4 Interrupt Control Register	INT4IC	
004Ah	UART2 Bus Collision Detection Interrupt Control Register	U2BCNIC	XXXXX000b
004Bh	DMA0 Interrupt Control Register	DM0IC	XXXXX000b
004Ch	DMA1 Interrupt Control Register	DM1IC	XXXXX000b
004Dh	CAN0 Error Interrupt Control Register	C01ERRIC	XXXXX000b
004Eh	A/D Conversion Interrupt Control Register	ADIC	XXXXX000b
	Key Input Interrupt Control Register	KUPIC	
004Fh	UART2 Transmit Interrupt Control Register	S2TIC	XXXXX000b
0050h	UART2 Receive Interrupt Control Register	S2RIC	XXXXX000b
0051h	UART0 Transmit Interrupt Control Register	S0TIC	XXXXX000b
0052h	UART0 Receive Interrupt Control Register	S0RIC	XXXXX000b
0053h	UART1 Transmit Interrupt Control Register	S1TIC	XXXXX000b
0054h	UART1 Receive Interrupt Control Register	S1RIC	XXXXX000b
0055h	Timer A0 Interrupt Control Register	TA0IC	XXXXX000b
0056h	Timer A1 Interrupt Control Register	TA1IC	XXXXX000b
0057h	Timer A2 Interrupt Control Register	TA2IC	XX00X000b
	INT7 Interrupt Control Register [1]	INT7IC	
0058h	Timer A3 Interrupt Control Register	TA3IC	XX00X000b
	INT6 Interrupt Control Register [1]	INT6IC	
0059h	Timer A4 Interrupt Control Register	TA4IC	XXXXX000b
005Ah	Timer B0 Interrupt Control Register	TB0IC	XXXXX000b
	SI/O6 Interrupt Control Register [1]	S6IC	
005Bh	Timer B1 Interrupt Control Register	TB1IC	XX00X000b
	INT8 Interrupt Control Register [1]	INT8IC	
005Ch	Timer B2 Interrupt Control Register	TB2IC	XXXXX000b
005Dh	INT0 Interrupt Control Register	INT0IC	XX00X000b
005Eh	INT1 Interrupt Control Register	INT1IC	XX00X000b
005Fh	INT2 Interrupt Control Register	INT2IC	XX00X000b
0060h	CAN0 Message Box 0: Identifier / DLC		XXh
0061h			XXh
0062h			XXh
0063h			XXh
0064h			XXh
0065h			XXh
0066h	CAN0 Message Box 0: Data Field		XXh
0067h			XXh
0068h			XXh
0069h			XXh
006Ah			XXh
006Bh			XXh
006Ch			XXh
006Dh			XXh
006Eh	CAN0 Message Box 0: Time Stamp		XXh
006Fh			XXh
0070h	CAN0 Message Box 1: Identifier / DLC		XXh
0071h			XXh
0072h			XXh
0073h			XXh
0074h			XXh
0075h			XXh
0076h	CAN0 Message Box 1: Data Field		XXh
0077h			XXh
0078h			XXh
0079h			XXh
007Ah			XXh
007Bh			XXh
007Ch			XXh
007Dh			XXh
007Eh	CAN0 Message Box 1: Time Stamp		XXh
007Fh			XXh

X: Undefined

NOTES:

1. These registers exist only in the 128-pin version.
2. Blank spaces are reserved. No access is allowed.

Table 4.3 SFR Information (3)

Address	Register	Symbol	After Reset
0080h	CAN0 Message Box 2: Identifier / DLC		XXh
0081h			XXh
0082h			XXh
0083h			XXh
0084h			XXh
0085h			XXh
0086h	CAN0 Message Box 2: Data Field		XXh
0087h			XXh
0088h			XXh
0089h			XXh
008Ah			XXh
008Bh			XXh
008Ch			XXh
008Dh			XXh
008Eh	CAN0 Message Box 2: Time Stamp		XXh
008Fh			XXh
0090h	CAN0 Message Box 3: Identifier / DLC		XXh
0091h			XXh
0092h			XXh
0093h			XXh
0094h			XXh
0095h			XXh
0096h	CAN0 Message Box 3: Data Field		XXh
0097h			XXh
0098h			XXh
0099h			XXh
009Ah			XXh
009Bh			XXh
009Ch			XXh
009Dh			XXh
009Eh	CAN0 Message Box 3: Time Stamp		XXh
009Fh			XXh
00A0h	CAN0 Message Box 4: Identifier / DLC		XXh
00A1h			XXh
00A2h			XXh
00A3h			XXh
00A4h			XXh
00A5h			XXh
00A6h	CAN0 Message Box 4: Data Field		XXh
00A7h			XXh
00A8h			XXh
00A9h			XXh
00AAh			XXh
00ABh			XXh
00ACh			XXh
00ADh			XXh
00AEh	CAN0 Message Box 4: Time Stamp		XXh
00AFh			XXh
00B0h	CAN0 Message Box 5: Identifier / DLC		XXh
00B1h			XXh
00B2h			XXh
00B3h			XXh
00B4h			XXh
00B5h			XXh
00B6h	CAN0 Message Box 5: Data Field		XXh
00B7h			XXh
00B8h			XXh
00B9h			XXh
00BAh			XXh
00BBh			XXh
00BCh			XXh
00BDh			XXh
00BEh	CAN0 Message Box 5: Time Stamp		XXh
00BFh			XXh

X: Undefined

Table 4.4 SFR Information (4)

Address	Register	Symbol	After Reset
00C0h	CAN0 Message Box 6: Identifier / DLC		XXh
00C1h			XXh
00C2h			XXh
00C3h			XXh
00C4h			XXh
00C5h			XXh
00C6h	CAN0 Message Box 6: Data Field		XXh
00C7h			XXh
00C8h			XXh
00C9h			XXh
00CAh			XXh
00CBh			XXh
00CCh			XXh
00CDh			XXh
00CEh	CAN0 Message Box 6: Time Stamp		XXh
00CFh			XXh
00D0h	CAN0 Message Box 7: Identifier / DLC		XXh
00D1h			XXh
00D2h			XXh
00D3h			XXh
00D4h			XXh
00D5h			XXh
00D6h	CAN0 Message Box 7: Data Field		XXh
00D7h			XXh
00D8h			XXh
00D9h			XXh
00DAh			XXh
00DBh			XXh
00DCh			XXh
00DDh			XXh
00DEh	CAN0 Message Box 7: Time Stamp		XXh
00DFh			XXh
00E0h	CAN0 Message Box 8: Identifier / DLC		XXh
00E1h			XXh
00E2h			XXh
00E3h			XXh
00E4h			XXh
00E5h			XXh
00E6h	CAN0 Message Box 8: Data Field		XXh
00E7h			XXh
00E8h			XXh
00E9h			XXh
00EAh			XXh
00EBh			XXh
00ECh			XXh
00EDh			XXh
00EEh	CAN0 Message Box 8: Time Stamp		XXh
00EFh			XXh
00F0h	CAN0 Message Box 9: Identifier / DLC		XXh
00F1h			XXh
00F2h			XXh
00F3h			XXh
00F4h			XXh
00F5h			XXh
00F6h	CAN0 Message Box 9: Data Field		XXh
00F7h			XXh
00F8h			XXh
00F9h			XXh
00FAh			XXh
00FBh			XXh
00FCh			XXh
00FDh			XXh
00FEh	CAN0 Message Box 9: Time Stamp		XXh
00FFh			XXh

X: Undefined

Table 4.5 SFR Information (5)

Address	Register	Symbol	After Reset
0100h	CAN0 Message Box 10: Identifier / DLC		XXh
0101h			XXh
0102h			XXh
0103h			XXh
0104h			XXh
0105h			XXh
0106h	CAN0 Message Box 10: Data Field		XXh
0107h			XXh
0108h			XXh
0109h			XXh
010Ah			XXh
010Bh			XXh
010Ch			XXh
010Dh			XXh
010Eh	CAN0 Message Box 10: Time Stamp		XXh
010Fh			XXh
0110h	CAN0 Message Box 11: Identifier / DLC		XXh
0111h			XXh
0112h			XXh
0113h			XXh
0114h			XXh
0115h			XXh
0116h	CAN0 Message Box 11: Data Field		XXh
0117h			XXh
0118h			XXh
0119h			XXh
011Ah			XXh
011Bh			XXh
011Ch			XXh
011Dh			XXh
011Eh	CAN0 Message Box 11: Time Stamp		XXh
011Fh			XXh
0120h	CAN0 Message Box 12: Identifier / DLC		XXh
0121h			XXh
0122h			XXh
0123h			XXh
0124h			XXh
0125h			XXh
0126h	CAN0 Message Box 12: Data Field		XXh
0127h			XXh
0128h			XXh
0129h			XXh
012Ah			XXh
012Bh			XXh
012Ch			XXh
012Dh			XXh
012Eh	CAN0 Message Box 12: Time Stamp		XXh
012Fh			XXh
0130h	CAN0 Message Box 13: Identifier / DLC		XXh
0131h			XXh
0132h			XXh
0133h			XXh
0134h			XXh
0135h			XXh
0136h	CAN0 Message Box 13: Data Field		XXh
0137h			XXh
0138h			XXh
0139h			XXh
013Ah			XXh
013Bh			XXh
013Ch			XXh
013Dh			XXh
013Eh	CAN0 Message Box 13: Time Stamp		XXh
013Fh			XXh

X: Undefined

Table 4.6 SFR Information (6) [1]

Address	Register	Symbol	After Reset
0140h	CAN0 Message Box 14: Identifier /DLC		XXh
0141h			XXh
0142h			XXh
0143h			XXh
0144h			XXh
0145h			XXh
0146h	CAN0 Message Box 14: Data Field		XXh
0147h			XXh
0148h			XXh
0149h			XXh
014Ah			XXh
014Bh			XXh
014Ch			XXh
014Dh			XXh
014Eh	CAN0 Message Box 14: Time Stamp		XXh
014Fh			XXh
0150h	CAN0 Message Box 15: Identifier /DLC		XXh
0151h			XXh
0152h			XXh
0153h			XXh
0154h			XXh
0155h			XXh
0156h	CAN0 Message Box 15: Data Field		XXh
0157h			XXh
0158h			XXh
0159h			XXh
015Ah			XXh
015Bh			XXh
015Ch			XXh
015Dh			XXh
015Eh	CAN0 Message Box 15: Time Stamp		XXh
015Fh			XXh
0160h	CAN0 Global Mask Register	C0GMR	XXh
0161h			XXh
0162h			XXh
0163h			XXh
0164h			XXh
0165h			XXh
0166h	CAN0 Local Mask A Register	C0LMAR	XXh
0167h			XXh
0168h			XXh
0169h			XXh
016Ah			XXh
016Bh			XXh
016Ch	CAN0 Local Mask B Register	C0LMBR	XXh
016Dh			XXh
016Eh			XXh
016Fh			XXh
0170h			XXh
0171h			XXh
0172h			
0173h			
0174h			
0175h			
0176h			
0177h			
0178h			
0179h			
017Ah			
017Bh			
017Ch			
017Dh			
017Eh			
017Fh			

X: Undefined

NOTE:
1. Blank spaces are reserved. No access is allowed.

Table 4.7 SFR Information (7) (2)

Address	Register	Symbol	After Reset
0180h			
0181h			
0182h			
0183h			
0184h			
0185h			
0186h			
0187h			
0188h			
0189h			
018Ah			
018Bh			
018Ch			
018Dh			
018Eh			
018Fh			
0190h			
0191h			
0192h			
0193h			
0194h			
0195h			
0196h			
0197h			
0198h			
0199h			
019Ah			
019Bh			
019Ch			
019Dh			
019Eh			
019Fh			
01A0h			
01A1h			
01A2h			
01A3h			
01A4h			
01A5h			
01A6h			
01A7h			
01A8h			
01A9h			
01AAh			
01ABh			
01ACh			
01ADh			
01AEh			
01AFh			
01B0h			
01B1h			
01B2h			
01B3h			
01B4h			
01B5h	Flash Memory Control Register 1 (1)	FMR1	0X00XX0Xb
01B6h			
01B7h	Flash Memory Control Register 0 (1)	FMR0	00000001b
01B8h	Address Match Interrupt Register 2	RMAD2	00h
01B9h			00h
01BAh			X0h
01BBh	Address Match Interrupt Enable Register 2	AIER2	XXXXXX00b
01BCh	Address Match Interrupt Register 3	RMAD3	00h
01BDh			00h
01BEh			X0h
01BFh			

X: Undefined

NOTES:

1. These registers are included in the flash memory version. Cannot be accessed by users in the mask ROM version.
2. Blank spaces are reserved. No access is allowed.

Table 4.8 SFR Information (8) (3)

Address	Register	Symbol	After Reset
01C0h	Timer B3, B4, B5 Count Start Flag	TBSR	000XXXXXb
01C1h			
01C2h	Timer A1-1 Register	TA11	XXh
01C3h			XXh
01C4h	Timer A2-1 Register	TA21	XXh
01C5h			XXh
01C6h	Timer A4-1 Register	TA41	XXh
01C7h			XXh
01C8h	Three-Phase PWM Control Register 0	INVC0	00h
01C9h	Three-Phase PWM Control Register 1	INVC1	00h
01CAh	Three-Phase Output Buffer Register 0	IDB0	00111111b
01CBh	Three-Phase Output Buffer Register 1	IDB1	00111111b
01CCh	Dead Time Timer	DTT	XXh
01CDh	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	XXh
01CEh			
01CFh	Interrupt Source Select Register 2	IFSR2	X0000000b
01D0h	Timer B3 Register	TB3	XXh
01D1h			XXh
01D2h	Timer B4 Register	TB4	XXh
01D3h			XXh
01D4h	Timer B5 Register	TB5	XXh
01D5h			XXh
01D6h	SI/O6 Transmit/Receive Register (1)	S6TRR	XXh
01D7h			
01D8h	SI/O6 Control Register (1)	S6C	01000000b
01D9h	SI/O6 Bit Rate Register (1)	S6BRG	XXh
01DAh	SI/O3, 4, 5, 6 Transmit/Receive Register (2)	S3456TRR	XXXX0000b
01DBh	Timer B3 Mode Register	TB3MR	00XX0000b
01DCh	Timer B4 Mode Register	TB4MR	00XX0000b
01DDh	Timer B5 Mode Register	TB5MR	00XX0000b
01DEh	Interrupt Source Select Register 0	IFSR0	00h
01DFh	Interrupt Source Select Register 1	IFSR1	00h
01E0h	SI/O3 Transmit/Receive Register	S3TRR	XXh
01E1h			
01E2h	SI/O3 Control Register	S3C	01000000b
01E3h	SI/O3 Bit Rate Register	S3BRG	XXh
01E4h	SI/O4 Transmit/Receive Register	S4TRR	XXh
01E5h			
01E6h	SI/O4 Control Register	S4C	01000000b
01E7h	SI/O4 Bit Rate Register	S4BRG	XXh
01E8h	SI/O5 Transmit/Receive Register (1)	S5TRR	XXh
01E9h			
01EAh	SI/O5 Control Register (1)	S5C	01000000b
01EBh	SI/O5 Bit Rate Register (1)	S5BRG	XXh
01ECh	UART0 Special Mode Register 4	U0SMR4	00h
01EDh	UART0 Special Mode Register 3	U0SMR3	000X0X0Xb
01EEh	UART0 Special Mode Register 2	U0SMR2	X0000000b
01EFh	UART0 Special Mode Register	U0SMR	X0000000b
01F0h	UART1 Special Mode Register 4	U1SMR4	00h
01F1h	UART1 Special Mode Register 3	U1SMR3	000X0X0Xb
01F2h	UART1 Special Mode Register 2	U1SMR2	X0000000b
01F3h	UART1 Special Mode Register	U1SMR	X0000000b
01F4h	UART2 Special Mode Register 4	U2SMR4	00h
01F5h	UART2 Special Mode Register 3	U2SMR3	000X0X0Xb
01F6h	UART2 Special Mode Register 2	U2SMR2	X0000000b
01F7h	UART2 Special Mode Register	U2SMR	X0000000b
01F8h	UART2 Transmit/Receive Mode Register	U2MR	00h
01F9h	UART2 Bit Rate Register	U2BRG	XXh
01FAh	UART2 Transmit Buffer Register	U2TB	XXh
01FBh			XXh
01FCh	UART2 Transmit/Receive Control Register 0	U2C0	00001000b
01FDh	UART2 Transmit/Receive Control Register 1	U2C1	00000010b
01FEh	UART2 Receive Buffer Register	U2RB	XXh
01FFh			XXh

X: Undefined

NOTES:
1. These registers exist only in the 128-pin version.
2. Bits S5TRF and S6TRF in the S3456TRR register are used in the 128-pin version.
3. Blank spaces are reserved. No access is allowed.

Table 4.9 SFR Information (9) [1]

Address	Register	Symbol	After Reset
0200h	CAN0 Message Control Register 0	C0MCTL0	00h
0201h	CAN0 Message Control Register 1	C0MCTL1	00h
0202h	CAN0 Message Control Register 2	C0MCTL2	00h
0203h	CAN0 Message Control Register 3	C0MCTL3	00h
0204h	CAN0 Message Control Register 4	C0MCTL4	00h
0205h	CAN0 Message Control Register 5	C0MCTL5	00h
0206h	CAN0 Message Control Register 6	C0MCTL6	00h
0207h	CAN0 Message Control Register 7	C0MCTL7	00h
0208h	CAN0 Message Control Register 8	C0MCTL8	00h
0209h	CAN0 Message Control Register 9	C0MCTL9	00h
020Ah	CAN0 Message Control Register 10	C0MCTL10	00h
020Bh	CAN0 Message Control Register 11	C0MCTL11	00h
020Ch	CAN0 Message Control Register 12	C0MCTL12	00h
020Dh	CAN0 Message Control Register 13	C0MCTL13	00h
020Eh	CAN0 Message Control Register 14	C0MCTL14	00h
020Fh	CAN0 Message Control Register 15	C0MCTL15	00h
0210h	CAN0 Control Register	C0CTLR	X0000001b
0211h			XX0X0000b
0212h	CAN0 Status Register	C0STR	00h
0213h			X0000001b
0214h	CAN0 Slot Status Register	C0SSTR	00h
0215h			00h
0216h	CAN0 Interrupt Control Register	C0ICR	00h
0217h			00h
0218h	CAN0 Extended ID Register	C0IDR	00h
0219h			00h
021Ah	CAN0 Configuration Register	C0CONR	XXh
021Bh			XXh
021Ch	CAN0 Receive Error Count Register	C0RECR	00h
021Dh	CAN0 Transmit Error Count Register	C0TECR	00h
021Eh	CAN0 Time Stamp Register	C0TSR	00h
021Fh			00h
0220h			
0221h			
0222h			
0223h			
0224h			
0225h			
0226h			
0227h			
0228h			
0229h			
022Ah			
022Bh			
022Ch			
022Dh			
022Eh			
022Fh			
0230h	CAN1 Control Register	C1CTLR	X0000001b
0231h			XX0X0000b
0232h			
0233h			
0234h			
0235h			
0236h			
0237h			
0238h			
0239h			
023Ah			
023Bh			
023Ch			
023Dh			
023Eh			
023Fh			

X: Undefined

NOTE:

1. Blank spaces are reserved. No access is allowed.

Table 4.10 SFR Information (10) [(1)]

Address	Register	Symbol	After Reset
0240h			
0241h			
0242h	CAN0 Acceptance Filter Support Register	C0AFS	XXh
0243h			XXh
0244h			
0245h			
0246h			
0247h			
0248h			
0249h			
024Ah			
024Bh			
024Ch			
024Dh			
024Eh			
024Fh			
0250h			
0251h			
0252h			
0253h			
0254h			
0255h			
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh	Peripheral Clock Select Register	PCLKR	00h
025Fh	CAN0 Clock Select Register	CCLKR	00h
0260h			
0261h			
0262h			
0263h			
0264h			
0265h			
0266h			
0267h			
0268h			
0269h			
026Ah			
026Bh			
026Ch			
026Dh			
026Eh			
026Fh			
0270h to 0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h			
0379h			
037Ah			
037Bh			
037Ch			
037Dh			
037Eh			
037Fh			

X: Undefined

NOTE:
1. Blank spaces are reserved. No access is allowed.

Table 4.11 SFR Information (11) (2)

Address	Register	Symbol	After Reset
0380h	Count Start Flag	TABSR	00h
0381h	Clock Prescaler Reset Flag	CPSRF	0XXXXXXXb
0382h	One-Shot Start Flag	ONSF	00h
0383h	Trigger Select Register	TRGSR	00h
0384h	Up/Down Flag	UDF	00h (1)
0385h			
0386h	Timer A0 Register	TA0	XXh
0387h			XXh
0388h	Timer A1 Register	TA1	XXh
0389h			XXh
038Ah	Timer A2 Register	TA2	XXh
038Bh			XXh
038Ch	Timer A3 Register	TA3	XXh
038Dh			XXh
038Eh	Timer A4 Register	TA4	XXh
038Fh			XXh
0390h	Timer B0 Register	TB0	XXh
0391h			XXh
0392h	Timer B1 Register	TB1	XXh
0393h			XXh
0394h	Timer B2 Register	TB2	XXh
0395h			XXh
0396h	Timer A0 Mode Register	TA0MR	00h
0397h	Timer A1 Mode Register	TA1MR	00h
0398h	Timer A2 Mode Register	TA2MR	00h
0399h	Timer A3 Mode Register	TA3MR	00h
039Ah	Timer A4 Mode Register	TA4MR	00h
039Bh	Timer B0 Mode Register	TB0MR	00XX0000b
039Ch	Timer B1 Mode Register	TB1MR	00XX0000b
039Dh	Timer B2 Mode Register	TB2MR	00XX0000b
039Eh	Timer B2 Special Mode Register	TB2SC	XXXXXX00b
039Fh			
03A0h	UART0 Transmit/Receive Mode Register	U0MR	00h
03A1h	UART0 Bit Rate Register	U0BRG	XXh
03A2h	UART0 Transmit Buffer Register	U0TB	XXh
03A3h			XXh
03A4h	UART0 Transmit/Receive Control Register 0	U0C0	00001000b
03A5h	UART0 Transmit/Receive Control Register 1	U0C1	00XX0010b
03A6h	UART0 Receive Buffer Register	U0RB	XXh
03A7h			XXh
03A8h	UART1 Transmit/Receive Mode Register	U1MR	00h
03A9h	UART1 Bit Rate Register	U1BRG	XXh
03AAh	UART1 Transmit Buffer Register	U1TB	XXh
03ABh			XXh
03ACh	UART1 Transmit/Receive Control Register 0	U1C0	00001000b
03ADh	UART1 Transmit/Receive Control Register 1	U1C1	00XX0010b
03AEh	UART1 Receive Buffer Register	U1RB	XXh
03AFh			XXh
03B0h	UART Transmit/Receive Control Register 2	UCON	X0000000b
03B1h			
03B2h			
03B3h			
03B4h			
03B5h			
03B6h			
03B7h			
03B8h	DMA0 Request Source Select Register	DM0SL	00h
03B9h			
03BAh	DMA1 Request Source Select Register	DM1SL	00h
03BBh			
03BCh	CRC Data Register	CRCD	XXh
03BDh			XXh
03BEh	CRC Input Register	CRCIN	XXh
03BFh			

X: Undefined

NOTES:
1. Bits TA2P to TA4P in the UDF register are set to 0 after reset. However, the contents in these bits are undefined when read.
2. Blank spaces are reserved. No access is allowed.

Table 4.12 SFR Information (12) [3]

Address	Register	Symbol	After Reset
03C0h	A/D Register 0	AD0	XXh
03C1h			XXh
03C2h	A/D Register 1	AD1	XXh
03C3h			XXh
03C4h	A/D Register 2	AD2	XXh
03C5h			XXh
03C6h	A/D Register 3	AD3	XXh
03C7h			XXh
03C8h	A/D Register 4	AD4	XXh
03C9h			XXh
03CAh	A/D Register 5	AD5	XXh
03CBh			XXh
03CCh	A/D Register 6	AD6	XXh
03CDh			XXh
03CEh	A/D Register 7	AD7	XXh
03CFh			XXh
03D0h			
03D1h			
03D2h			
03D3h			
03D4h	A/D Control Register 2	ADCON2	00h
03D5h			
03D6h	A/D Control Register 0	ADCON0	00000XXXb
03D7h	A/D Control Register 1	ADCON1	00h
03D8h	D/A Register 0	DA0	00h
03D9h			
03DAh	D/A Register 1	DA1	00h
03DBh			
03DCh	D/A Control Register	DACON	00h
03DDh			
03DEh	Port P14 Control Register [2]	PC14	XX00XXXXb
03DFh	Pull-Up Control Register 3 [2]	PUR3	00h
03E0h	Port P0 Register	P0	XXh
03E1h	Port P1 Register	P1	XXh
03E2h	Port P0 Direction Register	PD0	00h
03E3h	Port P1 Direction Register	PD1	00h
03E4h	Port P2 Register	P2	XXh
03E5h	Port P3 Register	P3	XXh
03E6h	Port P2 Direction Register	PD2	00h
03E7h	Port P3 Direction Register	PD3	00h
03E8h	Port P4 Register	P4	XXh
03E9h	Port P5 Register	P5	XXh
03EAh	Port P4 Direction Register	PD4	00h
03EBh	Port P5 Direction Register	PD5	00h
03ECh	Port P6 Register	P6	XXh
03EDh	Port P7 Register	P7	XXh
03EEh	Port P6 Direction Register	PD6	00h
03EFh	Port P7 Direction Register	PD7	00h
03F0h	Port P8 Register	P8	XXh
03F1h	Port P9 Register	P9	XXh
03F2h	Port P8 Direction Register	PD8	00X00000b
03F3h	Port P9 Direction Register	PD9	00h
03F4h	Port P10 Register	P10	XXh
03F5h	Port P11 Register [2]	P11	XXh
03F6h	Port P10 Direction Register	PD10	00h
03F7h	Port P11 Direction Register [2]	PD11	00h
03F8h	Port P12 Register [2]	P12	XXh
03F9h	Port P13 Register [2]	P13	XXh
03FAh	Port P12 Direction Register [2]	PD12	00h
03FBh	Port P13 Direction Register [2]	PD13	00h
03FCh	Pull-up Control Register 0	PUR0	00h
03FDh	Pull-up Control Register 1	PUR1	00000000b [1] 00000010b
03FEh	Pull-up Control Register 2	PUR2	00h
03FFh	Port Control Register	PCR	00h

X: Undefined

NOTES:

1. At hardware reset, the register is as follows:
   - 00000000b where "L" is input to the CNVSS pin
   - 00000010b where "H" is input to the CNVSS pin

   At software reset, watchdog timer reset and oscillation stop detection reset, the register is as follows:
   - 00000000b where the PM01 to PM00 bits in the PM0 register are 00b (single-chip mode)
   - 00000010b where the PM01 to PM00 bits in the PM0 register are 01b (memory expansion mode) or 11b (microprocessor mode)
2. These registers exist only in the128-pin version.
3. Blank spaces are reserved. No access is allowed.

5. Resets

Hardware reset, software reset, watchdog timer reset, and oscillation stop detection reset are available to reset the MCU.

5.1 Hardware Reset

The MCU resets pins, the CPU and SFR by setting the $\overline{\text{RESET}}$ pin. If the supply voltage meets the recommended operating conditions, the MCU resets all pins when an “L” signal is applied to the $\overline{\text{RESET}}$ pin (see **Table 5.1 Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”**). The oscillation circuit is also reset and the main clock starts oscillation. The MCU resets the CPU and SFR when the signal applied to the $\overline{\text{RESET}}$ pin changes low (“L”) to high (“H”). The MCU executes the program in an address indicated by the reset vector. The internal RAM is not reset. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while writing data to the internal RAM, the internal RAM is in an undefined state.

Figure 5.1 shows an Example Reset Circuit. Figure 5.2 shows a Reset Sequence. Table 5.1 lists the Pin States when $\overline{\text{RESET}}$ Pin Level is “L”.

5.1.1 Reset on a Stable Supply Voltage

(1) Apply “L” to the $\overline{\text{RESET}}$ pin

(2) Apply 20 or more clock cycles to the XIN pin

(3) Apply “H” to the $\overline{\text{RESET}}$ pin

5.1.2 Power-on Reset

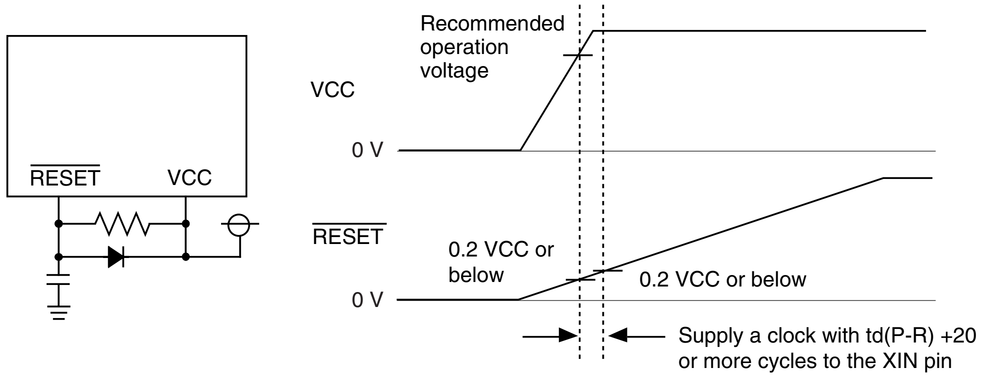
(1) Apply “L” to the $\overline{\text{RESET}}$ pin

(2) Raise the supply voltage to the recommended operating level

(3) Insert td(P-R) ms as wait time for the internal voltage to stabilize

(4) Apply 20 or more clock cycles to the XIN pin

(5) Apply “H” to the $\overline{\text{RESET}}$ pin


RESET
VCC
Recommended operation voltage
VCC
0 V
RESET
0.2 VCC or below
0.2 VCC or below
0 V
Supply a clock with td(P-R) +20 or more cycles to the XIN pin

NOTE
1. Use the shortest possible wiring to connect external circuit.

Figure 5.1 Example Reset Circuit

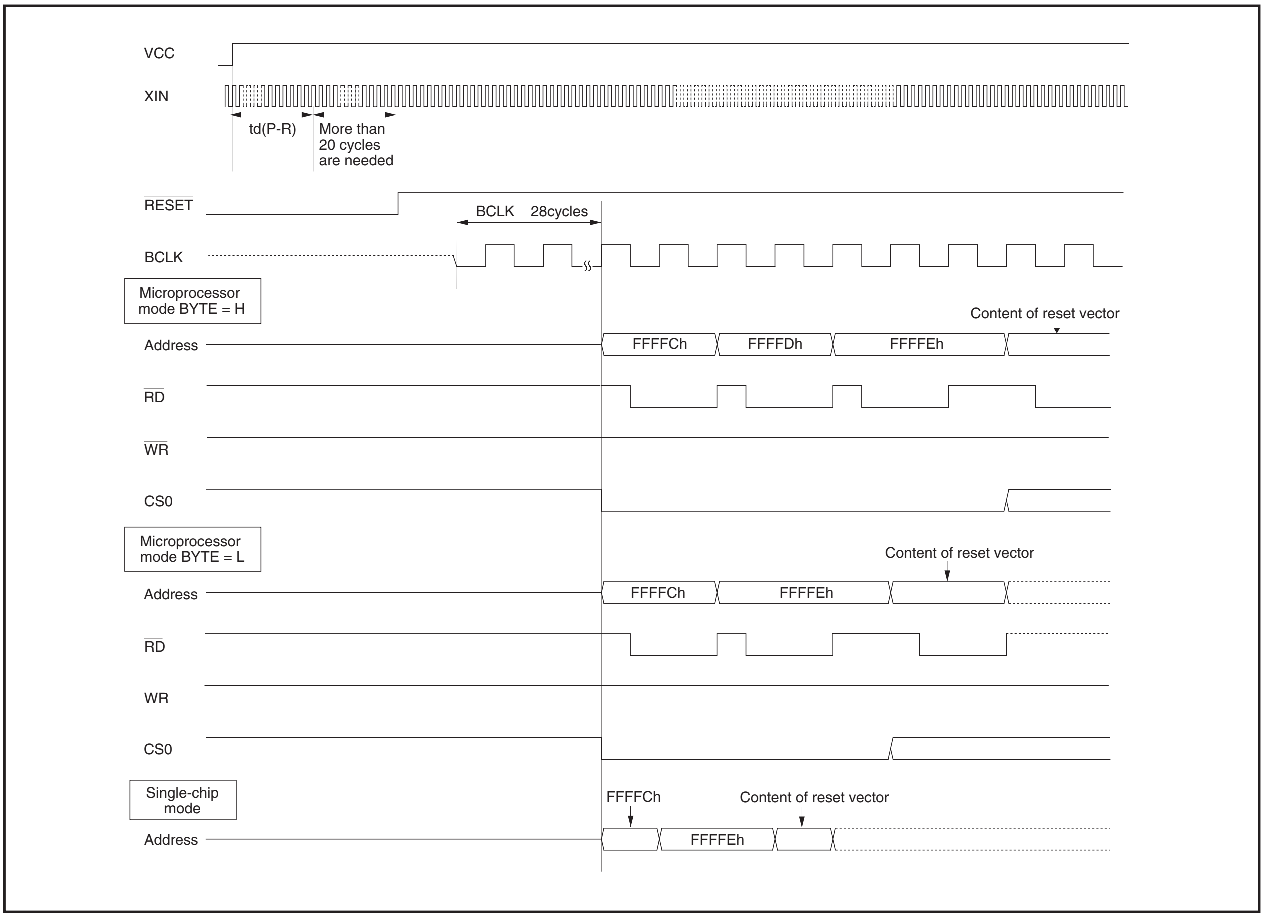


Figure 5.2 Reset Sequence

Table 5.1 Pin Status when RESET Pin Level is “L”

Pin Name	Status		
	CNVSS = VSS	CNVSS = VCC [(1)]	
		BYTE = VSS	BYTE = VCC
P0	Input port	Data input	Data input
P1	Input port	Data input	Input port
P2, P3, P4_0 to P4_3	Input port	Address output (undefined)	Address output (undefined)
P4_4	Input port	CS0 output (“H” is output)	CS0 output (“H” is output)
P4_5 to P4_7	Input port	Input port (Pulled high)	Input port (Pulled high)
P5_0	Input port	WR output (“H” is output)	WR output (“H” is output)
P5_1	Input port	BHE output (undefined)	BHE output (undefined)
P5_2	Input port	RD output (“H” is output)	RD output (“H” is output)
P5_3	Input port	BCLK output	BCLK output
P5_4	Input port	HLDA output (The output value depends on the input to the HOLD pin)	HLDA output (The output value depends on the input to the HOLD pin)
P5_5	Input port	HOLD input	HOLD input
P5_6	Input port	ALE output (“L” is output)	ALE output (“L” is output)
P5_7	Input port	RDY input	RDY input
P6, P7, P8_0 to P8_4, P8_6, P8_7, P9, P10	Input port	Input port	Input port
P11, P12, P13, P14_0, P14_1 [(2)]	Input port	Input port	Input port

NOTES:

1. Shown here is the valid pin state when the internal power supply voltage has stabilized after power-on. When CNVSS = VCC, the pin state is indeterminate until the internal power supply voltage stabilizes.
2. Pins P11, P12, P13, P14_0, and P14_1 are only in the 128-pin version.

5.2 Software Reset

The MCU resets pins, the CPU and SFR when the PM03 bit in the PM0 register is set to 1 (MCU reset). Then the MCU executes the program in an address determined by the reset vector.

Set the PM03 bit to 1 while the main clock is selected as the CPU clock and the main clock oscillation is stable.

In the software reset, the MCU does not reset a part of the SFR. Refer to **4. Special Function Registers (SFRs)** for details.

Processor mode remains unchanged since bits PM01 to PM00 in the PM0 register are not reset.

5.3 Watchdog Timer Reset

The MCU resets pins, the CPU and SFR when the PM12 bit in the PM1 register is set to 1 (reset when watchdog timer underflows) and the watchdog timer underflows. Then the MCU executes the program in an address determined by the reset vector.

In the watchdog timer reset, the MCU does not reset a part of the SFR. Refer to **4. Special Function Registers (SFRs)** for details.

Processor mode remains unchanged since bits PM01 to PM00 in the PM0 register are not reset.

5.4 Oscillation Stop Detection Reset

The MCU resets and stops pins, the CPU and SFR when the CM27 bit in the CM2 register is 0 (reset at oscillation stop, re-oscillation detection), if it detects main clock oscillation circuit stop. Refer to **8.5 Oscillation Stop and Re-Oscillation Detection Function** for details.

In the oscillation stop detection reset, the MCU does not reset a part of the SFR. Refer to **4. Special Function Registers (SFRs)** for details.

Processor mode remains unchanged since bits PM01 to PM00 in the PM0 register are not reset.

5.5 Internal Space

Figure 5.3 shows CPU Register Status After Reset. Refer to **4. Special Function Registers (SFRs)** for SFR states after reset.

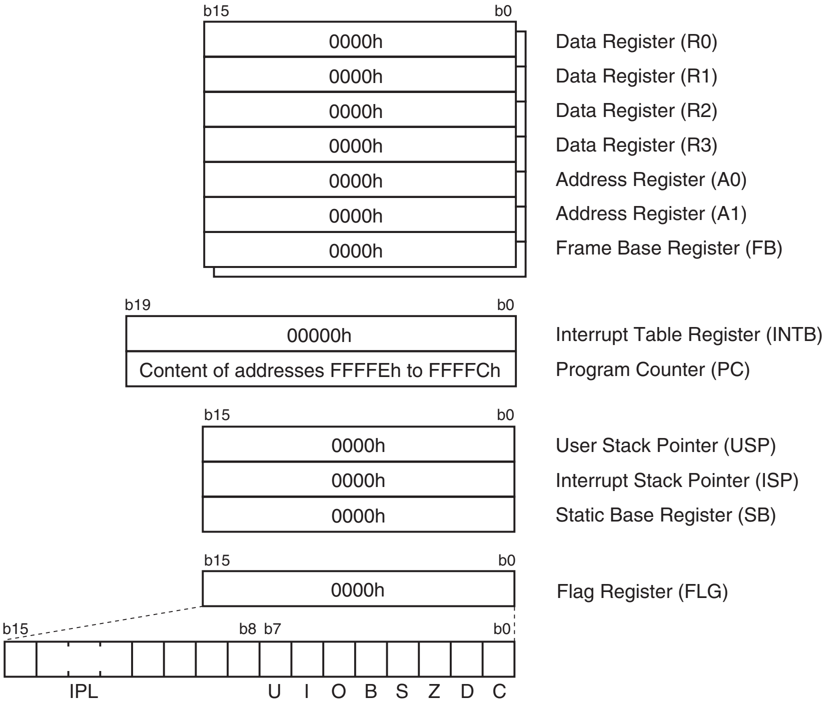


Figure 5.3 CPU Register Status After Reset

6. Processor Mode

6.1 Types of Processor Mode

Three processor modes are available to choose from: single-chip mode, memory expansion mode, and microprocessor mode. Table 6.1 shows the Features of Processor Modes.

Table 6.1 Features of Processor Modes

Processor Mode	Access Space	Pins Which are Assigned I/O Ports
Single-chip mode	SFR, internal RAM, internal ROM	All pins are I/O ports or peripheral function I/O pins
Memory expansion mode	SFR, internal RAM, internal ROM, external area (1)	Some pins serve as bus control pins (1)
Microprocessor mode	SFR, internal RAM, external area (1)	Some pins serve as bus control pins (1)

NOTE:

1. Refer to **7. Bus**.

6.2 Setting Processor Modes

Processor mode is set by using the CNVSS pin and bits PM01 to PM00 in the PM0 register.

Table 6.2 shows the Processor Mode after Hardware Reset. Table 6.3 shows Bits PM01 to PM00 Set Values and Processor Modes.

Table 6.2 Processor Mode after Hardware Reset

CNVSS Pin Input Level	Processor Mode
VSS	Single-chip mode
VCC (1) (2)	Microprocessor mode

NOTES:

1. If the MCU is reset in hardware by applying VCC to the CNVSS pin, the internal ROM cannot be accessed regardless of bits PM01 to PM00.
2. The multiplexed bus cannot be assigned to the entire $\overline{\text{CS}}$ space.

Table 6.3 Bits PM01 to PM00 Set Values and Processor Modes

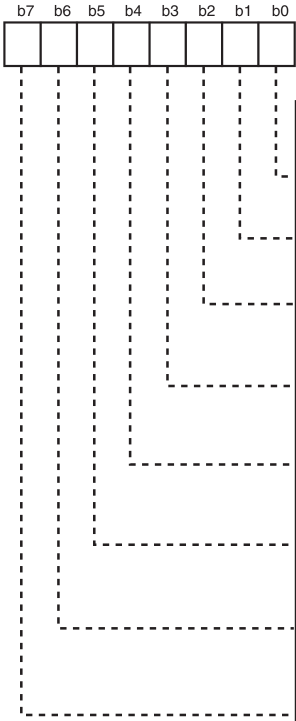
Bits PM01 to PM00	Processor Mode
00b	Single-chip mode
01b	Memory expansion mode
10b	Do not set a value
11b	Microprocessor mode

Rewriting bits PM01 to PM00 places the MCU in the corresponding processor mode regardless of whether the input level on the CNVSS pin is “H” or “L”. Note, however, that bits PM01 to PM00 cannot be rewritten to 01b (memory expansion mode) or 11b (microprocessor mode) at the same time bits PM07 to PM02 are rewritten. Note also that these bits cannot be rewritten to enter microprocessor mode in the internal ROM, nor can they be rewritten to exit microprocessor mode in areas overlapping the internal ROM.

If the MCU is reset in hardware by applying VCC to the CNVSS pin (hardware reset), the internal ROM cannot be accessed regardless of bits PM01 to PM00.

Figures 6.1 and 6.2 show the PM0 Register and PM1 Register. Figure 6.3 shows the Memory Map in Single-chip Mode. Figures 6.4 to 6.7 show the Memory Map and $\overline{\text{CS}}$ Area in Memory Expansion Mode and Microprocessor Mode.

Processor Mode Register 0 [1]



Symbol	Address	After Reset [2]
PM0	0004h	00000000b (CNVSS pin = L) 00000011b (CNVSS pin = H)

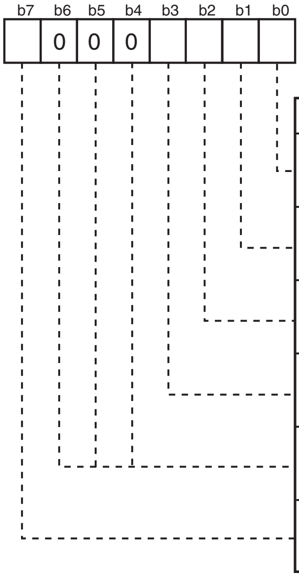
Bit Symbol	Bit Name	Function	RW
PM00	Processor mode bits [2]	b1 b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Do not set a value 1 1 : Microprocessor mode	RW
PM01			RW
PM02	R/W Mode select bit [3]	0 : $\overline{RD}$, $\overline{BHE}$, $\overline{WR}$ 1 : $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$	RW
PM03	Software reset bit	Setting this bit to 1 resets the MCU. When read, the content is 0.	RW
PM04	Multiplexed bus space select bits [3]	b5 b4 0 0 : Multiplexed bus is unused (Separate bus in the entire $\overline{CS}$ space) 0 1 : Allocated to $\overline{CS2}$ space 1 0 : Allocated to $\overline{CS1}$ space 1 1 : Allocated to the entire $\overline{CS}$ space [4]	RW
PM05			RW
PM06	Port P4_0 to P4_3 function select bit [3]	0 : Address output 1 : Port function (Address is not output)	RW
PM07	BCLK output disable bit [3]	0 : BCLK is output 1 : BCLK is not output (Pin is left high-impedance)	RW

NOTES:

1. Rewrite this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. Bits PM01 to PM00 do not change at software reset, watchdog timer reset and oscillation stop detection reset.
3. Effective when bits PM01 to PM00 are set to 01b (memory expansion mode) or 11b (microprocessor mode).
4. To set bits PM01 to PM00 are 01b and bits PM05 to PM04 are 11b (multiplexed bus assigned to the entire $\overline{CS}$ space), apply an "H" signal to the BYTE pin (external data bus is 8-bit width).
   While the CNVSS pin is held "H" (VCC), do not rewrite bits PM05 to PM04 to 11b after reset.
   If bits PM05 to PM04 are set to 11b during memory expansion mode, P3_1 to P3_7 and P4_0 to P4_3 become I/O ports, in which case the accessible area for each $\overline{CS}$ is 256 bytes.

Figure 6.1 PM0 Register

Processor Mode Register 1 [1]



Symbol	Address	After Reset
PM1	0005h	00001000b

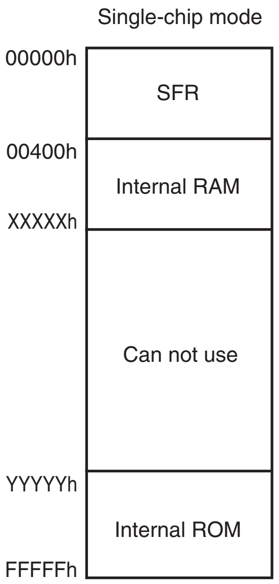
Bit Symbol	Bit Name	Function	RW
PM10	$\overline{\text{CS2}}$ area switch bit (Data block enable bit) [2]	0 : 08000h to 26FFFh (Block A disable) 1 : 10000h to 26FFFh (Block A enable)	RW
PM11	Port P3_7 to P3_4 function select bit [3]	0 : Address output 1 : Port function	RW
PM12	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Watchdog timer reset [4]	RW
PM13	Internal reserved area expansion bit [5]	See **NOTE 7**	RW
- (b6-b4)	Reserved bits	Set to 0	RW
PM17	Wait bit [6]	0 : No wait state 1 : With wait state (1 wait)	RW

NOTES:

1. Rewrite this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. For the mask ROM version, this bit is set to 0.
   For the flash memory version, the PM10 bit controls whether block A is enabled or disabled. When the PM10 bit is set to 1, 0F000h to 0FFFFh (block A) can be used as internal ROM area.
   In addition, the PM10 bit is automatically set to 1 while the FMR01 bit in the FMR0 register is set to 1 (CPU rewrite mode).
3. Effective when bits PM01 to PM00 are set to 01b (memory expansion mode) or 11b (microprocessor mode).
4. The PM12 bit is set to 1 by writing a 1 in a program. (writing a 0 has no effect.)
5. Be sure to set this bit to 0 except for products with internal ROM area over 192 Kbytes.
   The PM13 bit is automatically set to 1 while the FMR01 bit in the FMR0 register is set to 1 (CPU rewrite mode).
6. When the PM17 bit is set to 1 (with wait state), one wait state is inserted when accessing the internal RAM or internal ROM.
   When the PM17 bit is set to 1 and accesses an external area, set the CSiW bit (i = 0 to 3) in the CSR register to 0 (with wait state).
7. The access area is changed by the PM13 bit as listed in the table below.

Access Area		PM13 = 0	PM13 = 1
Internal	RAM	Up to addresses 00400h to 03FFFh (15 Kbytes)	The entire area is usable
	ROM	Up to addresses D0000h to FFFFFh (192 Kbytes)	The entire area is usable
External		Addresses 04000h to 07FFFh are usable Addresses 80000h to CFFFFh are usable	Addresses 04000h to 07FFFh are reserved Addresses 80000h to CFFFFh are reserved (Memory expansion mode)

Figure 6.2 PM1 Register



PM13 bit in PM1 register = 0 [1]

Internal RAM		Internal ROM	
Capacity	Address XXXXXh	Capacity	Address YYYYYh
16 Kbytes	03FFFh	192 Kbytes	D0000h
20 Kbytes	03FFFh	256 Kbytes	D0000h
31 Kbytes	03FFFh	384 Kbytes	D0000h
		512 Kbytes	D0000h

PM13 bit = 1

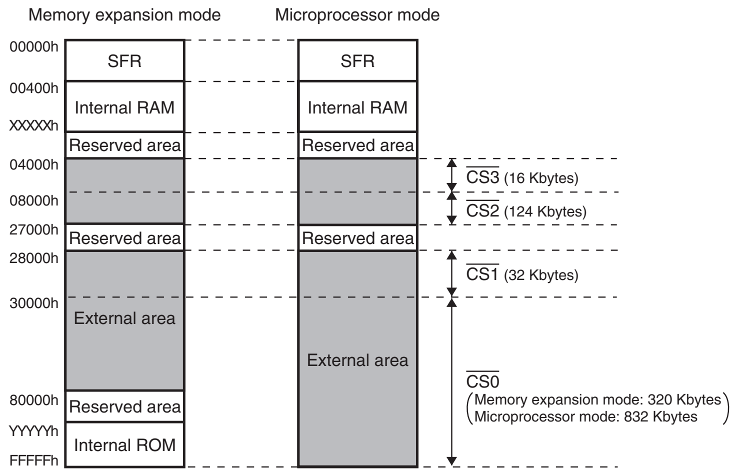
Internal RAM		Internal ROM	
Capacity	Address XXXXXh	Capacity	Address YYYYYh
16 Kbytes	043FFh	192 Kbytes	D0000h
20 Kbytes	053FFh	256 Kbytes	C0000h
31 Kbytes	07FFFh	384 Kbytes	A0000h
		512 Kbytes	80000h

NOTES:

1. If the PM13 bit in the PM1 register is set to 0, 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.
2. For the mask ROM version, set the PM10 bit in the PM1 register to 0 (block A disabled, addresses 08000h to 26FFFh for $\overline{CS2}$ area).

Figure 6.3 Memory Map in Single-chip Mode

■ When PM13 = 0 and PM10 = 0



Internal RAM	
Capacity	Address XXXXXh [1]
16 Kbytes	03FFFh
20 Kbytes	03FFFh
31 Kbytes	03FFFh

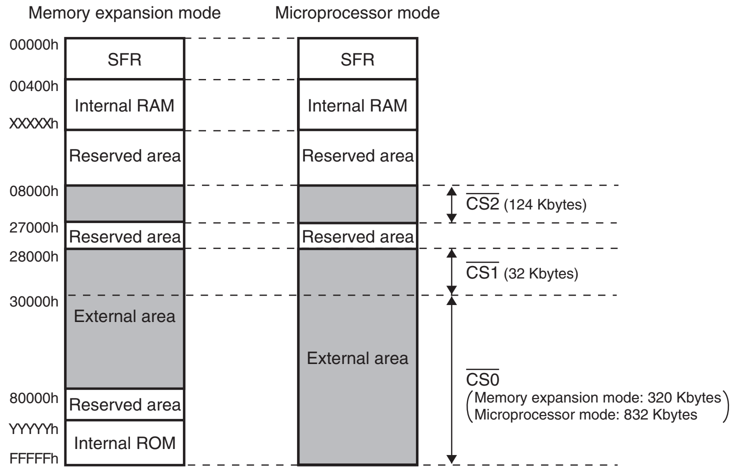
Internal ROM	
Capacity	Address YYYYYh [1]
192 Kbytes	D0000h
256 Kbytes	D0000h
384 Kbytes	D0000h
512 Kbytes	D0000h

NOTE:

1. If the PM13 bit in the PM1 register is set to 0, 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.

Figure 6.4 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (1)

■ When PM13 = 1 and PM10 = 0

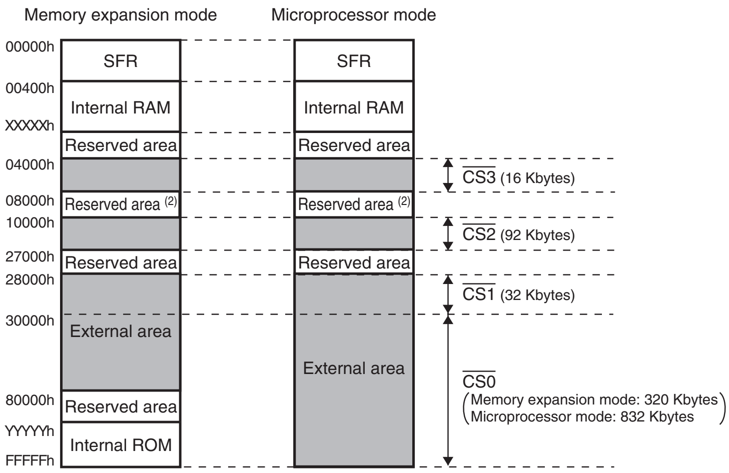


Internal RAM	
Capacity	Address XXXXXh
16 Kbytes	043FFh
20 Kbytes	053FFh
31 Kbytes	07FFFh

Internal ROM	
Capacity	Address YYYYYh
192 Kbytes	D0000h
256 Kbytes	C0000h
384 Kbytes	A0000h
512 Kbytes	80000h

Figure 6.5 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (2)

■ When PM13 = 0 and PM10 = 1



Internal RAM	
Capacity	Address XXXXXh (1)
16 Kbytes	03FFFh
20 Kbytes	03FFFh
31 Kbytes	03FFFh

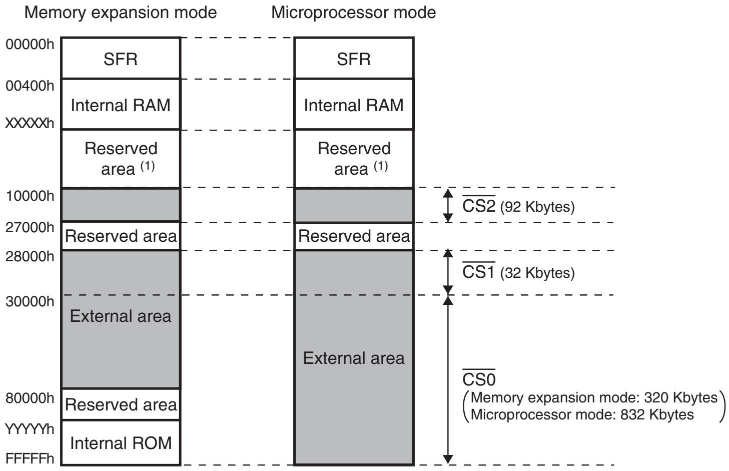
Internal ROM	
Capacity	Address YYYYYh (1)
192 Kbytes	D0000h
256 Kbytes	D0000h
384 Kbytes	D0000h
512 Kbytes	D0000h

NOTES:

1. If the PM13 bit in the PM1 register is set to 0, 15 Kbytes of the internal RAM and 192 Kbytes of the internal ROM can be used.
2. For the flash memory version, when the PM10 bit is set to 1, 0F000h to 0FFFFh (block A) can be used as internal ROM area.

Figure 6.6 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (3)

■ When PM13 = 1 and PM10 = 1



Internal RAM	
Capacity	Address XXXXXh
16 Kbytes	043FFh
20 Kbytes	053FFh
31 Kbytes	07FFFh

Internal ROM	
Capacity	Address YYYYYh
192 Kbytes	D0000h
256 Kbytes	C0000h
384 Kbytes	A0000h
512 Kbytes	80000h

NOTE:

1. For the flash memory version, when the PM10 bit is set to 1, 0F000h to 0FFFFh (block A) can be used as internal ROM area.

Figure 6.7 Memory Map and CS Area in Memory Expansion Mode and Microprocessor Mode (4)

7. Bus

During memory expansion or microprocessor mode, some pins serve as the bus control pins to perform data input/output to and from external devices. These bus control pins include A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}/\overline{\text{WR}}$, $\overline{\text{WRH}}/\overline{\text{BHE}}$, ALE, $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, and BCLK.

7.1 Bus Mode

The bus mode, either multiplexed or separate, can be selected using bits PM05 to PM04 in the PM0 register.

7.1.1 Separate Bus

In this bus mode, data and address are separate.

7.1.2 Multiplexed Bus

In this bus mode, data and address are multiplexed.

7.1.2.1 When the input level on BYTE pin is high (8-bit data bus)

D0 to D7 and A0 to A7 are multiplexed.

7.1.2.2 When the input level on BYTE pin is low (16-bit data bus)

D0 to D7 and A1 to A8 are multiplexed. D8 to D15 are not multiplexed. Do not use D8 to D15.
External devices connecting to a multiplexed bus are allocated to only the even addresses of the MCU. Odd addresses cannot be accessed.

Table 7.1 shows the Difference between Separate Bus and Multiplexed Bus.

Table 7.1 Difference between Separate Bus and Multiplexed Bus

Pin Name (1)	Separate Bus	Multiplexed Bus	
		BYTE = H	BYTE = L
P0_0 to P0_7/D0 to D7	D0 to D7	(NOTE 2)	(NOTE 2)
P1_0 to P1_7/D8 to D15	D8 to D15	I/O Port P1_0 to P1_7	(NOTE 2)
P2_0/A0(/D0/-)	A0	A0 D0	A0
P2_1 to P2_7/A1 to A7 (/D1 to D7/D0 to D6)	A1 to A7	A1 to A7 D1 to D7	A1 to A7 D0 to D6
P3_0/A8(/-/D7)	A8	A8	A8 D7

NOTES :

1. See **Table 7.6 Pin Functions for Each Processor Mode** for bus control signals other than the above.
2. It changes with a setup of bits PM05 to PM04 in the PM0 register, and area to access. See **Table 7.6 Pin Functions for Each Processor Mode** for details.

7.2 Bus Control

The following describes the signals needed for accessing external devices and the functionality of software wait.

7.2.1 Address Bus

The address bus consists of 20 lines, A0 to A19. The address bus width can be chosen to be 12, 16 or 20 bits by using the PM06 bit in the PM0 register and the PM11 bit in the PM1 register. Table 7.2 shows Bits PM06 and PM11 Set Values and Address Bus Widths.

When processor mode is changed from single-chip mode to memory expansion mode, the address bus is undefined until any external area is accessed.

Table 7.2 Bits PM06 and PM11 Set Value and Address Bus Width

Set Value [1]	Pin Function	Address Bus Width
PM11 = 1	P3_4 to P3_7	12 bits
PM06 = 1	P4_0 to P4_3	
PM11 = 0	A12 to A15	16 bits
PM06 = 1	P4_0 to P4_3	
PM11 = 0	A12 to A15	20 bits
PM06 = 0	A16 to A19	

NOTE:

1. No values other than those shown above can be set.

7.2.2 Data Bus

When input on the BYTE pin is high (data bus is an 8-bit width), 8 lines D0 to D7 comprise the data bus; when input on the BYTE pin is low (data bus is a 16-bit width), 16 lines D0 to D15 comprise the data bus. Do not change the input level on the BYTE pin while in operation.

7.2.3 Chip Select Signal

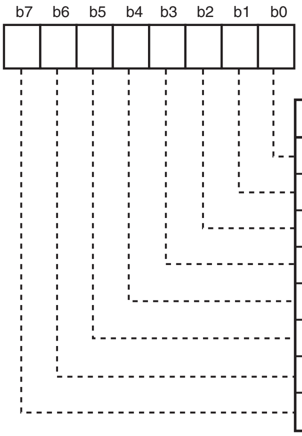
The chip select (hereafter referred to as the $\overline{\mathrm{CS}}$) signals are output from the $\overline{\mathrm{CSi}}$ (i = 0 to 3) pins. These pins can be chosen to function as I/O ports or as $\overline{\mathrm{CS}}$ by using the CSi bit in the CSR register.

Figure 7.1 shows the CSR Register.

During 1 Mbyte mode, the external area can be separated into up to 4 by the $\overline{\mathrm{CSi}}$ signal which is output from the $\overline{\mathrm{CSi}}$ pin.

Figure 7.2 shows the Example of Address Bus and $\overline{\mathrm{CSi}}$ Signal Output.

Chip Select Control Register



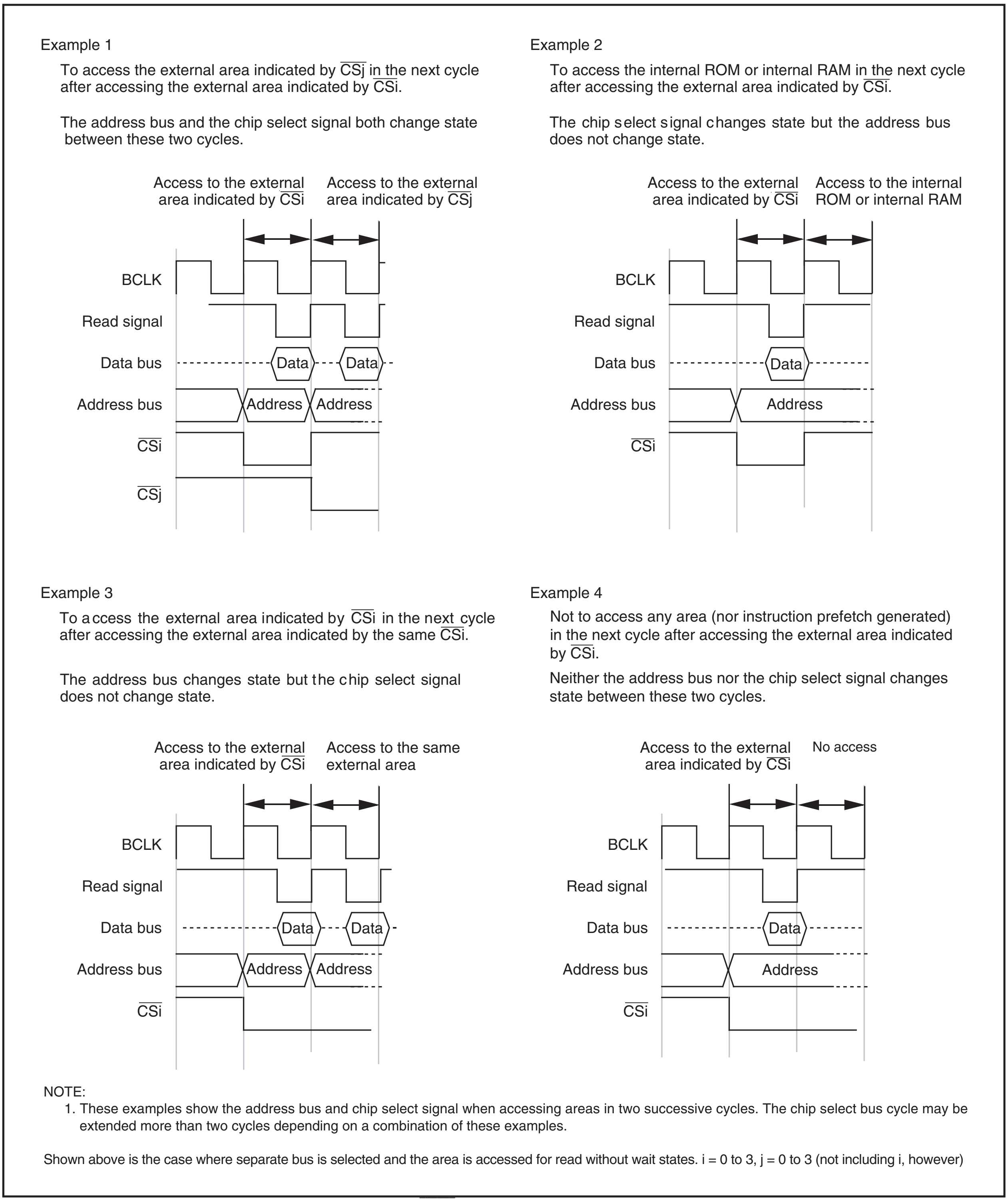
Symbol	Address	After Reset
CSR	0008h	00000001b

Bit Symbol	Bit Name	Function	RW
CS0	$\overline{\mathrm{CS0}}$ output enable bit	0 : Chip select output disabled (functions as I/O port) 1 : Chip select output enabled	RW
CS1	$\overline{\mathrm{CS1}}$ output enable bit		RW
CS2	$\overline{\mathrm{CS2}}$ output enable bit		RW
CS3	$\overline{\mathrm{CS3}}$ output enable bit		RW
CS0W	$\overline{\mathrm{CS0}}$ wait bit	0 : With wait state 1 : Without wait state [1] [2] [3]	RW
CS1W	$\overline{\mathrm{CS1}}$ wait bit		RW
CS2W	$\overline{\mathrm{CS2}}$ wait bit		RW
CS3W	$\overline{\mathrm{CS3}}$ wait bit		RW

NOTES:

1. Where the $\overline{\mathrm{RDY}}$ signal is used in the area indicated by $\overline{\mathrm{CSi}}$ (i = 0 to 3) or the multiplexed bus is used, set the CSiW bit to 0 (wait state).
2. If the PM17 bit in the PM1 register is set to 1 (with wait state), set the CSiW bit to 0 (with wait state).
3. When the CSiW bit = 0 (with wait state), the number of wait states (in terms of clock cycles) can be selected using bits CSEi1W to CSEi0W in the CSE register.

Figure 7.1 CSR Register



Example 1

To access the external area indicated by $\overline{CSj}$ in the next cycle after accessing the external area indicated by $\overline{CSi}$.

The address bus and the chip select signal both change state between these two cycles.

Access to the external area indicated by $\overline{CSi}$
Access to the external area indicated by $\overline{CSj}$
BCLK
Read signal
Data bus
Data
Data
Address bus
Address
Address
$\overline{CSi}$
$\overline{CSj}$

Example 2

To access the internal ROM or internal RAM in the next cycle after accessing the external area indicated by $\overline{CSi}$.

The chip select signal changes state but the address bus does not change state.

Access to the external area indicated by $\overline{CSi}$
Access to the internal ROM or internal RAM
BCLK
Read signal
Data bus
Data
Address bus
Address
$\overline{CSi}$

Example 3

To access the external area indicated by $\overline{CSi}$ in the next cycle after accessing the external area indicated by the same $\overline{CSi}$.

The address bus changes state but the chip select signal does not change state.

Access to the external area indicated by $\overline{CSi}$
Access to the same external area
BCLK
Read signal
Data bus
Data
Data
Address bus
Address
Address
$\overline{CSi}$

Example 4

Not to access any area (nor instruction prefetch generated) in the next cycle after accessing the external area indicated by $\overline{CSi}$.

Neither the address bus nor the chip select signal changes state between these two cycles.

NOTE:

1. These examples show the address bus and chip select signal when accessing areas in two successive cycles. The chip select bus cycle may be extended more than two cycles depending on a combination of these examples.

Shown above is the case where separate bus is selected and the area is accessed for read without wait states. i = 0 to 3, j = 0 to 3 (not including i, however)

Figure 7.2 Example of Address Bus and $\overline{CSi}$ Signal Output

7.2.4 Read and Write Signals

When the data bus is 16-bit width, the read and write signals can be chosen to be a combination of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, and $\overline{\mathrm{BHE}}$ or a combination of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WRL}}$, and $\overline{\mathrm{WRH}}$ by using the PM02 bit in the PM0 register. When the data bus is 8-bit width, use a combination of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, and $\overline{\mathrm{BHE}}$.

Table 7.3 shows the Operation of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WRL}}$, and $\overline{\mathrm{WRH}}$ Signals. Table 7.4 shows the Operation of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, and $\overline{\mathrm{BHE}}$ Signals.

Table 7.3 Operation of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WRL}}$, and $\overline{\mathrm{WRH}}$ Signals

Data Bus Width	$\overline{\mathrm{RD}}$	$\overline{\mathrm{WRL}}$	$\overline{\mathrm{WRH}}$	Status of External Data Bus
16 bits (BYTE pin input = L)	L	H	H	Read data
	H	L	H	Write 1 byte of data to an even address
	H	H	L	Write 1 byte of data to an odd address
	H	L	L	Write data to both even and odd addresses

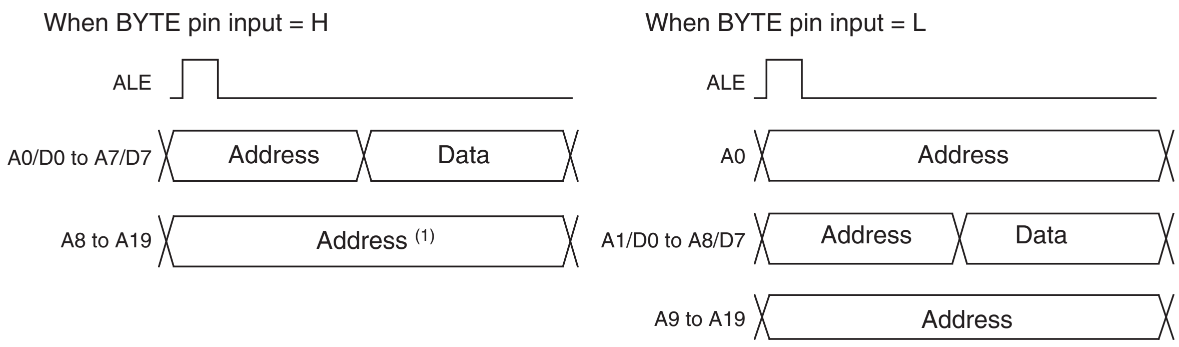
Table 7.4 Operation of $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, and $\overline{\mathrm{BHE}}$ Signals

Data Bus Width	$\overline{\mathrm{RD}}$	$\overline{\mathrm{WR}}$	$\overline{\mathrm{BHE}}$	A0	Status of External Data Bus
16 bits (BYTE pin input = L)	H	L	L	H	Write 1 byte of data to an odd address
	L	H	L	H	Read 1 byte of data from an odd address
	H	L	H	L	Write 1 byte of data to an even address
	L	H	H	L	Read 1 byte of data from an even address
	H	L	L	L	Write data to both even and odd addresses
	L	H	L	L	Read data from both even and odd addresses
8 bits (BYTE pin input = H)	H	L	Not used	H to L	Write 1 byte of data
	L	H	Not used	H to L	Read 1 byte of data

7.2.5 ALE Signal

The ALE signal latches the address when accessing the multiplexed bus space. Latch the address when the ALE signal falls.

Figure 7.3 shows the ALE Signal, Address Bus and Data Bus.



NOTE:
1. If the entire $\overline{\mathrm{CS}}$ space is assigned a multiplexed bus, these pins function as I/O ports.

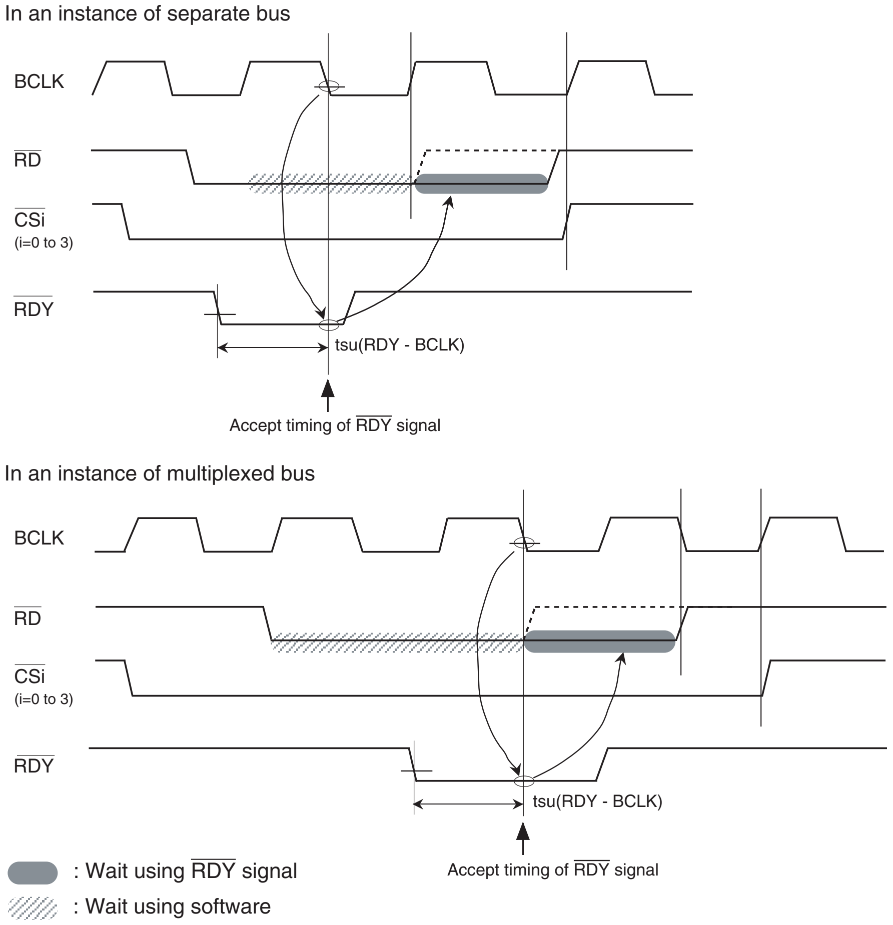
Figure 7.3 ALE Signal, Address Bus, and Data Bus

7.2.6 RDY Signal

This signal is provided for accessing external devices which need to be accessed at low speed. If input on the RDY pin is asserted low at the last falling edge of BCLK of the bus cycle, one wait state is inserted in the bus cycle. While in a wait state, the following signals retain the state in which they were when the RDY signal was acknowledged.

A0 to A19, D0 to D15, CS0 to CS3, RD, WRL, WRH, WR, BHE, ALE, HLDA

Then, when the input on the RDY pin is detected high at the falling edge of BCLK, the remaining bus cycle is executed. Figure 7.4 shows an Example in which Wait State was Inserted into Read Cycle by RDY Signal. To use the RDY signal, set the corresponding bit (bits CS3W to CS0W) in the CSR register to 0 (with wait state). When not using the RDY signal, the RDY pin must be pulled-up.



tsu(RDY-BCLK): RDY input setup time

Shown above is the case where bits CSEi1W to CSEi0W (i = 0 to 3) in the CSE register are 00b (one wait state).

Figure 7.4 Example in which Wait State was Inserted into Read Cycle by RDY Signal

7.2.7 $\overline{\text{HOLD}}$ Signal

This signal is used to transfer control of the bus from the CPU or DMAC to an external circuit. When the input on $\overline{\text{HOLD}}$ pin is pulled low, the MCU is placed in a hold state after the bus access then in process finishes. The MCU remains in a hold state while the $\overline{\text{HOLD}}$ pin is held low, during which time the $\overline{\text{HLDA}}$ pin outputs a low-level signal.

Table 7.5 shows the MCU Status in Hold State.

Bus-using priorities are given to $\overline{\text{HOLD}}$, DMAC, and CPU in order of decreasing precedence (see **Figure 7.5 Bus-using Priorities**). However, if the CPU is accessing an odd address in word units, the DMAC cannot gain control of the bus during two separate accesses.

$\overline{\text{HOLD}}$ > DMAC > CPU

Figure 7.5 Bus-using Priorities

Table 7.5 MCU Status in Hold State

Item		Status
BCLK		Output
A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$		High-impedance
I/O ports	P0, P1, P3, P4 [(1)]	High-impedance
	P6 to P14 [(3)]	Maintains status when hold signal is received
$\overline{\text{HLDA}}$		Output “L”
Internal peripheral circuits		ON (but watchdog timer stops [(2)])
ALE signal		Undefined

NOTES:

1. When I/O port function is selected.
2. The watchdog timer does not stop when the PM22 bit in the PM2 register is set to 1 (the count source for the watchdog timer is the on-chip oscillator clock).
3. Ports P11 to P14 are only in the 128-pin version.

7.2.8 BCLK Output

If the PM07 bit in the PM0 register is set to 0 (output enable), a clock with the same frequency as that of the CPU clock is output as BCLK from the BCLK pin. Refer to **8.2 CPU Clock and Peripheral Function Clock**.

Table 7.6 shows the Pin Functions for Each Processor Mode.

Table 7.6 Pin Functions for Each Processor Mode

Processor Mode		Memory Expansion Mode or Microprocessor Mode				Memory Expansion Mode
Bits PM05 to PM04		00b (separate bus)		01b ($\overline{\text{CS2}}$ is for multiplexed bus and others are for separate bus) 10b ($\overline{\text{CS1}}$ is for multiplexed bus and others are for separate bus)		11b (multiplexed bus for the entire space) [1]
Data bus width		8 bits	16 bits	8 bits	16 bits	8 bits
BYTE pin		“H”	“L”	“H”	“L”	“H”
P0_0 to P0_7		D0 to D7		D0 to D7 [4]		I/O ports
P1_0 to P1_7		I/O ports	D8 to D15	I/O ports	D8 to D15 [4]	I/O ports
P2_0		A0		A0/D0 [2]	A0	A0/D0
P2_1 to P2_7		A1 to A7		A1 to A7 /D1 to D7 [2]	A1 to A7 /D0 to D6 [2]	A1 to A7/D1 to D7
P3_0		A8			A8/D7 [2]	A8
P3_1 to P3_3		A9 to A11				I/O ports
P3_4 to P3_7	PM11 = 0	A12 to A15				I/O ports
	PM11 = 1	I/O ports				
P4_0 to P4_3	PM06 = 0	A16 to A19				I/O ports
	PM06 = 1	I/O ports				
P4_4	CS0 = 0	I/O ports				
	CS0 = 1	$\overline{\text{CS0}}$				
P4_5	CS1 = 0	I/O ports				
	CS1 = 1	$\overline{\text{CS1}}$				
P4_6	CS2 = 0	I/O ports				
	CS2 = 1	$\overline{\text{CS2}}$				
P4_7	CS3 = 0	I/O ports				
	CS3 = 1	$\overline{\text{CS3}}$				
P5_0	PM02 = 0	$\overline{\text{WR}}$				
	PM02 = 1	– [3]	$\overline{\text{WRL}}$	– [3]	$\overline{\text{WRL}}$	– [3]
P5_1	PM02 = 0	$\overline{\text{BHE}}$				
	PM02 = 1	– [3]	$\overline{\text{WRH}}$	– [3]	$\overline{\text{WRH}}$	– [3]
P5_2		$\overline{\text{RD}}$				
P5_3		BCLK				
P5_4		$\overline{\text{HLDA}}$				
P5_5		$\overline{\text{HOLD}}$				
P5_6		ALE				
P5_7		$\overline{\text{RDY}}$				

I/O ports: Function as I/O ports or peripheral function I/O pins.

NOTES:

1. For setting bits PM01 to PM00 to 01b (memory expansion mode) and bits PM05 to PM04 to 11b (multiplexed bus assigned to the entire $\overline{\text{CS}}$ space), apply “H” to the BYTE pin (external data bus is an 8-bit width). While the CNVSS pin is held “H” (VCC), do not rewrite bits PM05 to PM04 to 11b after reset. If bits PM05 to PM04 are set to 11b during memory expansion mode, P3_1 to P3_7 and P4_0 to P4_3 become I/O ports, in which case the accessible area for each $\overline{\text{CS}}$ is 256 bytes.
2. In separate bus mode, these pins serve as the address bus.
3. If the data bus is 8-bit width, make sure the PM02 bit is set to 0 ($\overline{\text{RD}}$, $\overline{\text{BHE}}$, $\overline{\text{WR}}$).
4. When accessing the area that uses a multiplexed bus, these pins output an undefined value during a write.

7.2.9 External Bus Status when Internal Area Accessed

Table 7.7 shows the External Bus Status When Internal Area Accessed.

Table 7.7 External Bus Status When Internal Area Accessed

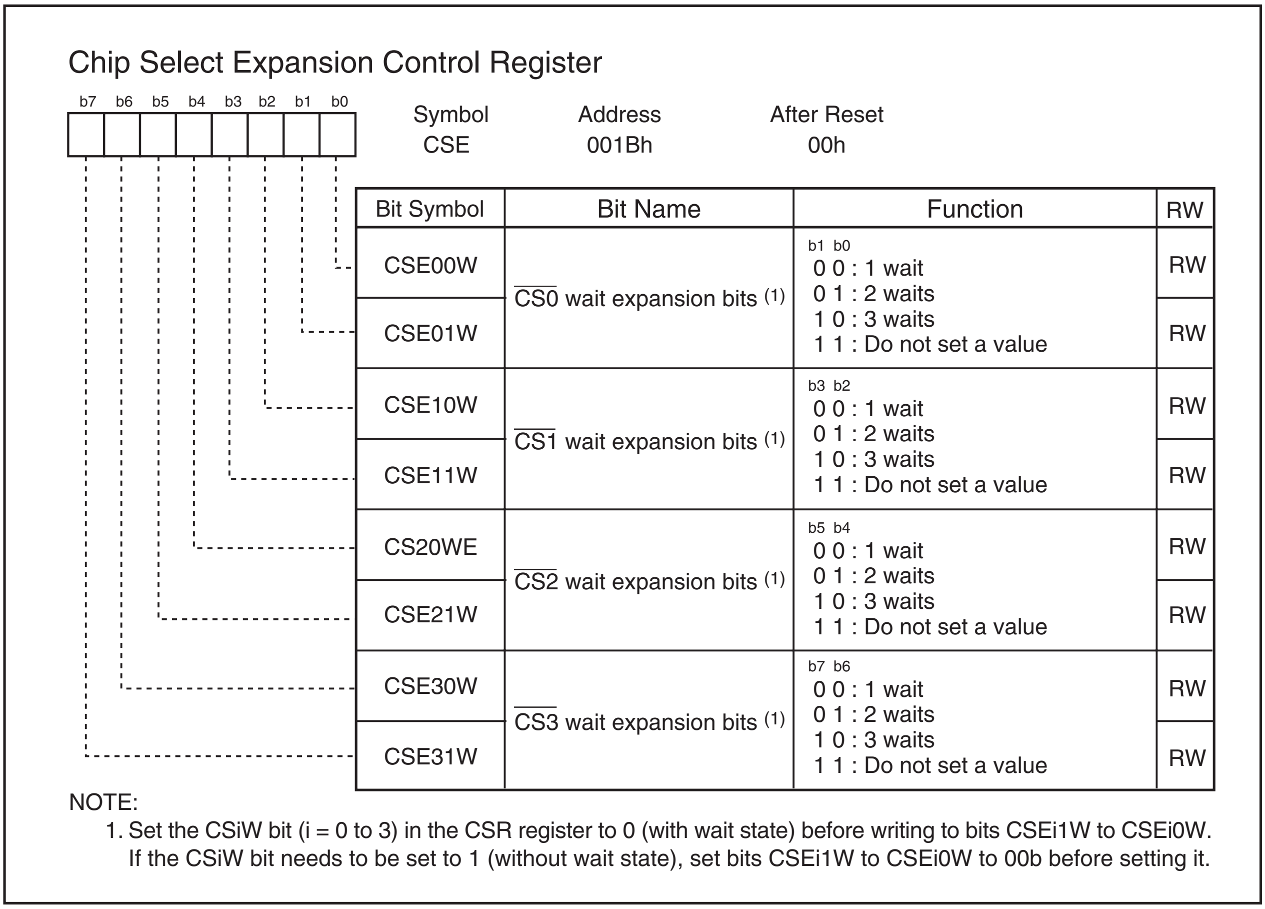
Item		SFR Accessed	Internal ROM, Internal RAM Accessed
A0 to A19		Address output	Maintain status before accessed address of external area or SFR
D0 to D15	When read	High-impedance	High-impedance
	When write	Output data	Undefined
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$		$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$ output	Output “H”
$\overline{\text{BHE}}$		$\overline{\text{BHE}}$ output	Maintain status before accessed status of external area or SFR
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$		Output “H”	Output “H”
ALE		Output “L”	Output “L”

7.2.10 Software Wait

Software wait states can be inserted by using the PM17 bit in the PM1 register, bits CS0W to CS3W in the CSR register, and the CSE register. The SFR area is unaffected by these control bits. This area is always accessed in 2 BCLK or 3 BCLK cycles as determined by the PM20 bit in the PM2 register. See **Table 7.8 Bit and Bus Cycle Related to Software Wait** for details.

To use the $\overline{\text{RDY}}$ signal, set the corresponding bit of bits CS3W to CS0W to 0 (with wait state).

Figure 7.6 shows the CSE Register. Table 7.8 shows the Software Wait Related Bits and Bus Cycles. Figures 7.7 and 7.8 show the Typical Bus Timings Using Software Wait.



Chip Select Expansion Control Register

Symbol	Address	After Reset
CSE	001Bh	00h

Bit Symbol	Bit Name	Function	RW
CSE00W	$\overline{\text{CS0}}$ wait expansion bits (1)	b1 b0 0 0 : 1 wait 0 1 : 2 waits 1 0 : 3 waits 1 1 : Do not set a value	RW
CSE01W			RW
CSE10W	$\overline{\text{CS1}}$ wait expansion bits (1)	b3 b2 0 0 : 1 wait 0 1 : 2 waits 1 0 : 3 waits 1 1 : Do not set a value	RW
CSE11W			RW
CS20WE	$\overline{\text{CS2}}$ wait expansion bits (1)	b5 b4 0 0 : 1 wait 0 1 : 2 waits 1 0 : 3 waits 1 1 : Do not set a value	RW
CSE21W			RW
CSE30W	$\overline{\text{CS3}}$ wait expansion bits (1)	b7 b6 0 0 : 1 wait 0 1 : 2 waits 1 0 : 3 waits 1 1 : Do not set a value	RW
CSE31W			RW

NOTE:

1. Set the CSiW bit (i = 0 to 3) in the CSR register to 0 (with wait state) before writing to bits CSEi1W to CSEi0W. If the CSiW bit needs to be set to 1 (without wait state), set bits CSEi1W to CSEi0W to 00b before setting it.

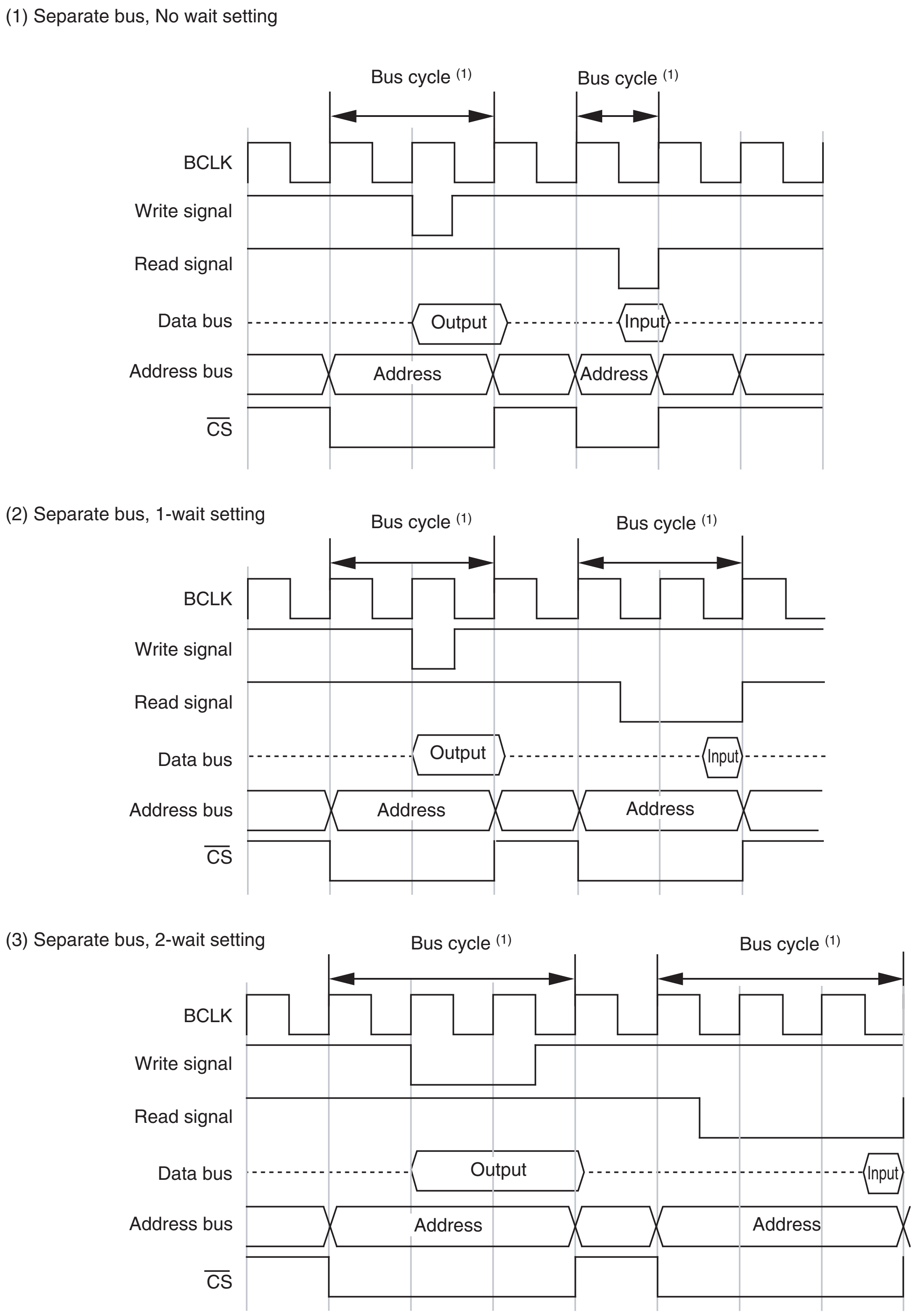
Figure 7.6 CSE Register

Table 7.8 Software Wait Related Bits and Bus Cycles

Area	Bus Mode	PM2 Register PM20 Bit	PM1 Register PM17 Bit [5]	CSR Register CS3W Bit [1] CS2W Bit [1] CS1W Bit [1] CS0W Bit [1]	CSE Register Bits CS31W to CS30W Bits CS21W to CS20W Bits CS11W to CS10W Bits CS01W to CS00W	Software Wait	Bus Cycle
SFR	–	0	–	–	–	–	3 BCLK cycles [4]
	–	1	–	–	–	–	2 BCLK cycles [4]
Internal ROM, RAM	–	–	0	–	–	No wait	1 BCLK cycle [3]
	–	–	1	–	–	1 wait	2 BCLK cycles
External area	Separate bus	–	0	1	00b	No wait	1 BCLK cycle (read)
							2 BCLK cycles (write)
		–	–	0	00b	1 wait	2 BCLK cycles [3]
		–	–	0	01b	2 waits	3 BCLK cycles
		–	–	0	10b	3 waits	4 BCLK cycles
		–	1	0	00b	1 wait	2 BCLK cycles
	Multiplexed bus [2]	–	–	0	00b	1 wait	3 BCLK cycles
		–	–	0	01b	2 waits	3 BCLK cycles
		–	–	0	10b	3 waits	4 BCLK cycles
		–	1	0	00b	1 wait	3 BCLK cycles

NOTES:

1. To use the $\overline{\text{RDY}}$ signal, set this bit to 0.
2. To access in multiplexed bus mode, set the corresponding bit of bits CS0W to CS3W to 0 (with wait state).
3. After reset, the PM17 bit is set to 0 (without wait state), all of bits CS0W to CS3W are set to 0 (with wait state), and the CSE register is set to 00h (one wait state for $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$). Therefore, the internal RAM and internal ROM are accessed with no wait state, and all external areas are accessed with one wait state.
4. When the selected CPU clock source is the PLL clock, the number of wait cycles can be altered by the PM20 bit in the PM2 register. When using PLL clock over 16 MHz, be sure to set the PM20 bit to 0 (2 wait cycles).
5. When the PM17 bit is set to 1 and access an external area, set the CSiW bits (i = 0 to 3) to 0 (with wait state).



(1) Separate bus, No wait setting

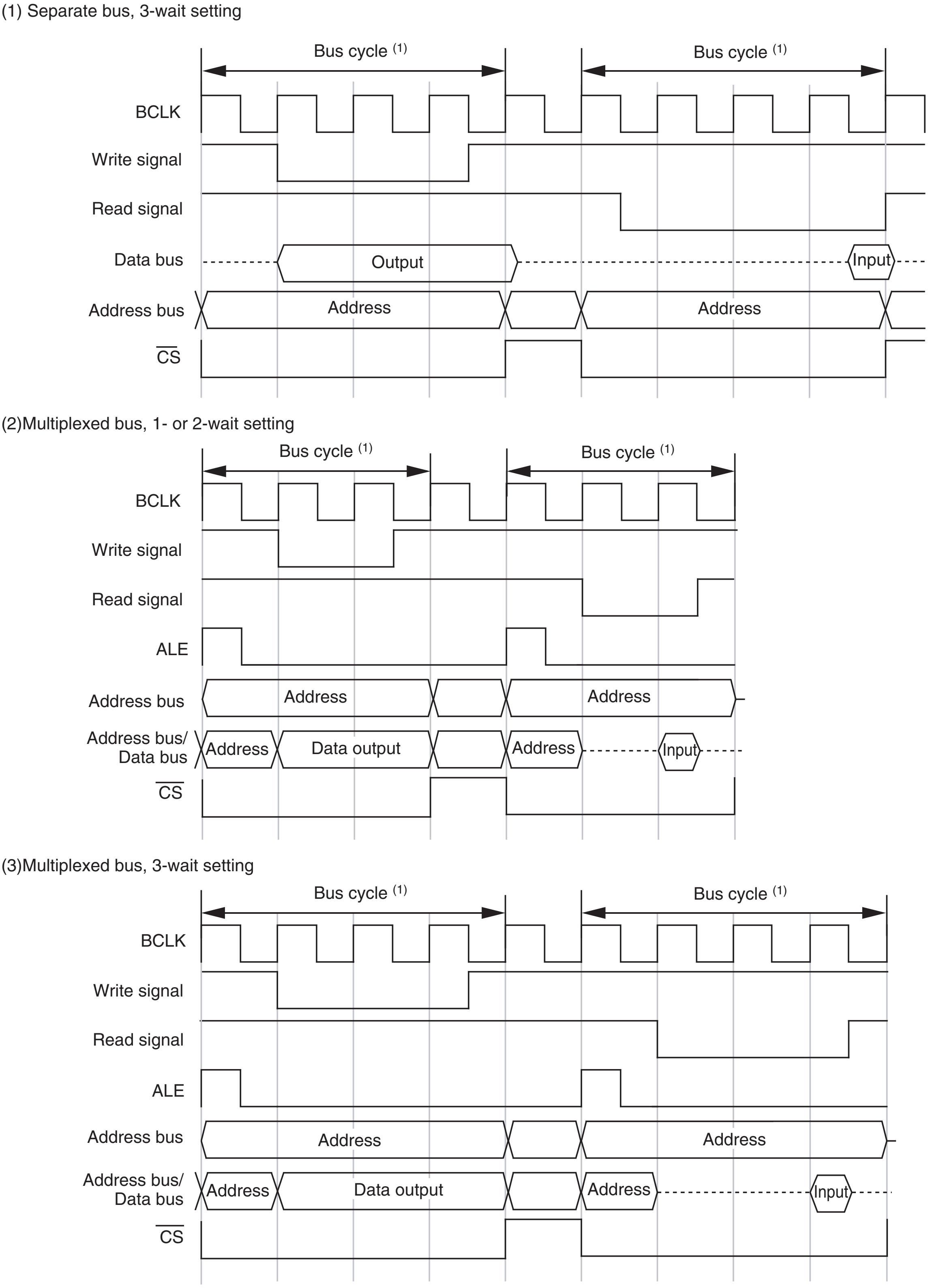
(2) Separate bus, 1-wait setting

(3) Separate bus, 2-wait setting

NOTE:

1. These example timing charts indicate bus cycle length. After this bus cycle sometimes come read and write cycles in succession.

Figure 7.7 Typical Bus Timings Using Software Wait (1)



NOTE:

1. These example timing charts indicate bus cycle length. After this bus cycle sometimes come read and write cycles in succession.

Figure 7.8 Typical Bus Timings Using Software Wait (2)

8. Clock Generation Circuit

8.1 Types of Clock Generation Circuit

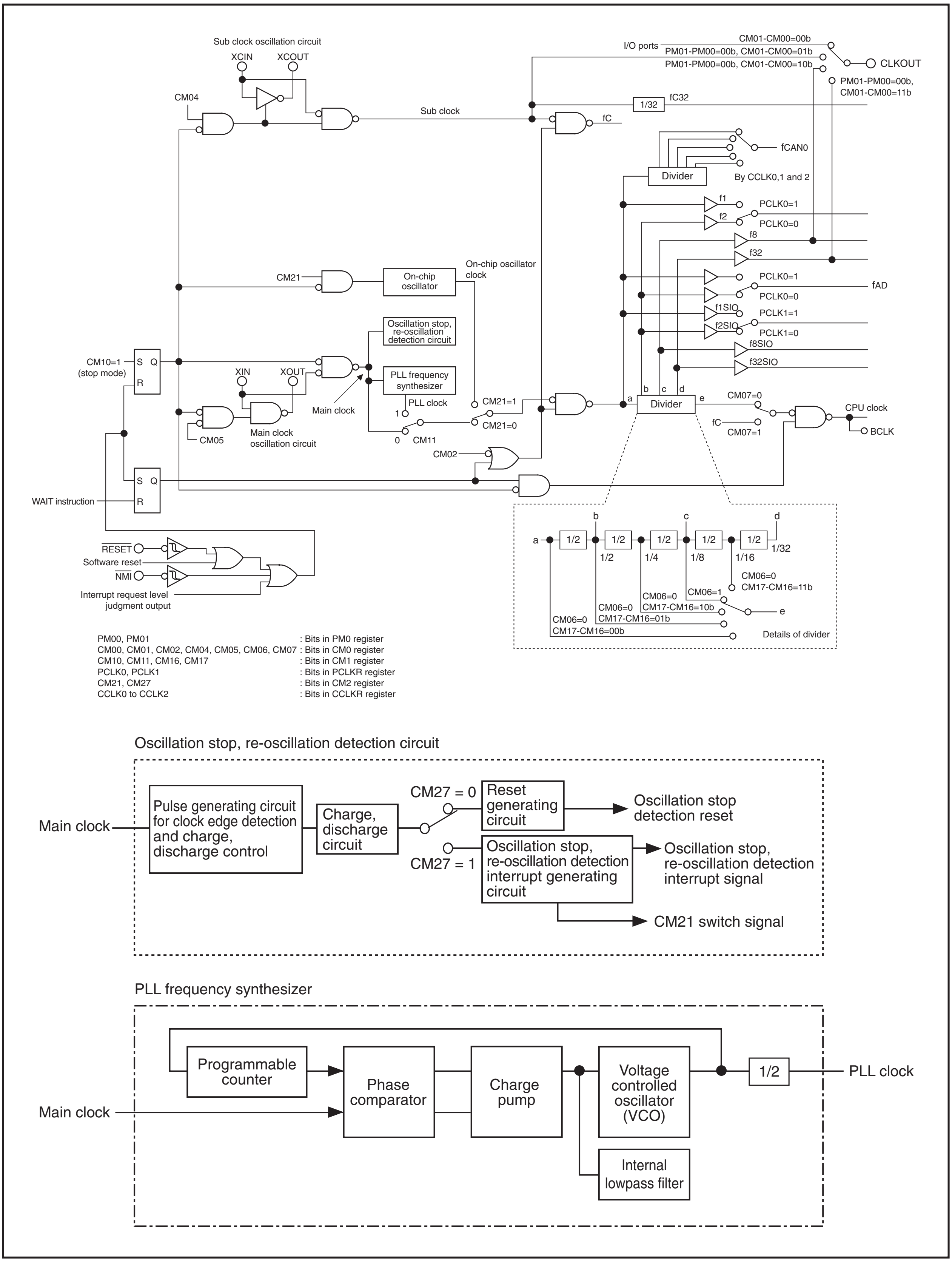
Four circuits are incorporated to generate the system clock signal:

- Main clock oscillation circuit
- Sub clock oscillation circuit
- On-chip oscillator
- PLL frequency synthesizer

Table 8.1 lists the Clock Generation Circuit Specifications. Figure 8.1 shows the Clock Generation Circuit. Figures 8.2 to 8.8 show the clock-related registers.

Table 8.1 Clock Generation Circuit Specifications

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit	On-chip Oscillator	PLL Frequency Synthesizer
Use of clock	• CPU clock source • Peripheral function clock source	• CPU clock source • Clock source of timer A, B	• CPU clock source • Peripheral function clock source • CPU and peripheral function clock sources when the main clock stops oscillating	• CPU clock source • Peripheral function clock source
Clock frequency	0 to 16 MHz	32.768 kHz	About 1 MHz	16 MHz, 20 MHz, 24 MHz
Usable oscillator	•Ceramic oscillator •Crystal oscillator	•Crystal oscillator	-	-
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	-	-
Oscillation stop and re-oscillation detection function	Available	Available	Available	Available
Oscillation status after reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally derived clock can be input		-	-

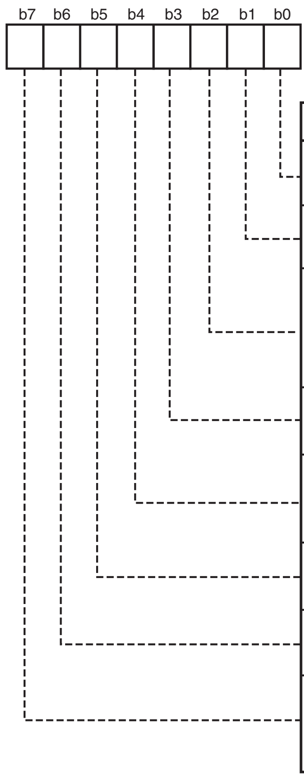


PM00, PM01	: Bits in PM0 register
CM00, CM01, CM02, CM04, CM05, CM06, CM07	: Bits in CM0 register
CM10, CM11, CM16, CM17	: Bits in CM1 register
PCLK0, PCLK1	: Bits in PCLKR register
CM21, CM27	: Bits in CM2 register
CCLK0 to CCLK2	: Bits in CCLKR register

Oscillation stop, re-oscillation detection circuit
Main clock
Pulse generating circuit for clock edge detection and charge, discharge control
Charge, discharge circuit
CM27 = 0
CM27 = 1
Reset generating circuit
Oscillation stop detection reset
Oscillation stop, re-oscillation detection interrupt generating circuit
Oscillation stop, re-oscillation detection interrupt signal
CM21 switch signal
PLL frequency synthesizer
Programmable counter
Phase comparator
Charge pump
Voltage controlled oscillator (VCO)
Internal lowpass filter
1/2
PLL clock
Main clock

Figure 8.1 Clock Generation Circuit

System Clock Control Register 0 (1)



Symbol	Address	After Reset
CM0	0006h	01001000b

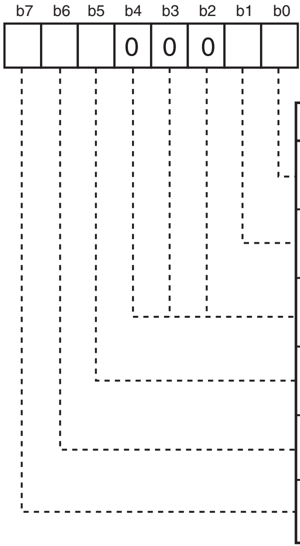
Bit Symbol	Bit Name	Function	RW
CM00	Clock output function select bits (Valid only in single-chip mode)	b1 b0 0 0 : I/O port P5_7 0 1 : Output fC 1 0 : Output f8 1 1 : Output f32	RW
CM01			RW
CM02	WAIT mode peripheral function clock stop bit	0 : Peripheral function clock does not stop in wait mode 1 : Peripheral function clock stops in wait mode (2)	RW
CM03	XCIN-XCOUT drive capacity select bit (3)	0 : LOW 1 : HIGH	RW
CM04	Port XC select bit (3)	0 : I/O port P8_6, P8_7 1 : XCIN-XCOUT oscillation function (4)	RW
CM05	Main clock stop bit (5) (6) (7)	0 : On 1 : Off (8) (9)	RW
CM06	Main clock division select bit 0 (7) (10) (12)	0 : Bits CM16 and CM17 enabled 1 : Divide-by-8 mode	RW
CM07	System clock select bit (6) (11)	0 : Main clock, PLL clock, or on-chip oscillator clock 1 : Sub clock	RW

NOTES:

1. Rewrite this register after setting the PRC0 bit in the PRCR register to 1 (write enabled).
2. The fC32 clock does not stop. In low-speed or low power dissipation mode, do not set this bit to 1 (peripheral clock stops in wait mode).
3. The CM03 bit is set to 1 (high) while the CM04 bit is set to 0 (I/O port) or when entering stop mode.
4. To use a sub clock, set this bit to 1. Also make sure ports P8_6 and P8_7 are directed for input, with no pull-ups.
5. This bit is provided to stop the main clock when the low power dissipation mode or on-chip oscillator low power dissipation mode is selected. This bit cannot be used for detection as to whether the main clock stops or not. To stop the main clock, set bits as follows:
   (a) Set the CM07 bit to 1 (sub clock selected) or the CM21 bit in the CM2 register to 1 (on-chip oscillator selected) with the sub clock stably oscillating.
   (b) Set the CM20 bit in the CM2 register to 0 (oscillation stop, re-oscillation detection function disabled).
   (c) Set the CM05 bit to 1 (stop).
6. To use the main clock as the clock source for the CPU clock, set bits as follows:
   (a) Set the CM05 bit to 0 (oscillate).
   (b) Wait until the main clock oscillation stabilizes.
   (c) Set bits CM11, CM21, and CM07 to 0.
7. When the CM21 bit = 0 (on-chip oscillator stops) and the CM05 bit = 1 (main clock stops), the CM06 bit is fixed to 1 (divide-by-8 mode) and the CM15 bit is fixed to 1 (drive capability high).
8. During external clock input, set the CM05 bit to 0 (oscillate).
9. When the CM05 bit is set to 1, the XOUT pin is held "H". Because the on-chip feedback resistor remains connected, the XIN pin is pulled "H" to the same level as XOUT via the feedback resistor.
10. When entering stop mode from high-speed or medium-speed mode, on-chip oscillator mode or on-chip oscillator low power dissipation mode, the CM06 bit is set to 1 (divide-by-8 mode).
11. After setting the CM04 bit to 1 (XCIN-XCOUT oscillator function), wait until the sub clock oscillates stably before switching the CM07 bit from 0 to 1 (sub clock).
12. To return from on-chip oscillator mode to high-speed or medium-speed mode, set bits CM06 and CM15 to 1.

Figure 8.2 CM0 Register

System Clock Control Register 1 (1)



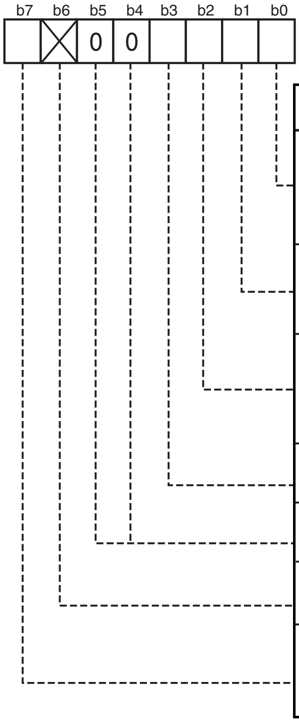
Symbol	Address	After Reset
CM1	0007h	00100000b

Bit Symbol	Bit Name	Function	RW
CM10	All clock stop control bit (2) (3)	0 : Clock on 1 : All clocks off (stop mode)	RW
CM11	System clock select bit 1 (4)	0 : Main clock 1 : PLL clock (5)	RW
- (b4-b2)	Reserved bits	Set to 0	RW
CM15	XIN-XOUT drive capacity select bit (6)	0 : LOW 1 : HIGH	RW
CM16	Main clock division select bits 1 (7)	b7 b6 0 0 : No division mode 0 1 : Divide-by-2 mode 1 0 : Divide-by-4 mode 1 1 : Divide-by-16 mode	RW
CM17			RW

NOTES:
1. Rewrite this register after setting the PRC0 bit in the PRCR register to 1 (write enabled)
2. If the CM10 bit is 1 (stop mode), XOUT is held "H" and the on-chip feedback resistor is disconnected. Pins XCIN and XCOUT are in high-impedance state. When the CM11 bit is set to 1 (PLL clock), or the CM20 bit in the CM2 register is set to 1 (oscillation stop, re-oscillation detection function enabled), do not set the CM10 bit to 1.
3. When the PM22 bit in the PM2 register is set to 1 (on-chip oscillator clock is selected as watchdog timer count source), this bit remains unchanged even if writing to the CM10 bit.
4. This bit is valid when the CM07 bit is 0 and the CM21 bit is 0.
5. After setting the PLC07 bit in the PLC0 register to 1 (PLL operation), wait tsu(PLL) elapses before setting the CM11 bit to 1 (PLL clock).
6. When entering stop mode from high-speed or medium-speed mode, or when the CM05 bit is set to 1 (main clock stops) in low-speed mode, the CM15 bit is set to 1 (drive capability high).
7. This bit is valid when the CM06 bit is 0 (bits CM16 and CM17 enabled).

Figure 8.3 CM1 Register

Oscillation Stop Detection Register (1)



Symbol	Address	After Reset
CM2	000Ch	0X000000b (2)

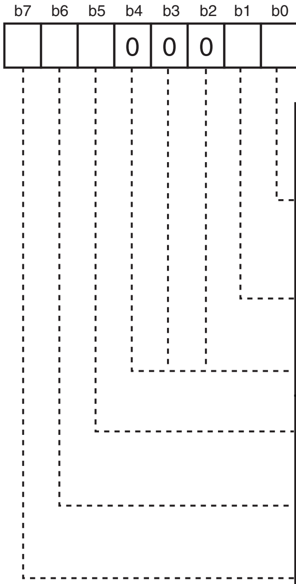
Bit Symbol	Bit Name	Function	RW
CM20	Oscillation stop, re-oscillation detection enable bit (2) (3) (4)	0 : Oscillation stop, re-oscillation detection function disabled 1 : Oscillation stop, re-oscillation detection function enabled	RW
CM21	System clock select bit 2 (2) (5) (6) (7) (8) (11)	0 : Main clock or PLL clock 1 : On-chip oscillator clock (On-chip oscillator oscillates)	RW
CM22	Oscillation stop, re-oscillation detection flag (9)	0 : Main clock stop, re-oscillation not detected 1 : Main clock stop, re-oscillation detected	RW
CM23	XIN monitor flag (10)	0 : Main clock oscillates 1 : Main clock stops	RO
- (b5-b4)	Reserved bits	Set to 0	RW
- (b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
CM27	Operation select bit (when an oscillation stop, re-oscillation is detected) (2)	0 : Oscillation stop detection reset 1 : Oscillation stop, re-oscillation detection interrupt	RW

NOTES:

1. Rewrite this register after setting the PRC0 bit in the PRCR register to 1 (write enabled).
2. Bits CM20, CM21, and CM27 remain unchanged at oscillation stop detection reset.
3. Set the CM20 bit to 0 (disabled) before entering stop mode. Exit stop mode before setting the CM20 bit back to 1 (enabled).
4. Set the CM20 bit to 0 (disabled) before setting the CM05 bit in the CM0 register to 1 (main clock stops).
5. When the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled), the CM27 bit is set to 1 (oscillation stop, re-oscillation detection interrupt), and the CPU clock source is the main clock, the CM21 bit is set to 1 (on-chip oscillator clock) if the main clock stop is detected.
6. If the CM20 bit is set to 1 and the CM23 bit is set to 1 (main clock stops), do not set the CM21 bit to 0.
7. This bit is valid when the CM07 bit in the CM0 register is set to 0.
8. Where the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled), the CM27 bit is set to 1 (oscillation stop, re-oscillation detection interrupt), and the CM11 bit is set to 1 (PLL clock is selected as the CPU clock source), the CM21 bit remains unchanged even if a main clock stop is detected. When the CM22 bit is set to 0 under these conditions, an oscillation stop, re-oscillation detection interrupt request is generated at main clock stop detection. Set the CM21 bit to 1 (on-chip oscillator clock) in the interrupt routine.
9. This bit is set to 1 when the main clock is detected and the main clock re-oscillation is detected. When this bit changes state from 0 to 1, an oscillation stop and re-oscillation detection interrupt request is generated. Use this bit in an interrupt routine to discriminate the interrupt sources between the oscillation stop and re-oscillation detection interrupt and the watchdog timer interrupt. This bit is set to 0 by writing 0 in a program. (This bit remains unchanged even if writing 1. Nor is it set to 0 when an oscillation stop and re-oscillation detection interrupt request is acknowledged.)
   If an oscillation stop or a re-oscillation is detected when the CM22 bit = 1, no oscillation stop and re-oscillation detection interrupt requests are generated.
10. Determine the main clock status by reading the CM23 bit several times in an oscillation stop or re-oscillation detection interrupt routine.
11. When the CM21 bit is set to 0 (on-chip oscillator stops) and the CM05 bit is set to 1 (main clock stops), the CM06 bit is fixed to 1 (divide-by-8 mode) and the CM15 bit is fixed to 1 (drive capability high).

Figure 8.4 CM2 Register

Peripheral Clock Select Register [1]



Symbol	Address	After Reset
PCLKR	025Eh	00h

Bit Symbol	Bit Name	Function	RW
PCLK0	Timers A, B, and A/D clock select bit (Clock source for the timers A, B, the dead time timer and A/D)	0 : Divide-by-2 of fAD, f2 1 : fAD, f1	RW
PCLK1	SI/O clock select bit (Clock source for UART0 to UART2, SI/O3 to SI/O6) [5]	0 : f2SIO 1 : f1SIO	RW
- (b4-b2)	Reserved bits	Set to 0	RW
PCLK5	Pin function switch bit	0: Normal mode 1: Switching mode [4]	RW
PCLK6	Software interrupt number/SFR location switch bit	0: Normal mode 1: Switching mode [2]	RW
PCLK7	A/D clock direct input bit	0: Normal mode 1: Switching mode [3]	RW

NOTES:

1. Rewrite this register after setting the PRC0 bit in the PRCR register to 1 (write enabled)
2. If this bit is set to 1, the software interrupt number and SFR location can be changed as follows.
   (1) Software interrupt number of the key input interrupt in the vector table can be changed from 14 to 13.
   - No.13 is changed from the CAN0 error interrupt to the CAN0 error/key input interrupt.
   - No.14 is changed from the A/D/key input interrupt to the A/D interrupt.

   (2) Address of the KUPIC register in the SFR can be changed from 004Eh to 004Dh.
   - Address 004Dh is changed from the C01ERRIC register to the C01ERRIC/KUPIC register.
   - Address 004Eh is changed from the ADIC/KUPIC register to the ADIC register.
3. When this bit = 1, the A/D clock is set to divide-by-1 of fAD mode regardless of whether the PCLK0 bit is set.
4. When the PCLK5 bit and the SM43 bit in the S4C register = 1, the pin function of SI/O4 can be changed as follows.
   • P8_0/TA4OUT/U/(SIN4)
   • P7_5/TA2IN/$\overline{W}$/(SOUT4)
   • P7_4/TA2OUT/W/(CLK4)
5. SI/O5 and SI/O6 are only in the 128-pin version.

Figure 8.5 PCLKR Register

CAN0 Clock Select Register [1]

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0				

Symbol	Address	After Reset
CCLKR	025Fh	00h

Bit Symbol	Bit Name	Function	RW
CCLK0	CAN0 clock select bits [2]	b2 b1 b0 0 0 0 No division 0 0 1 : Divide-by-2 0 1 0 : Divide-by-4 0 1 1 : Divide-by-8 1 0 0: Divide-by-16 1 0 1 : 1 1 0 : Do not set a value 1 1 1 :	RW
CCLK1			RW
CCLK2			RW
CCLK3	CAN0 CPU interface sleep bit [3]	0: CAN0 CPU interface operating 1: CAN0 CPU interface in sleep	RW
- (b6-b4)	Reserved bit	Set to 0	RW
- (b7)	Reserved bit	Set to 1	RW

NOTES:
1. Rewrite this register after setting the PRC0 bit in the PRCR register to 1 (write enabled).
2. Set to this bit after setting the C1CTLR register to 0020h, and set only when the Reset bit in the C0CTLR register = 1 (reset/Initialization mode).
3. Before setting this bit to 1, set the Sleep bit in the C0CTLR to 1 (sleep mode enabled).

Figure 8.6 CCLKR Register

Processor Mode Register 2 [1]

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	0	0		0	

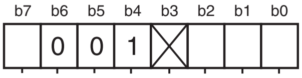
Symbol	Address	After Reset
PM2	001Eh	XXX00000b

Bit Symbol	Bit Name	Function	RW
PM20	Specifying wait when accessing SFR at PLL operation [2]	0 : 2 waits 1 : 1 wait	RW
- (b1)	Reserved bit	Set to 0	RW
PM22	WDT count source protective bit [3] [4]	0 : CPU clock is used for the watchdog timer count source 1 : On-chip oscillator clock is used for the watchdog timer count source	RW
- (b4-b3)	Reserved bits	Set to 0	RW
- (b7-b5)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTES:
1. Rewrite this register after setting the PRC1 bit in the PRCR register to 1 (write enable).
2. The PM20 bit become effective when the PLC07 bit in the PLC0 register is set to 1 (PLL on). Change the PM20 bit when the PLC07 bit is set to 0 (PLL off). Set the PM20 bit to 0 (2 waits) when PLL clock > 16MHz.
3. Once this bit is set to 1, it cannot be set to 0 in a program.
4. Setting the PM22 bit to 1 results in the following conditions:
   - The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.
   - The CM10 bit in the CM1 register is disabled against write. (Writing a 1 has no effect, nor is stop mode entered.)
   - The watchdog timer does not stop when in wait mode or hold state.

Figure 8.7 PM2 Register

PLL Control Register 0 (1)



Symbol	Address	After Reset
PLC0	001Ch	0001X010b

Bit Symbol	Bit Name	Function	RW
PLC00	PLL multiplying factor select bits (2)	b2 b1 b0 0 0 0 : Do not set a value 0 0 1 : Multiply-by-2 0 1 0 : Multiply-by-4 0 1 1 : Multiply-by-6 1 0 0 : 1 0 1 : 1 1 0 : 1 1 1 : } Do not set a value	RW
PLC01			RW
PLC02			RW
- (b3)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
- (b4)	Reserved bit	Set to 1	RW
- (b6-b5)	Reserved bits	Set to 0	RW
PLC07	Operation enable bit (3)	0 : PLL Off 1 : PLL On	RW

NOTES:

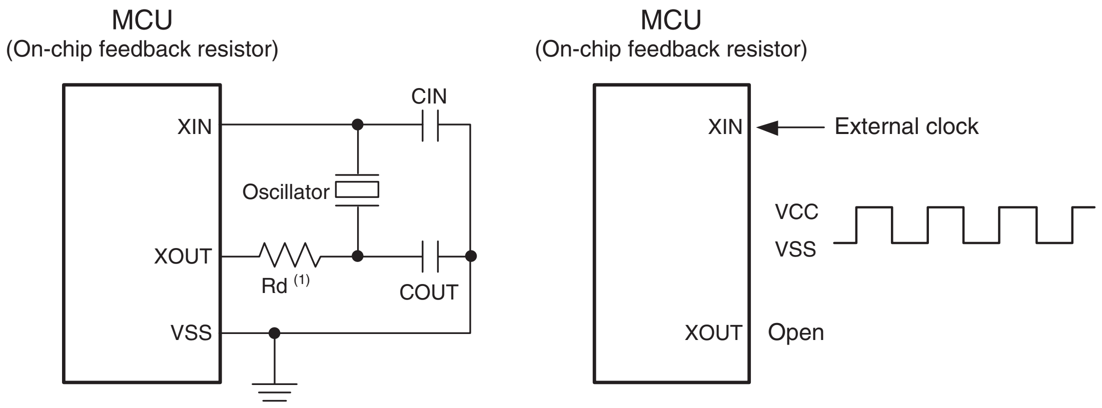
1. Rewrite this register after setting the PRC0 bit in the PRCR register to 1 (write enabled).
2. This bit can only be modified when the PLC07 bit = 0 (PLL turned off). The value once written to this bit cannot be modified.
3. Before setting this bit to 1, set the CM07 bit in the CM0 register to 0 (main clock), set bits CM17 to CM16 in the CM1 register to 00b (main clock undivided mode), and set the CM06 bit in the CM0 register to 0 (bits CM16 and CM17 enabled).

Figure 8.8 PLC0 Register

The following describes the clocks generated by the clock generation circuit.

8.1.1 Main Clock

The main clock is generated by the main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillation circuit is configured by connecting a resonator between pins XIN and XOUT. The main clock oscillation circuit has an on-chip feedback resistor, which is disconnected from the oscillation circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 8.9 shows an Examples of Main Clock Connection Circuit.
After reset, the main clock divided by 8 is selected for the CPU clock.
The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to 1 (main clock oscillation circuit turned off) after switching the clock source for the CPU clock to a sub clock or on-chip oscillator clock. In this case, XOUT goes “H”. Furthermore, because an on-chip feedback resistor remains on, XIN is pulled “H” to XOUT via the feedback resistor. Note, that if an externally generated clock is fed into the XIN pin, the main clock cannot be turned off by setting the CM05 bit to 1, unless the sub clock is selected as a CPU clock. If necessary, use an external circuit to turn off the clock.
During stop mode, all clocks including the main clock are turned off. Refer to **8.4 Power Control**.



NOTE:
1. Place a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by each oscillator the oscillator manufacturer.
When the oscillation drive capacity is set to low, check that oscillation is stable.
Also, place a feedback resistor between XIN and XOUT if the oscillator manufacturer recommends placing the resistor externally.

Figure 8.9 Examples of Main Clock Connection Circuit

8.1.2 Sub Clock

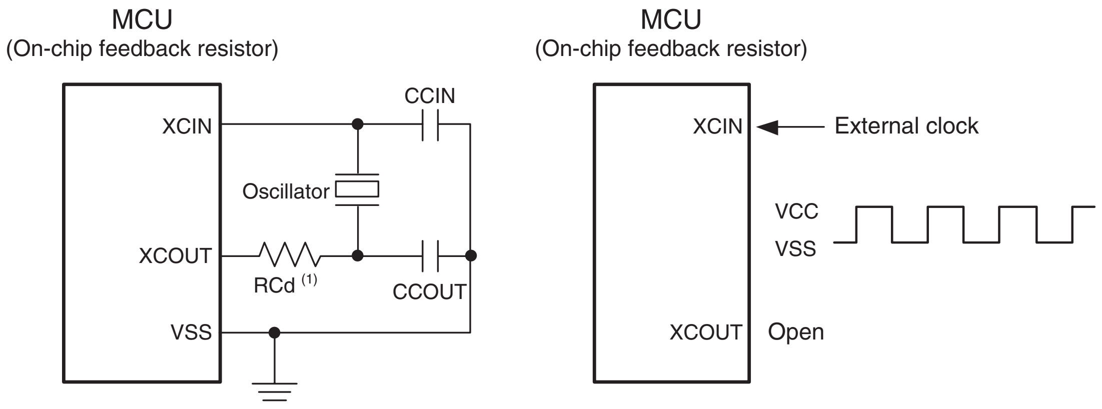
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources. In addition, an fC clock with the same frequency as that of the sub clock can be output from the CLKOUT pin.

The sub clock oscillation circuit is configured by connecting a crystal resonator between pins XCIN and XCOUT. The sub clock oscillation circuit has an on-chip feedback resistor, which is disconnected from the oscillation circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillation circuit may also be configured by feeding an externally generated clock to the XCIN pin. Figure 8.10 shows an Examples of Sub Clock Connection Circuit.

After reset, the sub clock is turned off. At this time, the feedback resistor is disconnected from the oscillation circuit.

To use the sub clock for the CPU clock, set the CM07 bit in the CM0 register to 1 (sub clock) after the sub clock becomes oscillating stably.

During stop mode, all clocks including the sub clock are turned off. Refer to **8.4 Power Control**.



NOTE:
1. Place a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by each oscillator the oscillator manufacturer.
When the oscillation drive capacity is set to low, check that oscillation is stable.
Also, place a feedback resistor between XCIN and XCOUT if the oscillator manufacturer recommends placing the resistor externally.

Figure 8.10 Examples of Sub Clock Connection Circuit

8.1.3 On-chip Oscillator Clock

This clock, approximately 1 MHz, is supplied by a on-chip oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit in the PM2 register is 1 (on-chip oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer (refer to **11.1 Count Source Protective Mode**).

After reset, the on-chip oscillator is turned off. It is turned on by setting the CM21 bit in the CM2 register to 1 (on-chip oscillator clock), and is used as the clock source for the CPU and peripheral function clocks, in place of the main clock. If the main clock stops oscillating when the CM20 bit in the CM2 register is 1 (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is 1 (oscillation stop, re-oscillation detection interrupt), the on-chip oscillator automatically starts operating, supplying the necessary clock for the MCU.

8.1.4 PLL Clock

The PLL clock is generated PLL frequency synthesizer. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the PLL clock is turned off. The PLL frequency synthesizer is activated by setting the PLC07 bit to 1 (PLL operation). When the PLL clock is used as the clock source for the CPU clock, wait tsu(PLL) for the PLL clock to be stable, and then set the CM11 bit in the CM1 register to 1.

Before entering wait mode or stop mode, be sure to set the CM11 bit to 0 (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to 0 (PLL stops). Figure 8.11 shows the Procedure to Use PLL Clock as CPU Clock Source.

The PLL clock frequency is determined by the equation below. When the PLL clock frequency is 16 MHz or more, set the PM20 bit in the PM2 register to 0 (2 waits).

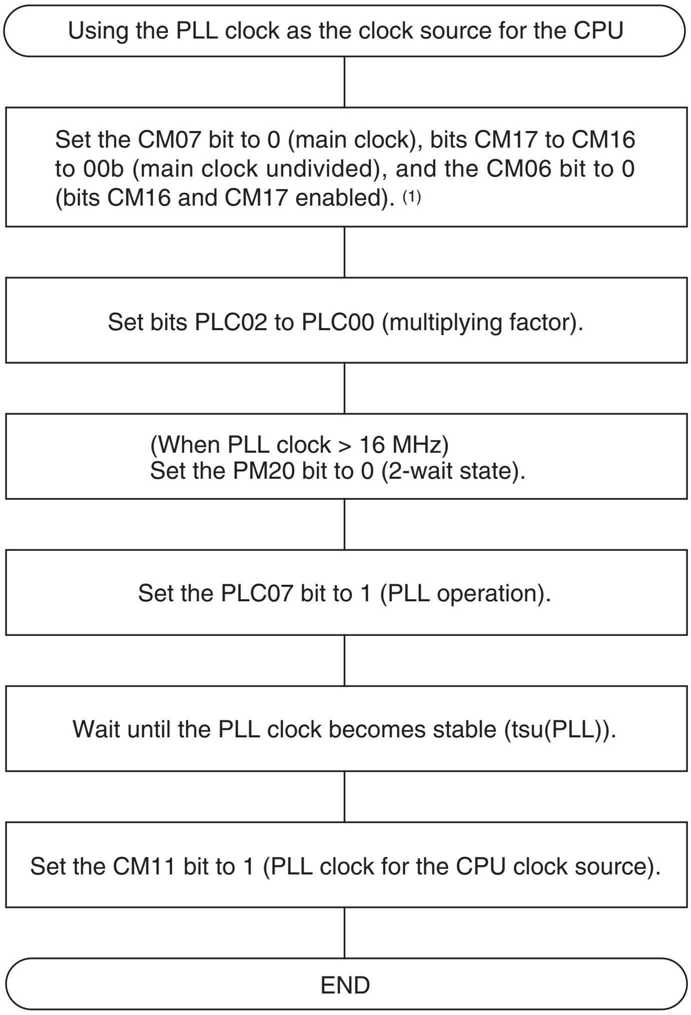
PLL clock frequency = f(XIN) × (multiplying factor set by bits PLC02 to PLC00 in the PLC0 register)
(However, PLL clock frequency = 16 MHz, 20 MHz or 24 MHz)

Bits PLC02 to PLC00 can be set only once after reset. Table 8.2 shows an Example for Setting PLL Clock Frequencies.

Table 8.2 Example for Setting PLL Clock Frequencies

XIN (MHz)	PLC02	PLC01	PLC00	Multiply Factor	PLL Clock (MHz) [1]
8	0	0	1	2	16
4	0	1	0	4	
10	0	0	1	2	20
5	0	1	0	4	
12	0	0	1	2	24
6	0	1	0	4	
4	0	1	1	6	

NOTE:
1. PLL clock frequency = 16 MHz , 20 MHz or 24 MHz


Using the PLL clock as the clock source for the CPU
Set the CM07 bit to 0 (main clock), bits CM17 to CM16 to 00b (main clock undivided), and the CM06 bit to 0 (bits CM16 and CM17 enabled). (1)
Set bits PLC02 to PLC00 (multiplying factor).
(When PLL clock > 16 MHz) Set the PM20 bit to 0 (2-wait state).
Set the PLC07 bit to 1 (PLL operation).
Wait until the PLL clock becomes stable (tsu(PLL)).
Set the CM11 bit to 1 (PLL clock for the CPU clock source).
END

NOTE:

1. PLL operating mode can be entered from high-speed mode.

Figure 8.11 Procedure to Use PLL Clock as CPU Clock Source

8.2 CPU Clock and Peripheral Function Clock

Two type clocks: CPU clock to operate the CPU and peripheral function clocks to operate the peripheral functions.

8.2.1 CPU Clock and BCLK

These are operating clocks for the CPU and watchdog timer.

The clock source for the CPU clock can be chosen to be the main clock, sub clock, on-chip oscillator clock or the PLL clock.

If the main clock or on-chip oscillator clock is selected as the clock source for the CPU clock, the selected clock source can be divided by 1 (undivided), 2, 4, 8, or 16 to produce the CPU clock. Use the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register to select the divide-by-n value.

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to 0 and bits CM17 to CM16 to 00b (undivided).

After reset, the main clock divided by 8 provides the CPU clock.

During memory expansion or microprocessor mode, a BCLK signal with the same frequency as the CPU clock can be output from the BCLK pin by setting the PM07 bit of PM0 register to 0 (output enabled).

Note that when entering stop mode from high-speed or medium-speed mode, on-chip oscillator mode or on-chip oscillator low power dissipation mode, or when the CM05 bit in the CM0 register is set to 1 (main clock turned off) in low-speed mode, the CM06 bit in the CM0 register is set to 1 (divide-by-8 mode).

8.2.2 Peripheral Function Clock (f1, f2, f8, f32, f1SIO, f2SIO, f8SIO, f32SIO, fAD, fCAN0, fC32)

These are operating clocks for the peripheral functions.

Two of these, fi (i = 1, 2, 8, 32) and fiSIO are derived from the main clock, PLL clock or on-chip oscillator clock by dividing them by i. The clock fi is used for timers A and B, and fiSIO is used for serial interface. The f8 and f32 clocks can be output from the CLKOUT pin.

The fAD clock is produced from the main clock, PLL clock or on-chip oscillator clock, and is used for the A/D converter.

The fCAN0 clock is derived from the main clock, PLL clock or on-chip oscillator clock by dividing them by 1 (undivided), 2, 4, 8, or 16, and is used for the CAN module.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to 1 (peripheral function clock turned off during wait mode), or when the MCU is in low power dissipation mode, the fi, fiSIO, fAD, and fCAN0 clocks are turned off [1].

The fC32 clock is produced from the sub clock, and is used for timers A and B. This clock can be used when the sub clock is on.

NOTE:

1. fCAN0 clock stops at “H” in CAN0 sleep mode.

8.3 Clock Output Function

During single-chip mode, the f8, f32, or fC clock can be output from the CLKOUT pin. Use bits CM01 to CM00 in the CM0 register to select.

8.4 Power Control

Normal operating mode, wait mode and stop mode are provided as the power consumption control. All mode states, except wait mode and stop mode, are called normal operating mode in this document.

8.4.1 Normal Operating Mode

Normal operating mode is further classified into seven sub modes.
In normal operating mode, because the CPU clock and the peripheral function clocks both are on, the CPU and the peripheral functions are operating. Power control is exercised by controlling the CPU clock frequency. The higher the CPU clock frequency, the greater the processing capability. The lower the CPU clock frequency, the smaller the power consumption in the chip. If the unnecessary oscillator circuits are turned off, the power consumption is further reduced.
Before the clock sources for the CPU clock can be switched over, the new clock source to which switched must be oscillating stably. If the new clock source is the main clock, sub clock or PLL clock, allow a sufficient wait time in a program until it becomes oscillating stably.
Note that operating modes cannot be changed directly from low speed or low power dissipation mode to on-chip oscillator or on-chip oscillator low power dissipation mode. Nor can operating modes be changed directly from on-chip oscillator or on-chip oscillator low power dissipation mode to low-speed or low power dissipation mode. Where the CPU clock source is changed from the on-chip oscillator to the main clock, change the operating mode to the medium-speed mode (divide-by-8 mode) after the clock was divided by 8 (the CM06 bit in the CM0 register was set to 1) in the on-chip oscillator mode.

8.4.1.1 High-Speed Mode

The main clock divided by 1 provides the CPU clock. If the sub clock is on, fC32 can be used as the count source for timers A and B.

8.4.1.2 PLL Operating Mode

The main clock multiplied by 2, 4, or 6 provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is on, fC32 can be used as the count source for timers A and B. PLL operating mode can be entered from high speed mode. If PLL operating mode is to be changed to wait or stop mode, first go to high speed mode before changing.

8.4.1.3 Medium-Speed Mode

The main clock divided by 2, 4, 8, or 16 provides the CPU clock. If the sub clock is on, fC32 can be used as the count source for timers A and B.

8.4.1.4 Low-Speed Mode

The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit in the CM2 register is set to 0 (on-chip oscillator turned off), and the on-chip oscillator clock is used when the CM21 bit is set to 1 (on-chip oscillator oscillating).
The fC32 clock can be used as the count source for timers A and B.

8.4.1.5 Low Power Dissipation Mode

In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The fC32 clock can be used as the count source for timers A and B.
Simultaneously when this mode is selected, the CM06 bit in the CM0 register becomes 1 (divide-by-8 mode). In the low power dissipation mode, do not change the CM06 bit. Consequently, the medium speed (divide-by-8) mode is to be selected when the main clock is operated next.

8.4.1.6 On-chip Oscillator Mode

The on-chip oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, fC32 can be used as the count source for timers A and B. When the operating mode is returned to the high-speed and medium-speed modes, set the CM06 bit in the CM0 register to 1 (divide-by-8 mode).

8.4.1.7 On-chip Oscillator Low Power Dissipation Mode

The main clock is turned off after being placed in on-chip oscillator mode. The CPU clock can be selected as in on-chip oscillator mode. The on-chip oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, fC32 can be used as the count source for timers A and B.

Table 8.3 lists the Setting Clock Related Bit and Modes.

Table 8.3 Setting Clock Related Bit and Modes

Modes		CM2 Register	CM1 Register		CM0 Register			
		CM21	CM11	CM17, CM16	CM07	CM06	CM05	CM04
PLL operating mode		0	1	00b	0	0	0	-
High-speed mode		0	0	00b	0	0	0	-
Medium-speed mode	Divide-by-2	0	0	01b	0	0	0	-
	Divide-by-4	0	0	10b	0	0	0	-
	Divide-by-8	0	0	-	0	1	0	-
	Divide-by-16	0	0	11b	0	0	0	-
Low-speed mode		-	0	-	1	-	0	1
Low power dissipation mode		0	0	-	1	1 (1)	1 (1)	1
On-chip oscillator mode	No division	1	0	00b	0	0	0	-
	Divide-by-2	1	0	01b	0	0	0	-
	Divide-by-4	1	0	10b	0	0	0	-
	Divide-by-8	1	0	-	0	1	0	-
	Divide-by-16	1	0	11b	0	0	0	-
On-chip oscillator low power dissipation mode		1	0	(NOTE 2)	0	(NOTE 2)	1	-

-: 0 or 1

NOTES:

1. When the CM05 bit is set to 1 (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and the CM06 bit is set to 1 (divide-by-8 mode) simultaneously.
2. The divide-by-n value can be selected the same way as in on-chip oscillator mode.

8.4.2 Wait Mode

In wait mode, the CPU clock is turned off, so are the CPU (because operated by the CPU clock) and the watchdog timer. However, if the PM22 bit in the PM2 register is 1 (on-chip oscillator clock for the watchdog timer count source), the watchdog timer remains active. Because the main clock, sub clock and on-chip oscillator clock all are on, the peripheral functions using these clocks keep operating.

8.4.2.1 Peripheral Function Clock Stop Function

If the CM02 bit in the CM0 register is 1 (peripheral function clocks turned off during wait mode), the f1, f2, f8, f32, f1SIO, f8SIO, f32SIO, fAD, and fCAN0 clocks are turned off when in wait mode, with the power consumption reduced that much. However, fC32 remains on.

8.4.2.2 Entering Wait Mode

The MCU is placed into wait mode by executing the WAIT instruction.

When the CM11 bit = 1 (CPU clock source is the PLL clock), be sure to set the CM11 bit in the CM1 register to 0 (CPU clock source is the main clock) before going to wait mode. The power consumption of the chip can be reduced by setting the PLC07 bit in the PLC0 register to 0 (PLL stops).

8.4.2.3 Pin Status During Wait Mode

Table 8.4 lists the Pin Status During Wait Mode.

Table 8.4 Pin Status During Wait Mode

Pin		Memory Expansion Mode Microprocessor Mode	Single-chip Mode
A0 to A19, D0 to D15, $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS3}}$, $\overline{\mathrm{BHE}}$		Retains status before wait mode	Does not become a bus control pin
$\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, $\overline{\mathrm{WRL}}$, $\overline{\mathrm{WRH}}$		“H”	
$\overline{\mathrm{HLDA}}$, BCLK		“H”	
ALE		“L”	
I/O ports		Retains status before wait mode	Retains status before wait mode
CLKOUT	When fC selected	Does not become a CLKOUT pin	Does not stop
	When f8, f32 selected		•CM02 bit = 0: Does not stop •CM02 bit = 1: Retains status before wait mode

8.4.2.4 Exiting Wait Mode

The MCU exits wait mode by a hardware reset, $\overline{\mathrm{NMI}}$ interrupt or peripheral function interrupt.

If the MCU exits wait mode by a hardware reset or $\overline{\mathrm{NMI}}$ interrupt, set the peripheral function interrupt priority bits ILVL2 to ILVL0 to 000b (interrupt disabled) before executing the WAIT instruction.

The peripheral function interrupts are affected by the CM02 bit. If the CM02 bit is 0 (peripheral function clocks not turned off during wait mode), peripheral function interrupts can be used to exit wait mode. If the CM02 bit is 1 (peripheral function clocks turned off during wait mode), the peripheral functions using the peripheral function clocks stop operating, so that only the peripheral functions clocked by external signals can be used to exit wait mode.

Table 8.5 lists the Interrupts to Exit Wait Mode and Use Conditions.

Table 8.5 Interrupts to Exit Wait Mode and Use Conditions

Interrupt	CM02 Bit = 0	CM02 Bit = 1
$\overline{\text{NMI}}$ interrupt	Can be used	Can be used
Serial interface interrupt	Can be used when operating with internal or external clock	Can be used when operating with external clock
Key input interrupt	Can be used	Can be used
A/D conversion interrupt	Can be used in one-shot mode or single sweep mode	- (Do not use)
Timer A interrupt Timer B interrupt	Can be used in all modes	Can be used in event counter mode or when the count source is fC32
$\overline{\text{INT}}$ interrupt	Can be used	Can be used
CAN0 wake-up interrupt	Can be used in CAN sleep mode	Can be used in CAN sleep mode

If the MCU exits wait mode by a peripheral function interrupt, set up the following before executing the WAIT instruction.

(1) Set bits ILVL2 to ILVL0 in the interrupt control register, for peripheral function interrupts used to exit wait mode.
Bits ILVL2 to ILVL0 in all other interrupt control registers, for peripheral function interrupts not used to exit wait mode, are set to 000b (interrupt disabled).

(2) Set the I flag to 1.

(3) Start operating the peripheral functions used to exit wait mode.
When the peripheral function interrupt is used, an interrupt routine is performed as soon as an interrupt request is acknowledged and the CPU clock is supplied again.

When the MCU exits wait mode by the peripheral function interrupt, the CPU clock is the same clock as the CPU clock executing the WAIT instruction.

8.4.3 Stop Mode

In stop mode, all oscillator circuits are turned off, so are the CPU clock and the peripheral function clocks. Therefore, the CPU and the peripheral functions clocked by these clocks stop operating. The least amount of power is consumed in this mode. If the voltage applied to VCC pin is VRAM or more, the internal RAM is retained.

However, the peripheral functions clocked by external signals keep operating.

Table 8.6 lists the Interrupts to Stop Mode and Use Conditions.

Table 8.6 Interrupts to Stop Mode and Use Conditions

Interrupt	Condition
$\overline{\text{NMI}}$ interrupt	Can be used
Key input interrupt	Can be used
INT interrupt	Can be used
Timer A interrupt Timer B interrupt	Can be used (when counting external pulses in event counter mode)
Serial interface interrupt	Can be used (when external clock is selected)
CAN0 wake-up interrupt	Can be used (when CAN sleep mode is selected)

8.4.3.1 Entering Stop Mode

The MCU is placed into stop mode by setting the CM10 bit in the CM1 register to 1 (all clocks turned off). At the same time, the CM06 bit in the CM0 register is set to 1 (divide-by-8 mode) and the CM15 bit in the CM1 register is set to 1 (main clock oscillator circuit drive capability high).

Before entering stop mode, set the CM20 bit in the CM2 register to 0 (oscillation stop, re-oscillation detection function disabled).

Also, if the CM11 bit in the CM1 register is 1 (PLL clock for the CPU clock source), set the CM11 bit to 0 (main clock for the CPU clock source) and the PLC07 bit in the PLC0 register to 0 (PLL turned off) before entering stop mode.

8.4.3.2 Pin Status in Stop Mode

Table 8.7 lists the Pin Status in Stop Mode.

Table 8.7 Pin Status in Stop Mode

Pin		Memory Expansion Mode Microprocessor Mode	Single-chip Mode
A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BHE}}$		Retains status before stop mode	Does not become a bus control pin
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$		“H”	
$\overline{\text{HLDA}}$, BCLK		“H”	
ALE		undefined	
I/O ports		Retains status before stop mode	Retains status before stop mode
CLKOUT	When fC selected	Does not become a CLKOUT pin	“H”
	When f8, f32 selected		Retains status before stop mode

8.4.3.3 Exiting Stop Mode

Stop mode is exited by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

When the hardware reset or $\overline{\text{NMI}}$ interrupt is used to exit stop mode, set all ILVL2 to ILVL0 bits in the interrupt control registers for the peripheral function interrupt to 000b (interrupt disabled) before setting the CM10 bit in the CM1 register to 1.

When the peripheral function interrupt is used to exit stop mode, set the CM10 bit to 1 after the following settings are completed.

(1) Set bits ILVL2 to ILVL0 in the interrupt control registers to decide the peripheral priority level of the peripheral function interrupt.
Set the interrupt priority levels of the interrupts, not being used to exit stop mode, to 0 by setting the all ILVL2 to ILVL0 bits to 000b (interrupt disabled).

(2) Set the I flag to 1.

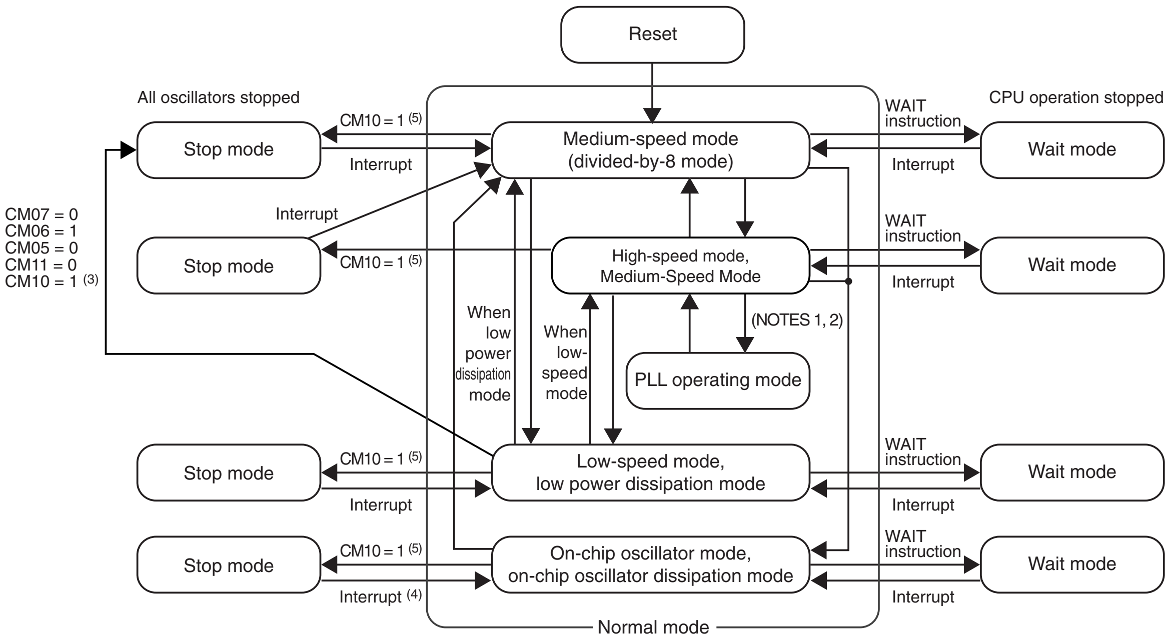
(3) Start operation of peripheral function being used to exit wait mode.
When exiting stop mode by the peripheral function interrupt, the interrupt routine is performed when an interrupt request is generated and the CPU clock is supplied again.

When stop mode is exited by the peripheral function interrupt or $\overline{\text{NMI}}$ interrupt, the CPU clock source is as follows, in accordance with the CPU clock source setting before the MCU had entered stop mode.

- When the sub clock is the CPU clock before entering stop mode: Sub clock
- When the main clock is the CPU clock source before entering stop mode:
  Main clock divided by 8
- When the on-chip oscillator clock is the CPU clock source before entering stop mode:
  On-chip oscillator clock divided by 8

Figure 8.12 shows the State Transition to Stop Mode and Wait Mode. Figure 8.13 shows the State Transition in Normal Operating Mode.

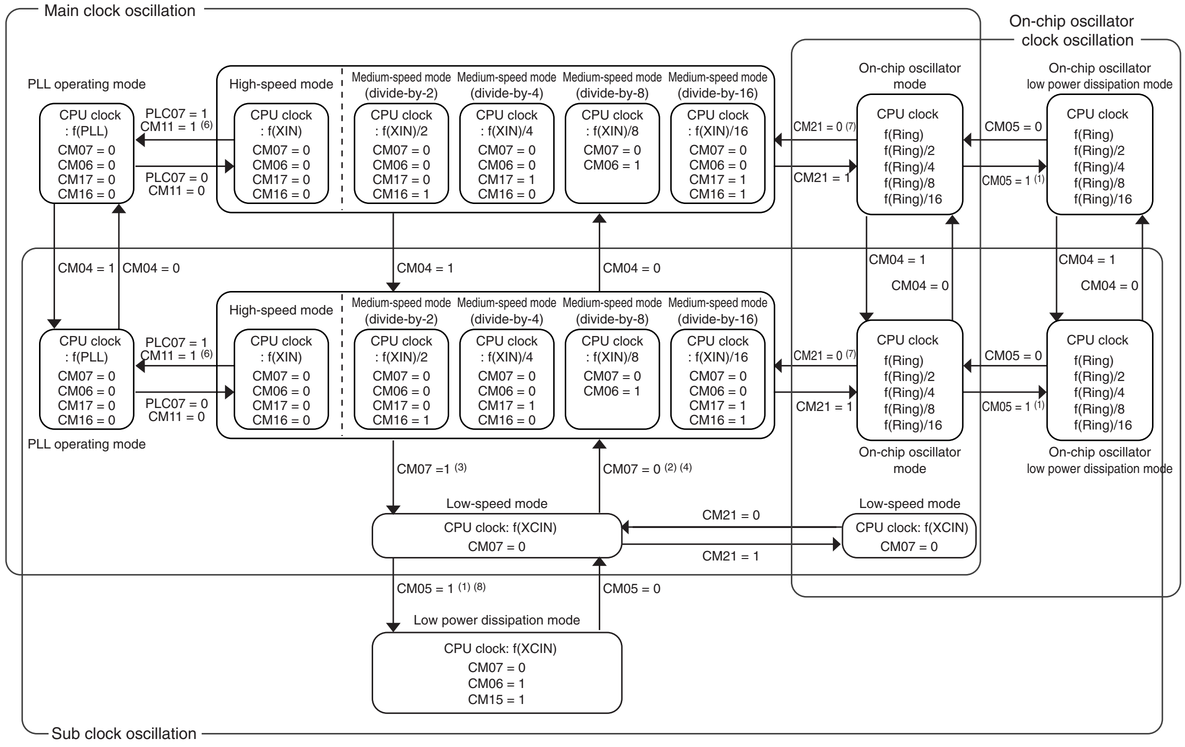
Table 8.8 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line show state after transition.



CM05, CM06, CM07: Bits in CM0 register
CM10, CM11: Bits in CM1 register

NOTES:
1. Do not go directly from PLL operating mode to wait or stop mode.
2. PLL operating mode can be entered from high-speed mode. Similarly, PLL operating mode can be changed back to high-speed mode.
3. Write to registers CM0 and CM1 per 16 bits with the CM21bit in the CM2 register = 0 (on-chip oscillator stops).
Since the operation starts from the main clock after exiting stop mode, the time until the CPU operates can be reduced.
4. The on-chip oscillator clock divided by 8 provides the CPU clock.
5. Before entering stop mode, be sure to set the CM20 bit in the CM2 register to 0 (oscillation stop, re-oscillation detection function disabled).

Figure 8.12 State Transition to Stop Mode and Wait Mode


Main clock oscillation
On-chip oscillator clock oscillation
PLL operating mode
High-speed mode
Medium-speed mode (divide-by-2)
Medium-speed mode (divide-by-4)
Medium-speed mode (divide-by-8)
Medium-speed mode (divide-by-16)
On-chip oscillator mode
On-chip oscillator low power dissipation mode
CPU clock : f(PLL) CM07 = 0 CM06 = 0 CM17 = 0 CM16 = 0
PLC07 = 1 CM11 = 1 (6)
PLC07 = 0 CM11 = 0
CPU clock : f(XIN) CM07 = 0 CM06 = 0 CM17 = 0 CM16 = 0
CPU clock : f(XIN)/2 CM07 = 0 CM06 = 0 CM17 = 0 CM16 = 1
CPU clock : f(XIN)/4 CM07 = 0 CM06 = 0 CM17 = 1 CM16 = 0
CPU clock : f(XIN)/8 CM07 = 0 CM06 = 1
CPU clock : f(XIN)/16 CM07 = 0 CM06 = 0 CM17 = 1 CM16 = 1
CM21 = 0 (7)
CM21 = 1
CPU clock f(Ring) f(Ring)/2 f(Ring)/4 f(Ring)/8 f(Ring)/16
CM05 = 0
CM05 = 1 (1)
CM04 = 1
CM04 = 0
CM07 =1 (3)
CM07 = 0 (2) (4)
Low-speed mode
CPU clock: f(XCIN) CM07 = 0
CM21 = 0
CM21 = 1
CM05 = 1 (1) (8)
CM05 = 0
Low power dissipation mode
CPU clock: f(XCIN) CM07 = 0 CM06 = 1 CM15 = 1
Sub clock oscillation

CM04, CM05, CM06, CM07: Bits in CM0 register
CM11, CM15, CM16, CM17: Bits in CM1 register
CM20, CM21 : Bits in CM2 register
PLC07 : Bit in PLC0 register

NOTES:
1. Avoid making a transition when the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled). Set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) before transiting.
2. Wait the main clock oscillation stabilizes.
3. Switch clock after oscillation of sub clock is sufficiently stable.
4. Change bits CM17 and CM16 before changing the CM06 bit.
5. Transit in accordance with arrow.
6. The PM20 bit in the PM2 register become effective when the PLC07 bit is set to 1 (PLL on). Change the PM20 bit when the PLC07 bit is set to 0 (PLL off). Set the PM20 bit to 0 (2 waits) when PLL clock > 16 MHz.
7. Set the CM06 bit to 1 (divide-by-8 mode) before changing back the operating mode from on-chip oscillator mode to high-speed or middle-speed mode.
8. When the CM21 bit = 0 (on-chip oscillator turned off) and the CM05 bit = 1 (main clock turned off), the CM06 bit is fixed to 1 (divide-by-8 mode) and the CM15 bit is fixed to 1 (drive capability High).

Figure 8.13 State Transition in Normal Operating Mode

Table 8.8 Allowed Transition and Setting (9)

		State after Transition							
		High-Speed Mode, Medium-Speed Mode	Low-Speed Mode (2)	Low Power Dissipation Mode	PLL Operating Mode (2)	On-chip Oscillator Mode	On-chip Oscillator Low Power Dissipation Mode	Stop Mode	Wait Mode
Current State	High-speed mode, medium-speed mode	(NOTE 8)	(9) (7)	–	(13) (3)	(15)	–	(16) (1)	(17)
	Low-speed mode (2)	(8)		(11) (1) (6)	–	–	–	(16) (1)	(17)
	Low power dissipation mode	–	(10)		–	–	–	(16) (1)	(17)
	PLL operating mode (2)	(12) (3)	–	–		–	–	–	–
	On-chip oscillator mode	(14) (4)	–	–	–	(NOTE 8)	(11) (1)	(16) (1)	(17)
	On-chip oscillator low power dissipation mode	–	–	–	–	(10)	(NOTE 8)	(16) (1)	(17)
	Stop mode	(18) (5)	(18)	(18)	–	(18) (5)	(18) (5)		–
	Wait mode	(18)	(18)	(18)	–	(18)	(18)	–	

-: Cannot transit

NOTES:

1. Avoid making a transition when the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled). Set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) before transiting.
2. On-chip oscillator clock oscillates and stops in low-speed mode. In this mode, the on-chip oscillator can be used as peripheral function clock. Sub clock oscillates and stops in PLL operating mode. In this mode, sub clock can be used as peripheral function clock.
3. PLL operating mode can only be entered from and changed to high-speed mode.
4. Set the CM06 bit to 1 (divide-by-8 mode) before transiting from on-chip oscillator mode to high-speed or medium-speed mode.
5. When exiting stop mode, the CM06 bit is set to 1 (divide-by-8 mode).
6. If the CM05 bit is set to 1 (main clock stop), then the CM06 bit is set to 1 (divide-by-8 mode).
7. A transition can be made only when sub clock is oscillating.
8. State transitions within the same mode (divide-by-n values changed or sub clock oscillation turned on or off) are shown in the table below.

		Sub Clock Oscillating					Sub Clock Turned Off				
		No Division	Divide-by-2	Divide-by-4	Divide-by-8	Divide-by-16	No Division	Divide-by-2	Divide-by-4	Divide-by-8	Divide-by-16
Sub Clock Oscillating	No division		(4)	(5)	(7)	(6)	(1)	–	–	–	–
	Divide-by-2	(3)		(5)	(7)	(6)	–	(1)	–	–	–
	Divide-by-4	(3)	(4)		(7)	(6)	–	–	(1)	–	–
	Divide-by-8	(3)	(4)	(5)		(6)	–	–	–	(1)	–
	Divide-by-16	(3)	(4)	(5)	(7)		–	–	–	–	(1)
Sub Clock Turned Off	No division	(2)	–	–	–	–		(4)	(5)	(7)	(6)
	Divide-by-2	–	(2)	–	–	–	(3)		(5)	(7)	(6)
	Divide-by-4	–	–	(2)	–	–	(3)	(4)		(7)	(6)
	Divide-by-8	–	–	–	(2)	–	(3)	(4)	(5)		(6)
	Divide-by-16	–	–	–	–	(2)	(3)	(4)	(5)	(7)	

9. ( ):setting method. See right table.

Setting		Operation
(1)	CM04=0	Sub clock turned off
(2)	CM04=1	Sub clock oscillating

(3)	CM06=0
	CM17=0
	CM16=0
		CPU clock no division mode

(4)	CM06=0
	CM17=0
	CM16=1
		CPU clock divide-by-2 mode

(5)	CM06=0
	CM17=1
	CM16=0
		CPU clock divide-by-4 mode

(6)	CM06=0
	CM17=1
	CM16=1
		CPU clock divide-by-16 mode

(7)	CM06=1	CPU clock divide-by-8 mode
(8)	CM07=0	Main clock, PLL clock or on-chip oscillator clock selected
(9)	CM07=1	Sub clock selected
(10)	CM05=0	Main clock oscillating
(11)	CM05=1	Main clock turned off

(12)	PLC07=0
	CM11=0
		Main clock selected

(13)	PLC07=1
	CM11=1
		PLL clock selected

(14)	CM21=0	Main clock or PLL clock selected
(15)	CM21=1	On-chip oscillator clock selected
(16)	CM10=1	Transition to stop mode
(17)	WAIT instruction	Transition to wait mode
(18)	Hardware interrupt	Exit stop mode or wait mode

CM04, CM05, CM06, CM07	: Bits in CM0 register
CM10, CM11, CM16, CM17	: Bits in CM1 register
CM20, CM21	: Bits in CM2 register
PLC07	: Bit in PLC0 register

8.5 Oscillation Stop and Re-oscillation Detection Function

The oscillation stop and re-oscillation detection function is such that main clock oscillation circuit stop and re-oscillation are detected. At oscillation stop, re-oscillation detection, reset or oscillation stop, re-oscillation detection interrupt request are generated. Which is to be generated can be selected using the CM27 bit in the CM2 register.

The oscillation stop and re-oscillation detection function can be enabled and disabled using the CM20 bit in the CM2 register.

Table 8.9 lists a Specification Overview of Oscillation Stop and Re-oscillation Detection Function.

Table 8.9 Specification Overview of Oscillation Stop and Re-oscillation Detection Function

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	f(XIN) ≥ 2 MHz
Enabling condition for oscillation stop and re-oscillation detection function	Set CM20 bit to 1 (enabled)
Operation at oscillation stop, re-oscillation detection	•Reset occurs (when CM27 bit = 0) •Oscillation stop, re-oscillation detection interrupt is generated (when CM27 bit =1)

8.5.1 Operation when CM27 Bit = 0 (Oscillation Stop Detection Reset)

Where main clock stop is detected when the CM20 bit is 1 (oscillation stop, re-oscillation detection function enabled), the MCU is initialized, coming to a halt (oscillation stop reset; refer to **4. Special Function Registers (SFRs)**, **5. Resets**).

This status is reset with hardware reset. Also, even when re-oscillation is detected, the MCU can be initialized and stopped; it is, however, necessary to avoid such usage (During main clock stop, do not set the CM20 bit to 1 and the CM27 bit to 0).

8.5.2 Operation when CM27 Bit = 1 (Oscillation Stop, Re-oscillation Detection Interrupt)

Where the main clock corresponds to the CPU clock source and the CM20 bit is 1 (oscillation stop, re-oscillation detection function enabled), the system is placed in the following state if the main clock comes to a halt:

- Oscillation stop, re-oscillation detection interrupt request is generated.
- The on-chip oscillator starts oscillation, and the on-chip oscillator clock becomes the clock source for CPU clock and peripheral functions in place of the main clock.
- CM21 bit = 1 (on-chip oscillator clock is the clock source for CPU clock)
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)

Where the PLL clock corresponds to the CPU clock source and the CM20 bit is 1, the system is placed in the following state if the main clock comes to a halt: Since the CM21 bit remains unchanged, set it to 1 (on-chip oscillator clock) inside the interrupt routine.

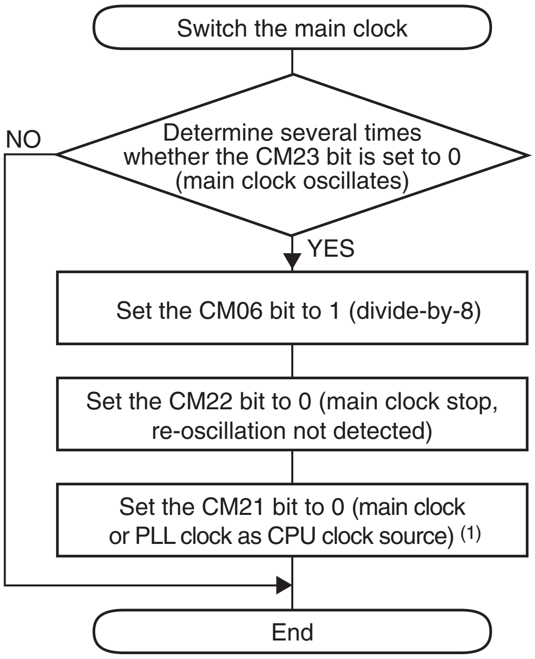
- Oscillation stop, re-oscillation detection interrupt request is generated.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

Where the CM20 bit is 1, the system is placed in the following state if the main clock re-oscillates from the stop condition:

- Oscillation stop, re-oscillation detection interrupt request is generated.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

8.5.3 How to Use Oscillation Stop and Re-oscillation Detection Function

- The oscillation stop, re-oscillation detection interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, the clock source for the CPU clock and peripheral function must be switched to the main clock in the program. Figure 8.14 shows the Procedure to Switch Clock Source from On-chip Oscillator to Main Clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt request occurrence, the CM22 bit becomes 1. When the CM22 bit is set at 1, oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to 0 in the program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is 1, an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the on-chip oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the on-chip oscillator clock.
- To enter wait mode while using the oscillation stop and re-oscillation detection function, set the CM02 bit to 0 (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop and re-oscillation detection function is provided in preparation for main clock stop due to external sources, set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated in the program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to 0.


Switch the main clock
NO
Determine several times whether the CM23 bit is set to 0 (main clock oscillates)
YES
Set the CM06 bit to 1 (divide-by-8)
Set the CM22 bit to 0 (main clock stop, re-oscillation not detected)
Set the CM21 bit to 0 (main clock or PLL clock as CPU clock source) (1)
End

CM06 bit : Bit in CM0 register
Bits CM21, CM22, CM 23: Bits in CM2 register

NOTE:
1. If the clock source for CPU clock is to be changed to PLL clock, set to PLL operating mode after set to high-speed mode.

Figure 8.14 Procedure to Switch Clock Source from On-chip Oscillator to Main Clock

9. Protection

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily.

Figure 9.1 shows the PRCR Register. The registers protected by the PRCR register are listed below.

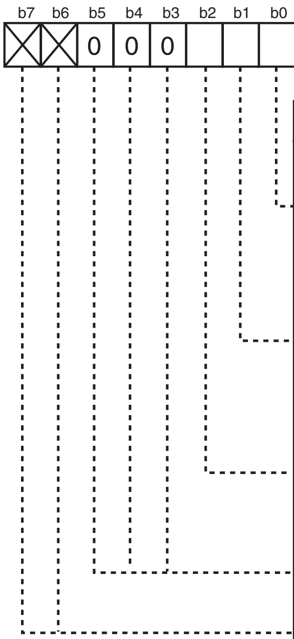
- Registers protected by the PRC0 bit: Registers CM0, CM1, CM2, PLC0, PCLKR, and CCLKR
- Registers protected by the PRC1 bit: Registers PM0, PM1, PM2, TB2SC, INVC0, and INVC1
- Registers protected by the PRC2 bit: Registers PD7, PD9, S3C, S4C, S5C, and S6C [1]

NOTE:

1. Registers S5C and S6C are only in the 128-pin version.

Set the PRC2 bit to 1 (write enabled) and then write to given address, and the PRC2 bit will be set to 0 (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to 1. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to 1 and the next instruction. Bits PRC0 and PRC1 are not automatically set to 0 by writing to given address. They can only be set to 0 in a program.

Protect Register



Symbol	Address	After Reset
PRCR	000Ah	XX000000b

Bit Symbol	Bit Name	Function	RW
PRC0	Protect bit 0	Writing to registers CM0, CM1, CM2, PLC0, PCLKR, CCLKR is enabled 0 : Write protected 1 : Write enabled	RW
PRC1	Protect bit 1	Writing to registers PM0, PM1, PM2, TB2SC, INVC0, INVC1 is enabled 0 : Write protected 1 : Write enabled	RW
PRC2	Protect bit 2	Writing to registers PD7, PD9, S3C, S4C, S5C, S6C is enabled [2] 0 : Write protected 1 : Write enabled [1]	RW
- (b5-b3)	Reserved bits	Set to 0	RW
- (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTES:

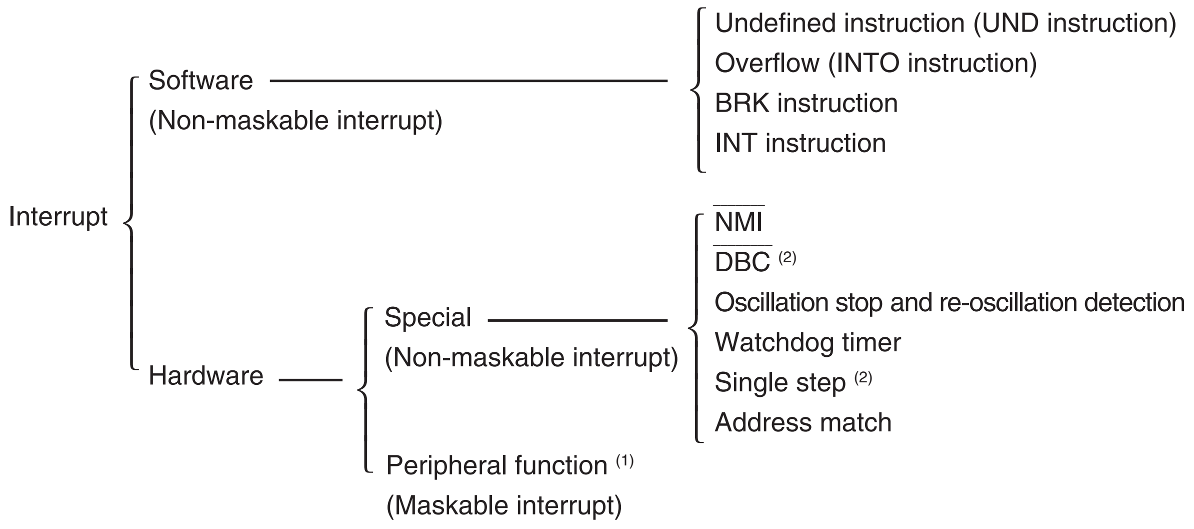
1. The PRC2 bit is set to 0 by writing to given address after setting it to 1. Other bits are not set to 0 by writing to given address, and must therefore be set in a program.
2. Registers S5C and S6C are only in the 128-pin veresion.

Figure 9.1 PRCR Register

10. Interrupts

10.1 Type of Interrupts

Figure 10.1 shows the Types of Interrupts.



NOTES:

1. The peripheral functions in the MCU are used to generate the peripheral interrupt.
2. Do not normally use this interrupt because it is provided exclusively for use by development tools.

Figure 10.1 Types of Interrupts

- Maskable interrupt: An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable interrupt: An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

10.2 Software Interrupts

A software interrupt is generated when executing certain instructions. Software interrupts are non-maskable interrupts.

10.2.1 Undefined Instruction Interrupt

An undefined instruction interrupt is generated when executing the UND instruction.

10.2.2 Overflow Interrupt

An overflow interrupt is generated when executing the INTO instruction with the O flag in the FLG register set to 1 (the operation resulted in an overflow). The following are instructions whose O flag changes by arithmetic: ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

10.2.3 BRK Interrupt

A BRK interrupt is generated when executing the BRK instruction.

10.2.4 INT Instruction Interrupt

An INT instruction interrupt is generated when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 1 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.

In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is set to 0 (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

10.3 Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral function interrupts.

10.3.1 Special Interrupts

Special interrupts are non-maskable interrupts.

10.3.1.1 $\overline{\mathrm{NMI}}$ Interrupt

An $\overline{\mathrm{NMI}}$ interrupt is generated when input on the $\overline{\mathrm{NMI}}$ pin changes state from high to low. For details, refer to **10.7 $\overline{\mathrm{NMI}}$ Interrupt**.

10.3.1.2 $\overline{\mathrm{DBC}}$ Interrupt

Do not normally use this interrupt because it is provided exclusively for use by development tools.

10.3.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to **11. Watchdog Timer**.

10.3.1.4 Oscillation Stop and Re-oscillation Detection Interrupt

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop and re-oscillation detection function, refer to **8. Clock Generation Circuit**.

10.3.1.5 Single-Step Interrupt

Do not normally use this interrupt because it is provided exclusively for use by development tools.

10.3.1.6 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by registers RMAD0 to RMAD3 that corresponds to one of the AIER0 or AIER1 bit in the AIER register or the AIER20 or AIER21 bit in the AIER2 register which is 1 (address match interrupt enabled). For details, refer to **10.10 Address Match Interrupt**.

10.3.2 Peripheral Function Interrupts

The peripheral function interrupt is generated when a request from the peripheral functions in the MCU is acknowledged. The peripheral function interrupt is a maskable interrupt. See **Table 10.2 Relocatable Vector Tables** about how the peripheral function interrupt occurs. Refer to the descriptions of each function for details.

10.4 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. Figure 10.2 shows the Interrupt Vector.

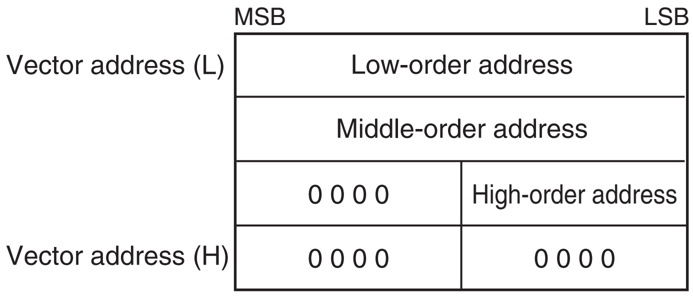


Figure 10.2 Interrupt Vector

10.4.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDCh to FFFFFh. Table 10.1 lists the Fixed Vector Tables. In the flash memory version of MCU, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to **21.2 Functions to Prevent Flash Memory from Rewriting**.

Table 10.1 Fixed Vector Tables

Interrupt Source	Vector table Addresses Address (L) to Address (H)	Reference
Undefined instruction (UND instruction)	FFFDCh to FFFDFh	M16C/60, M16C/20, M16C/Tiny Series Software Manual
Overflow (INTO instruction)	FFFE0h to FFFE3h	
BRK instruction [2]	FFFE4h to FFFE7h	
Address match	FFFE8h to FFFEBh	10.10 Address Match Interrupt
Single step [1]	FFFECh to FFFEFh	-
Oscillation stop and re-oscillation detection, Watchdog timer	FFFF0h to FFFF3h	8. Clock Generation Circuit 11. Watchdog Timer
DBC [1]	FFFF4h to FFFF7h	-
NMI	FFFF8h to FFFFBh	10.7 NMI Interrupt
Reset	FFFFCh to FFFFFh	5. Resets

NOTES:

1. Do not normally use this interrupt because it is provided exclusively for use by development tools.
2. If the contents of address FFFE7h is FFh, program execution starts from the address shown by the vector in the relocatable vector table.

10.4.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Table 10.2 lists the Relocatable Vector Tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

Table 10.2 Relocatable Vector Tables

Interrupt Source	Vector Address (1) Address (L) to Address (H)	Software Interrupt Number	Reference
BRK instruction (2)	+0 to +3 (0000h to 0003h)	0	M16C/60, M16C/20, 16C/Tiny Series Software Manual
CAN0 wake-up (10)	+4 to +7 (0004h to 0007h)	1	19. CAN Module
CAN0 successful reception	+8 to +11 (0008h to 000Bh)	2	
CAN0 successful transmission	+12 to +15 (000Ch to 000Fh)	3	
$\overline{\text{INT3}}$	+16 to +19 (0010h to 0013h)	4	10.6 $\overline{\text{INT}}$ Interrupt
Timer B5, SI/O5 (11)	+20 to +23 (0014h to 0017h)	5	13. Timers
Timer B4, UART1 bus collision detection (3) (9)	+24 to +27 (0018h to 001Bh)	6	15. Serial Interface
Timer B3, UART0 bus collision detection (4) (9)	+28 to +31 (001Ch to 001Fh)	7	
SI/O4, $\overline{\text{INT5}}$ (5)	+32 to +35 (0020h to 0023h)	8	15. Serial Interface
SI/O3, $\overline{\text{INT4}}$ (6)	+36 to +39 (0024h to 0027h)	9	10.6 $\overline{\text{INT}}$ Interrupt
UART2 bus collision detection (9)	+40 to +43 (0028h to 002Bh)	10	15. Serial Interface
DMA0	+44 to +47 (002Ch to 002Fh)	11	12. DMAC
DMA1	+48 to +51 (0030h to 0033h)	12	
CAN0 error (10) (16)	+52 to +55 (0034h to 0037h)	13	19. CAN Module
A/D, Key input (7) (16)	+56 to +59 (0038h to 003Bh)	14	16. A/D Convertor, 10.8 Key Input Interrupt
UART2 transmission, NACK2 (8)	+60 to +63 (003Ch to 003Fh)	15	15. Serial nterface
UART2 reception, ACK2 (8)	+64 to +67 (0040h to 0043h)	16	
UART0 transmission, NACK0 (8)	+68 to +71 (0044h to 0047h)	17	
UART0 reception, ACK0 (8)	+72 to +75 (0048h to 004Bh)	18	
UART1 transmission, NACK1 (8)	+76 to +79 (004Ch to 004Fh)	19	
UART1 reception, ACK1 (8)	+80 to +83 (0050h to 0053h)	20	
Timer A0	+84 to +87 (0054h to 0057h)	21	13. Timers
Timer A1	+88 to +91 (0058h to 005Bh)	22	
Timer A2, $\overline{\text{INT7}}$ (12)	+92 to +95 (005Ch to 005Fh)	23	13. Timers
Timer A3, $\overline{\text{INT6}}$ (13)	+96 to +99 (0060h to 0063h)	24	10.6 $\overline{\text{INT}}$ Interrupt
Timer A4	+100 to +103 (0064h to 0067h)	25	13. Timers
Timer B0, SI/O6 (14)	+104 to +107 (0068h to 006Bh)	26	13. Timers, 15. Serial Interface
Timer B1, $\overline{\text{INT8}}$ (15)	+108 to +111 (006Ch to 006Fh)	27	13. Timers, 10.6 $\overline{\text{INT}}$ Interrupt
Timer B2	+112 to +115 (0070h to 0073h)	28	13. Timers
$\overline{\text{INT0}}$	+116 to +119 (0074h to 0077h)	29	10.6 $\overline{\text{INT}}$ Interrupt
$\overline{\text{INT1}}$	+120 to +123 (0078h to 007Bh)	30	
$\overline{\text{INT2}}$	+124 to +127 (007Ch to 007Fh)	31	
INT instruction interrupt (2)	+128 to +131 (0080h to 0083h) to +252 to + 255 (00FCh to 00FFh)	32 to 63	M16C/60, M16C/20, 16C/Tiny Series Software Manual

NOTES:

1. Address relative to address in INTB.
2. These interrupts cannot be disabled using the I flag.
3. Use the IFSR07 bit in the IFSR0 register to select.
4. Use the IFSR06 bit in the IFSR0 register to select.
5. Use the IFSR17 bit in the IFSR1 register to select.
   When using SI/O4, set the IFSR03 bit in the IFSR0 register to 1 (SI/O4) simultaneously.
6. Use the IFSR16 bit in the IFSR1 register to select.
   When using SI/O3, set the IFSR00 bit in the IFSR0 register to 1 (SI/O3) simultaneously.
7. Use the IFSR01 bit in the IFSR0 register to select.
8. During I^{2}C mode, NACK and ACK interrupts comprise the interrupt source.
9. Bus collision detection: During IE mode, this bus collision detection constitutes the interrupt source.
   During I^{2}C mode, a start condition or a stop condition detection constitutes the interrupt source.
10. Use the IFSR02 bit in the IFSR0 register to select. When the IFSR02 bit = 0, CAN0/1 wake-up is selected. When the IFSR02 bit = 1, CAN0 wake-up/error is selected.
11. Use the IFSR04 bit in the IFSR0 register to select.
    SI/O5 is only in the 128-pin version. In the 100-pin version, set the IFSR04 bit to 0 (Timer B5).
12. Use the IFSR20 bit in the IFSR2 register to select.
    $\overline{\text{INT7}}$ is only in the 128-pin version. In the 100-pin version, set the IFSR20 bit to 0 (Timer A2).
13. Use the IFSR21 bit in the IFSR2 register to select.
    $\overline{\text{INT6}}$ is only in the 128-pin version. In the 100-pin version, set the IFSR21 bit to 0 (Timer A3).
14. Use the IFSR05 bit in the IFSR0 register to select.
    SI/O6 is only in the 128-pin version. In the 100-pin version, set the IFSR05 bit to 0 (Timer B0).
15. Use the IFSR22 bit in the IFSR2 register to select.
    $\overline{\text{INT8}}$ is only in the 128-pin version. In the 100-pin version, set the IFSR22 bit to 0 (Timer B1).
16. If the PCLK6 bit in the PCLKR register is set to 1, software interrupt number 13 can be changed to CAN0 error or key input interrupt, and software interrupt number 14 can be changed to A/D interrupt. (The software interrupt number of key input is changed from 14 to 13) Use the IFSR26 bit in the IFSR2 register to select when selecting CAN0 error or key input.

10.5 Interrupt Control

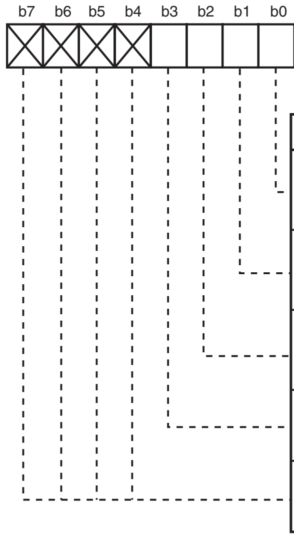
The following describes how to enable/disable the maskable interrupts, and how to set the priority in which order they are accepted. What is explained here does not apply to non-maskable interrupts.

Use the I flag in the FLG register, IPL, and bits ILVL2 to ILVL0 in the each interrupt control register to enable/disable the maskable interrupts. Whether an interrupt is requested is indicated by the IR bit in the each interrupt control register.

Figures 10.3 and 10.4 show the Interrupt Control Registers.

Interrupt Control Register [1]

Symbol	Address	After Reset
C01WKIC	0041h	XXXXX000b
C0RECIC	0042h	XXXXX000b
C0TRMIC	0043h	XXXXX000b
TB5IC/S5IC [5]	0045h	XXXXX000b
TB4IC/U1BCNIC [2]	0046h	XXXXX000b
TB3IC/U0BCNIC [3]	0047h	XXXXX000b
U2BCNIC	004Ah	XXXXX000b
DM0IC, DM1IC	004Bh, 004Ch	XXXXX000b
C01ERRIC [6]	004Dh	XXXXX000b
ADIC/KUPIC [6]	004Eh	XXXXX000b
S0TIC to S2TIC	0051h, 0053h, 004Fh	XXXXX000b
S0RIC to S2RIC	0052h, 0054h, 0050h	XXXXX000b
TA0IC, TA1IC	0055h, 0056h	XXXXX000b
TA4IC	0059h	XXXXX000b
TB0IC/S6IC [7]	005Ah	XXXXX000b
TB2IC	005Ch	XXXXX000b



Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW [4]
- (b7-b4)	Noting is assigned. If necessary, set to 0. When read, the content is undefined.		-

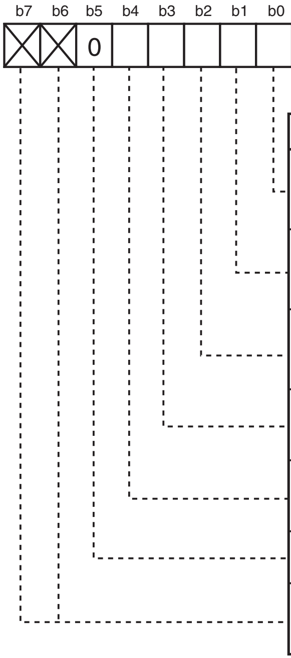
NOTES:

1. To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, refer to **23.8 Interrupt**.
2. Use the IFSR07 bit in the IFSR0 register to select.
3. Use the IFSR06 bit in the IFSR0 register to select.
4. This bit can only be reset by writing 0 (do not write 1).
5. Use the IFSR04 bit in the IFSR0 register to select.
   The S5IC register is only in the 128-pin version. In the 100-pin version, set the IFSR04 bit to 0 (timer B5).
6. If the PCLK6 bit in the PCLKR register is set to 1, C01ERRIC/KUPIC register can be assigned in an address 004Dh, and the ADIC register can be assigned in an address 004Eh. (SFR location of the KUPIC register is changed from address 004Eh to address 004Dh.)
7. Use the IFSR05 bit in the IFSR0 register to select.
   The S6IC register is only in the 128-pin version. In the 100-pin version, set the IFSR05 bit to 0 (timer B0).

Figure 10.3 Interrupt Control Registers (1)

Interrupt Control Register [1]

Symbol	Address	After Reset
INT3IC [2]	0044h	XX00X000b
S4IC/INT5IC [2] [7]	0048h	XX00X000b
S3IC/INT4IC [2] [8]	0049h	XX00X000b
INT0IC to INT2IC	005Dh to 005Fh	XX00X000b
TA2IC/INT7IC [9]	0057h	XX00X000b
TA3IC/INT6IC [10]	0058h	XX00X000b
TB1IC/INT8IC [11]	005Bh	XX00X000b



Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW [3]
POL	Polarity select bit	0 : Selects falling edge [4] [5] [6] 1 : Selects rising edge	RW
- (b5)	Reserved bit	Set to 0	RW
- (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		-

NOTES:

1. To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, refer to **23.8 Interrupt**.
2. When the BYTE pin is low and the processor mode is memory expansion or microprocessor mode, set the ILVL2 to ILVL0 bits in the INT5IC to INT3IC registers to 000b (interrupt disabled).
3. This bit can only be reset by writing 0 (do not write 1).
4. If bits IFSR10 to IFSR15 in the IFSR1 register and bits IFSR23 to IFSR25 in the IFSR2 register are 1 (both edges), set the POL bit in registers INT0IC to INT8IC to 0 (falling edge). Registers INT6IC to INT8IC are in the 128-pin version.
5. Set the POL bit in the S3IC register to 0 (falling edge) when the IFSR00 bit in the IFSR0 register = 1 and the IFSR16 bit in the IFSR1 register = 0 (SI/O3 selected).
6. Set the POL bit in the S4IC register to 0 (falling edge) when the IFSR03 bit in the IFSR0 register = 1 and the IFSR17 bit in the IFSR1 register = 0 (SI/O4 selected).
7. Use the IFSR17 bit in the IFSR1 register to select.
8. Use the IFSR16 bit in the IFSR1 register to select.
9. Use the IFSR20 bit in the IFSR2 register to select.
   The INT7IC register is only in the 128-pin version. In the 100-pin version, set the IFSR20 bit to 0 (timer A2).
10. Use the IFSR21 bit in the IFSR2 register to select.
    The INT6IC register is only in the 128-pin version. In the 100-pin version, set the IFSR21 bit to 0 (timer A3).
11. Use the IFSR22 bit in the IFSR2 register to select.
    The INT8IC register is only in the 128-pin version. In the 100-pin version, set the IFSR22 bit to 0 (timer B1).

Figure 10.4 Interrupt Control Registers (2)

10.5.1 I Flag

The I flag enables or disables the maskable interrupt. Setting the I flag to 1 (enabled) enables the maskable interrupt. Setting the I flag to 0 (disabled) disables all maskable interrupts.

10.5.2 IR Bit

The IR bit is set to 1 (interrupt requested) when an interrupt request is generated. Then, when the interrupt request is accepted and the CPU branches to the corresponding interrupt vector, the IR bit is set to 0 (interrupt not requested).
The IR bit can be set to 0 in a program. Note that do not write 1 to this bit.

10.5.3 Bits ILVL2 to ILVL0 and IPL

Interrupt priority levels can be set using bits ILVL2 to ILVL0.
Table 10.3 shows the settings of interrupt priority levels and Table 10.4 shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:

· I flag = 1
· IR bit = 1
· interrupt priority level > IPL

The I flag, IR bit, bits ILVL2 to ILVL0 and IPL are independent of each other. In no case do they affect one another.

Table 10.3 Settings of Interrupt Priority Levels

Bits ILVL2 to ILVL0	Interrupt Priority Level	Priority Order
000b	Level 0 (Interrupt disabled)	-
001b	Level 1	Low
010b	Level 2	↓
011b	Level 3	
100b	Level 4	
101b	Level 5	
110b	Level 6	
111b	Level 7	High

Table 10.4 Interrupt Priority Levels Enabled by IPL

IPL	Enabled Interrupt Priority Levels
000b	Interrupt levels 1 and above are enabled
001b	Interrupt levels 2 and above are enabled
010b	Interrupt levels 3 and above are enabled
011b	Interrupt levels 5 and above are enabled
100b	Interrupt levels 5 and above are enabled
101b	Interrupt levels 6 and above are enabled
110b	Interrupt levels 7 and above are enabled
111b	All maskable interrupts are disabled

10.5.4 Interrupt Sequence

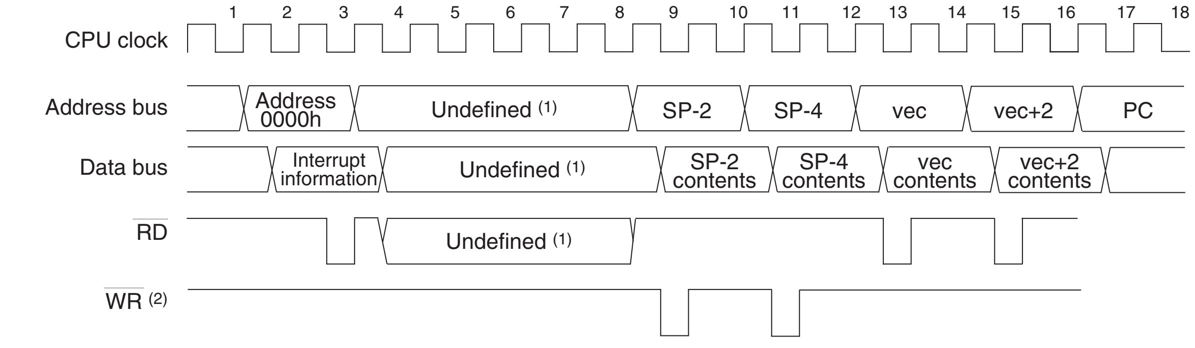
An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.
If an interrupt request is generated while an instruction is being executing, the CPU determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. However, for the SMOVB, SMOVF, SSTR or RMPA instruction, if an interrupt request is generated while the instruction is being executing, the MCU temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.
The CPU behavior during the interrupt sequence is described below.
Figure 10.5 shows the Time Required for Executing Interrupt Sequence.

(1) The CPU obtains interrupt information (interrupt number and interrupt request level) by reading address 000000h. Then, the IR bit applicable to the interrupt information is set to 0 (interrupt requested).
(2) The FLG register, prior to an interrupt sequence, is saved to a temporary register [(1)] within the CPU.
(3) Flags I, D, and U in the FLG register become as follows:
- The I flag is set to 0 (interrupt disabled)
- The D flag is set to 0 (single-step interrupt disabled)
- The U flag is set to 0 (ISP selected)

However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
(4) The temporary register [(1)] within the CPU is saved to the stack.
(5) The PC is saved to the stack.
(6) The interrupt priority level of the acknowledged interrupt in IPL is set.
(7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

After the interrupt sequence is completed, an instruction is executed from the starting address of the interrupt routine.

NOTE:
1. This register cannot be accessed by user.

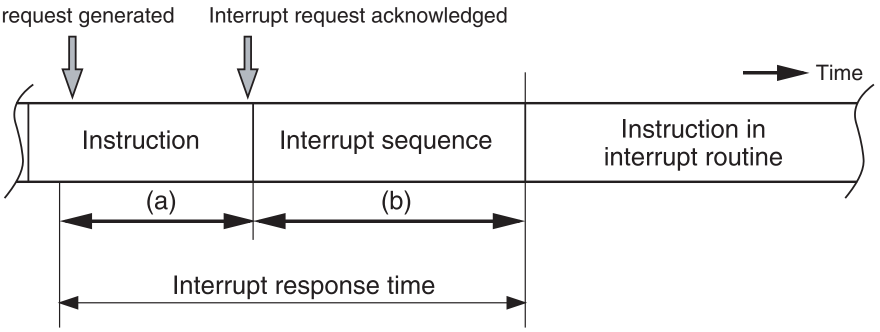


NOTES:
1. The undefined state depends on the instruction queue buffer.
A read cycle occurs when the instruction queue buffer is ready to accept instructions.
2. The WR signal timing shown here is for the case where the stack is located in the internal RAM.

Figure 10.5 Time Required for Executing Interrupt Sequence

10.5.5 Interrupt Response Time

Figure 10.6 shows the Interrupt Response Time. The interrupt response or interrupt acknowledge time denotes a time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of a time from when an interrupt request is generated till when the instruction then executing is completed ((a) on Figure 10.6) and a time during which the interrupt sequence is executed ((b) on Figure 10.6).


Interrupt request generated
Interrupt request acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
(b)
Interrupt response time

(a) A time from when an interrupt request is generated till when the instruction then executing is completed. The length of this time varies with the instruction being executed. The DIVX instruction requires the longest time, which is equal to 30 cycles (without wait state, the divisor being a register).

(b) A time during which the interrupt sequence is executed. For details, see the table below. Note, however, that the values in this table must be increased 2 cycles for the DBC interrupt and 1 cycle for the address match and single-step interrupts.

Interrupt Vector Address	SP Value	16-bit Bus, without Wait	8-bit Bus, without Wait
Even	Even	18 cycles	20 cycles
	Odd	19 cycles	
Odd	Even	19 cycles	
	Odd	20 cycles	

Figure 10.6 Interrupt Response Time

10.5.6 Variation of IPL when Interrupt Request is Accepted

When a maskable interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in Table 10.5 is set in the IPL. Table 10.5 shows the IPL Level that is Set to IPL when Software or Special Interrupts is Accepted.

Table 10.5 IPL Level that is Set to IPL when Software or Special Interrupt is Accepted

Interrupt Sources	Value that is Set to IPL
Oscillation stop and re-oscillation detection, Watchdog timer, NMI	7
Software, Address match, DBC, Single-step	Not changed

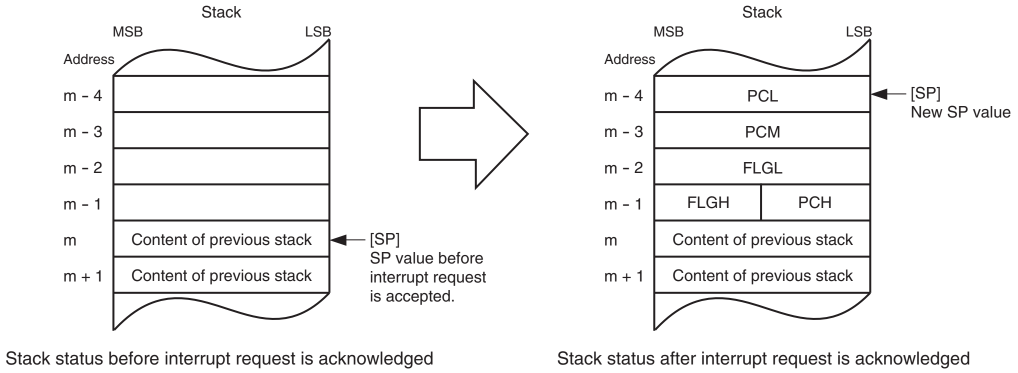
10.5.7 Saving Registers

In the interrupt sequence, the FLG register and PC are saved to the stack.

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits in the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved.

Figure 10.7 shows the Stack Status Before and After Acceptance of Interrupt Request.

The other necessary registers must be saved in a program at the beginning of the interrupt routine. Use the PUSHM instruction, and all registers except SP can be saved with a single instruction.


Stack
MSB
LSB
Address
m - 4
m - 3
m - 2
m - 1
m
m + 1
Content of previous stack
Content of previous stack
[SP]
SP value before interrupt request is accepted.
Stack
MSB
LSB
Address
m - 4
m - 3
m - 2
m - 1
m
m + 1
PCL
PCM
FLGL
FLGH
PCH
Content of previous stack
Content of previous stack
[SP]
New SP value
Stack status before interrupt request is acknowledged
Stack status after interrupt request is acknowledged

PCL : 8 low-order bits of PC
PCM : 8 middle-order bits of PC
PCH : 4 high-order bits of PC
FLGL : 8 low-order bits of FLG
FLGH: 4 high-order bits of FLG

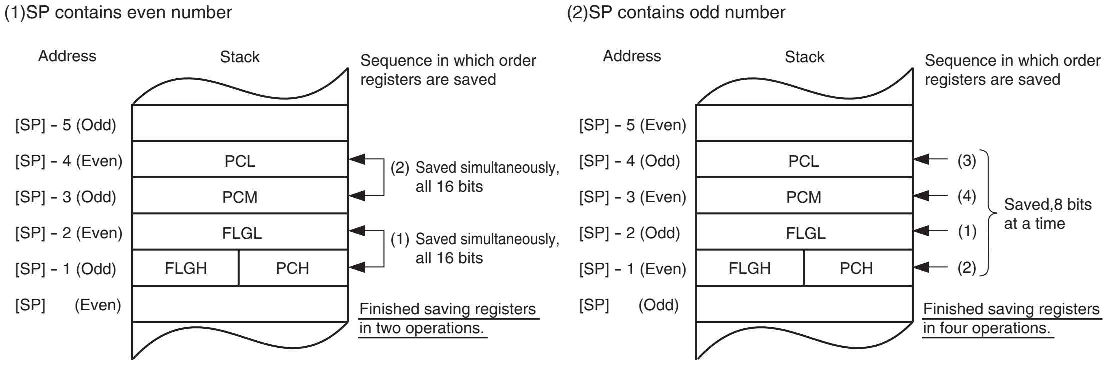
Figure 10.7 Stack Status Before and After Acceptance of Interrupt Request

The register saving operation carried out in the interrupt sequence is dependent on whether the SP (1), at the time of acceptance of an interrupt request, is even or odd. If the SP (1) is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time.

Figure 10.8 shows the Register Saving Operation.

NOTE:

1. When any INT instruction in software numbers 32 to 63 has been executed, this is the SP indicated by the U flag. Otherwise, it is the ISP.


(1)SP contains even number
Address
Stack
Sequence in which order registers are saved
[SP] - 5 (Odd)
[SP] - 4 (Even)
[SP] - 3 (Odd)
[SP] - 2 (Even)
[SP] - 1 (Odd)
[SP] (Even)
PCL
PCM
FLGL
FLGH
PCH
(2) Saved simultaneously, all 16 bits
(1) Saved simultaneously, all 16 bits
Finished saving registers in two operations.
(2)SP contains odd number
Address
Stack
Sequence in which order registers are saved
[SP] - 5 (Even)
[SP] - 4 (Odd)
[SP] - 3 (Even)
[SP] - 2 (Odd)
[SP] - 1 (Even)
[SP] (Odd)
PCL
PCM
FLGL
FLGH
PCH
(3)
(4)
(1)
(2)
Saved,8 bits at a time
Finished saving registers in four operations.

PCL : 8 low-order bits of PC
PCM : 8 middle-order bits of PC
PCH : 4 high-order bits of PC
FLGL : 8 low-order bits of FLG
FLGH: 4 high-order bits of FLG

NOTE:

1. [SP] denotes the initial value of the SP when interrupt request is acknowledged. After registers are saved, the SP content is [SP] minus 4.

Figure 10.8 Register Saving Operation

10.5.8 Returning from Interrupt Routine

The FLG register and PC in the state in which they were immediately before entering the interrupt sequence are restored from the stack by executing the REIT instruction at the end of the interrupt routine. Thereafter the CPU returns to the program which was being executed before accepting the interrupt request.

Return the other registers saved by a program within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

Register bank is switched back to the bank used prior to the interrupt sequence by the REIT instruction.

10.5.9 Interrupt Priority

If two or more interrupt requests are sampled at the same sampling points (a timing to detect whether an interrupt request is generated or not), the interrupt request with the highest priority is acknowledged.

For maskable interrupts (peripheral functions interrupt), any desired priority level can be selected using bits ILVL2 to ILVL0. However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, with the highest priority interrupt accepted.

The watchdog timer and other special interrupts have their priority levels set in hardware.

Figure 10.9 shows the Hardware Interrupts Priority.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

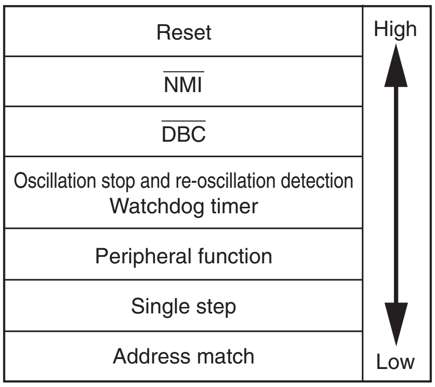
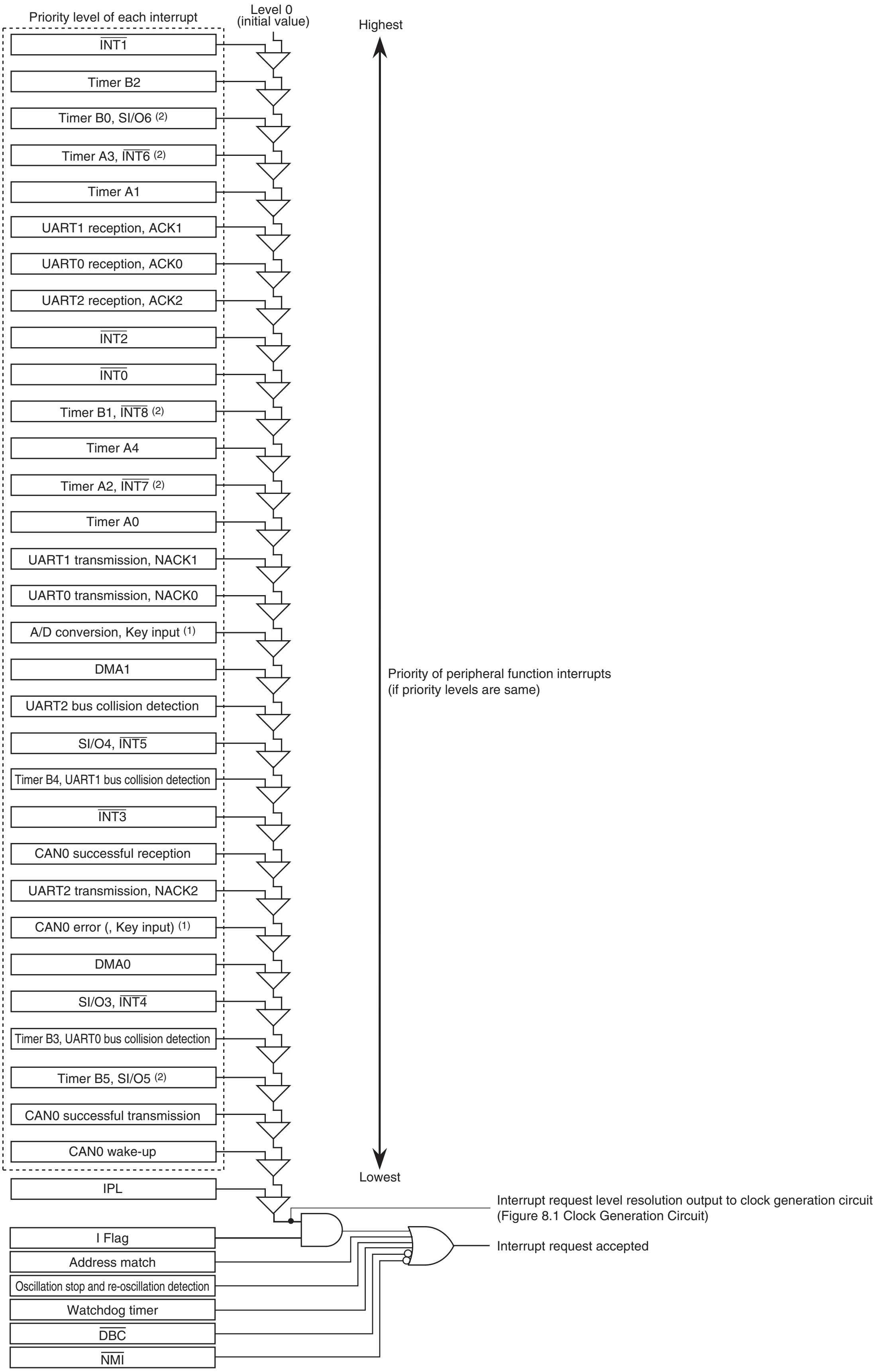


Figure 10.9 Hardware Interrupt Priority

10.5.10 Interrupt Priority Level Select Circuit

The interrupt priority level select circuit selects the highest priority interrupt when two or more interrupt requests are sampled at the same sampling point.

Figure 10.10 shows the Interrupts Priority Select Circuit.


Priority level of each interrupt
Level 0
(initial value)
Highest
INT1
Timer B2
Timer B0, SI/O6 (2)
Timer A3, INT6 (2)
Timer A1
UART1 reception, ACK1
UART0 reception, ACK0
UART2 reception, ACK2
INT2
INT0
Timer B1, INT8 (2)
Timer A4
Timer A2, INT7 (2)
Timer A0
UART1 transmission, NACK1
UART0 transmission, NACK0
A/D conversion, Key input (1)
DMA1
UART2 bus collision detection
SI/O4, INT5
Timer B4, UART1 bus collision detection
INT3
CAN0 successful reception
UART2 transmission, NACK2
CAN0 error (, Key input) (1)
DMA0
SI/O3, INT4
Timer B3, UART0 bus collision detection
Timer B5, SI/O5 (2)
CAN0 successful transmission
CAN0 wake-up
Priority of peripheral function interrupts
(if priority levels are same)
Lowest
IPL
Interrupt request level resolution output to clock generation circuit
(Figure 8.1 Clock Generation Circuit)
I Flag
Address match
Oscillation stop and re-oscillation detection
Watchdog timer
DBC
NMI
Interrupt request accepted

NOTES:
1. If the PCLK6 bit in the PCLKR register is set to 1, the priority level of key input interrupt can be changed.
2. SI/O5, SI/O6 and INT6 to INT8 are only in the 128-pin version.

Figure 10.10 Interrupts Priority Select Circuit

10.6 $\overline{\text{INT}}$ Interrupt

$\overline{\text{INTi}}$ interrupt (i = 0 to 8) [1] is triggered by the edges of external inputs. The edge polarity is selected using bits IFSR10 to IFSR15 in the IFSR1 register and bits IFSR23 to IFSR25 in the IFSR2 register.

$\overline{\text{INT4}}$ share the interrupt vector and interrupt control register with SI/O3, $\overline{\text{INT5}}$ share with SI/O4, $\overline{\text{INT6}}$ share with timer A3, $\overline{\text{INT7}}$ share with timer A2, $\overline{\text{INT8}}$ share with timer B1. To use the $\overline{\text{INT4}}$ to $\overline{\text{INT8}}$ interrupts [1], set the each bits as follows.

- To use the $\overline{\text{INT4}}$ interrupt: Set the IFSR16 bit in the IFSR1 register to 1 ($\overline{\text{INT4}}$).
- To use the $\overline{\text{INT5}}$ interrupt: Set the IFSR17 bit in the IFSR1 register to 1 ($\overline{\text{INT5}}$).
- To use the $\overline{\text{INT6}}$ interrupt: Set the IFSR21 bit in the IFSR2 register to 1 ($\overline{\text{INT6}}$). [1]
- To use the $\overline{\text{INT7}}$ interrupt: Set the IFSR20 bit in the IFSR2 register to 1 ($\overline{\text{INT7}}$). [1]
- To use the $\overline{\text{INT8}}$ interrupt: Set the IFSR22 bit in the IFSR2 register to 1 ($\overline{\text{INT8}}$). [1]

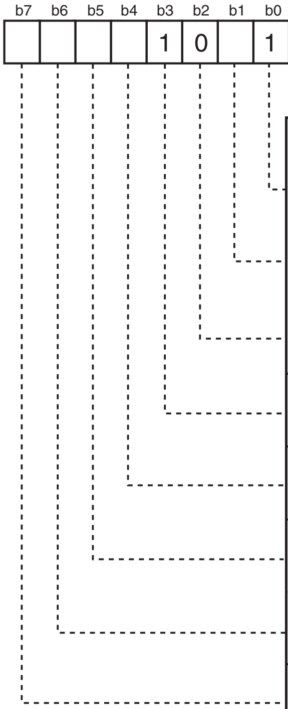
After modifying bits IFSR16, IFSR17, IFSR20, IFSR21, and IFSR22, set the corresponding IR bit to 0 (interrupt not requested) before enabling the interrupt.

NOTE:

1. $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ interrupts are only in the 128-pin version.

Figures 10.11 to 10.13 show Registers IFSR0, IFSR1, and IFSR2.

Interrupt Source Select Register 0



Symbol	Address	After Reset
IFSR0	01DEh	00h

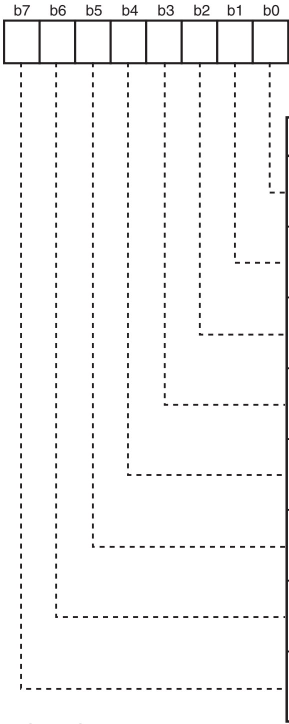
Bit Symbol	Bit Name	Function	RW
IFSR00	Interrupt request source select bit	0 : Do not set a value 1 : SI/O3	RW
IFSR01	Interrupt request source select bit (1)	0 : A/D conversion 1 : Key input	RW
IFSR02	Interrupt request source select bit	0 : CAN0 wake-up or error 1 : Do not set a value	RW
IFSR03	Interrupt request source select bit	0 : Do not set a value 1 : SI/O4	RW
IFSR04	Interrupt request source select bit (2)	0 : Timer B5 1 : SI/O5	RW
IFSR05	Interrupt request source select bit (3)	0 : Timer B0 1 : SI/O6	RW
IFSR06	Interrupt request source select bit (4)	0 : Timer B3 1 : UART0 bus collision detection	RW
IFSR07	Interrupt request source select bit (5)	0 : Timer B4 1 : UART1 bus collision detection	RW

NOTES:

1. When the PCLK6 bit in the PCLKR register = 0, A/D conversion and key input share the vector and interrupt control register. When using the A/D conversion interrupt, set the IFSR01 bit to 0 (A/D conversion). When using the key input interrupt, set the IFSR01 bit to 1 (key input).
2. Timer B5 and SI/O5 share the vector and interrupt control register. When using the timer B5 interrupt, set the IFSR04 bit to 0 (timer B5). When using SI/O5 interrupt, set the IFSR04 bit to 1 (SI/O5).
   The SI/O5 interrupt is only in the 128-pin version. In the 100-pin version, set the IFSR04 bit to 0 (timer B5).
3. Timer B0 and SI/O6 share the vector and interrupt control register. When using the timer B0 interrupt, set the IFSR05 bit to 0 (timer B0). When using SI/O6 interrupt, set the IFSR05 bit to 1 (SI/O6).
   The SI/O6 interrupt is only in the 128-pin version. In the 100-pin version, set the IFSR05 bit to 0 (timer B0).
4. Timer B3 and UART0 bus collision detection share the vector and interrupt control register.
   When using the timer B3 interrupt, set the IFSR06 bit to 0 (timer B3).
   When using UART0 bus collision detection, set the IFSR06 bit to 1 (UART0 bus collision detection).
5. Timer B4 and UART1 bus collision detection share the vector and interrupt control register.
   When using the timer B4 interrupt, set the IFSR07 bit to 0 (timer B4).
   When using UART1 bus collision detection, set the IFSR07 bit to 1 (UART1 bus collision detection).

Figure 10.11 IFSR0 Register

Interrupt Source Select Register 1



Symbol	Address	After Reset
IFSR1	01DFh	00h

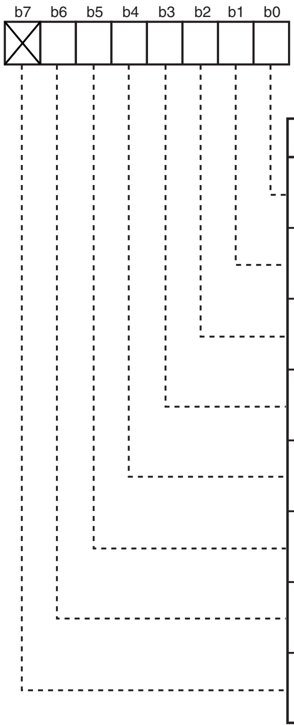
Bit Symbol	Bit Name	Function	RW
IFSR10	INT0 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR11	INT1 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR12	INT2 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR13	INT3 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR14	INT4 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR15	INT5 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR16	Interrupt request source select bit (2)	0 : SI/O3 (3) 1 : $\overline{\mathrm{INT4}}$	RW
IFSR17	Interrupt request source select bit (4)	0 : SI/O4 (5) 1 : $\overline{\mathrm{INT5}}$	RW

NOTES:

1. When setting this bit to 1 (both edges), make sure the POL bit in the INT0IC to INT5IC register is set to 0 (falling edge).
2. SI/O3 and $\overline{\mathrm{INT4}}$ share the vector and interrupt control register. When using SI/O3 interrupt, set the IFSR16 bit to 0 (SI/O3). When using $\overline{\mathrm{INT4}}$ interrupt, set the IFSR16 bit to 1 ($\overline{\mathrm{INT4}}$).
   During memory expansion and microprocessor modes, when the data bus is 16-bit width (BYTE pin is "L"), set this bit to 0 (SI/O3).
3. When setting this bit to 0 (SI/O3), make sure the IFSR00 bit in the IFSR0 register is set to 1 (SI/O3) simultaneously. And, make sure the POL bit in the S3IC register is set to 0 (falling edge).
4. SI/O4 and $\overline{\mathrm{INT5}}$ share the vector and interrupt control register. When using SI/O4 interrupt, set the IFSR17 bit to 0 (SI/O4). When using $\overline{\mathrm{INT5}}$ interrupt, set the IFSR17 bit to 1 ($\overline{\mathrm{INT5}}$).
   During memory expansion and microprocessor modes, when the data bus is 16-bit width (BYTE pin is "L"), set this bit to 0 (SI/O4).
5. When setting this bit to 0 (SI/O4), make sure the IFSR03 bit in the IFSR0 register is set to 1 (SI/O4) simultaneously. And, make sure the POL bit in the S4IC register is set to 0 (falling edge).

Figure 10.12 IFSR1 Register

Interrupt Source Select Register 2



Symbol	Address	After Reset
IFSR2	01CFh	X0000000b

Bit Symbol	Bit Name	Function	RW
IFSR20	Interrupt request source select bit (2) (6)	0 : Timer A2 1 : $\overline{\text{INT7}}$	RW
IFSR21	Interrupt request source select bit (3) (6)	0 : Timer A3 1 : $\overline{\text{INT6}}$	RW
IFSR22	Interrupt request source select bit (4) (6)	0 : Timer B1 1 : $\overline{\text{INT8}}$	RW
IFSR23	INT6 interrupt polarity switching bit (1) (6)	0 : One edge 1 : Both edges	RW
IFSR24	INT7 interrupt polarity switching bit (1) (6)	0 : One edge 1 : Both edges	RW
IFSR25	INT8 interrupt polarity switching bit (1) (6)	0 : One edge 1 : Both edges	RW
IFSR26	Interrupt request source select bit (5)	0 : CAN0 error 1 : key input	RW
- (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		-

NOTES:

1. When setting this bit to 1 (both edges), make sure the POL bit in registers INT6IC to INT8IC are set to 0 (falling edge). Registers INT6IC to INT8IC are only in the 128-pin version.
   In the 100-pin version, make sure bits IFSR23 to IFSR25 are set to 0 (one edge).
2. Timer A2 and $\overline{\text{INT7}}$ share the vector and interrupt control register.
   When using the timer A2 interrupt, set the IFSR20 bit to 0 (timer A2). When using $\overline{\text{INT7}}$ interrupt, set the IFSR20 bit to 1 ($\overline{\text{INT7}}$).
   The $\overline{\text{INT7}}$ interrupt is only in the 128-pin version. In the 100-pin version, set the IFSR20 bit to 0 (timer A2).
3. Timer A3 and $\overline{\text{INT6}}$ share the vector and interrupt control register.
   When using the timer A3 interrupt, set the IFSR21 bit to 0 (timer A3). When using $\overline{\text{INT6}}$ interrupt, set the IFSR21 bit to 1 ($\overline{\text{INT6}}$).
   The $\overline{\text{INT6}}$ interrupt is only in the 128-pin version. In the 100-pin version, set the IFSR21 bit to 0 (timer A3).
4. Timer B1 and $\overline{\text{INT8}}$ share the vector and interrupt control register.
   When using the timer B1 interrupt, set the IFSR22 bit to 0 (timer B1). When using $\overline{\text{INT8}}$ interrupt, set the IFSR22 bit to 1 ($\overline{\text{INT8}}$).
   The $\overline{\text{INT8}}$ interrupt is only in the 128-pin version. In the 100-pin version, set the IFSR22 bit to 0 (timer B1).
5. When the PCLK6 bit in the PCLKR register = 1, CAN0 error and key input share the vector and interrupt control register. When using the CAN0 error interrupt, set the IFSR26 bit to 0 (CAN0 error). When using the key input interrupt, set the IFSR26 bit to 1 (key input).
6. When using the $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ interrupts, set these bits after setting the PU37 bit in the PUR3 register to 1.

Figure 10.13 IFSR2 Register

10.7 NMI Interrupt

An NMI interrupt request is generated when input on the NMI pin changes state from high to low. The NMI interrupt is a non-maskable interrupt.

The input level of this NMI interrupt input pin can be read by accessing the P8_5 bit in the P8 register.

This pin cannot be used as an input port.

10.8 Key Input Interrupt

Of P10_4 to P10_7, a key input interrupt request is generated when input on any of pins P10_4 to P10_7 which has had bits PD10_4 to PD10_7 in the PD10 register set to 0 (input) goes low. Key input interrupts can be used as a key-on wake up function, the function which gets the MCU out of wait or stop mode. However, if you intend to use the key input interrupt, do not use P10_4 to P10_7 as analog input ports. Figure 10.14 shows the Key Input Interrupt Block Diagram. Note, however, that while input on any pin which has had bits PD10_4 to PD10_7 set to 0 (input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

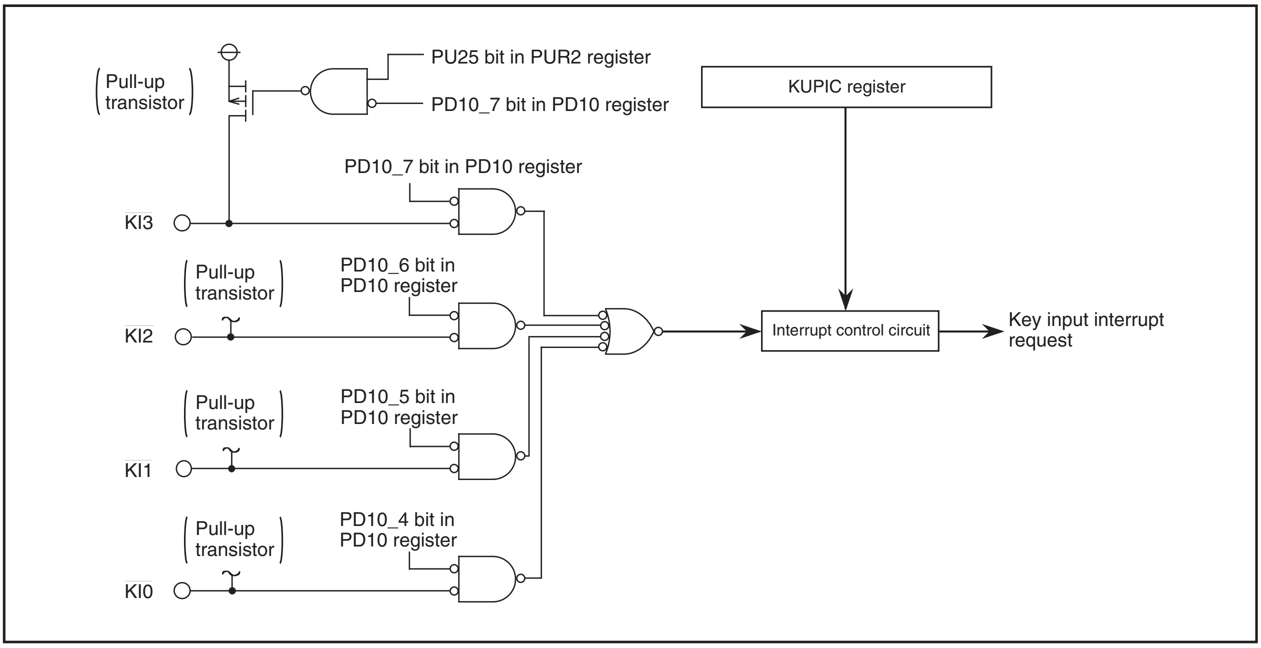

Pull-up transistor
PU25 bit in PUR2 register
PD10_7 bit in PD10 register
KUPIC register
PD10_7 bit in PD10 register
KI3
Pull-up transistor
PD10_6 bit in PD10 register
KI2
Interrupt control circuit
Key input interrupt request
Pull-up transistor
PD10_5 bit in PD10 register
KI1
Pull-up transistor
PD10_4 bit in PD10 register
KI0

Figure 10.14 Key Input Interrupt Block Diagram

10.9 CAN0 Wake-up Interrupt

CAN0 wake-up interrupt request is generated when a falling edge is input to CRX0. The CAN0 wake-up interrupt is enabled only when the PortEn bit = 1 (CTX/CRX function) and Sleep bit = 1 (sleep mode enabled) in the C0CTLR register. Figure 10.15 shows the CAN0 Wake-up Interrupt Block Diagram. Please note that the wake-up message will be lost.

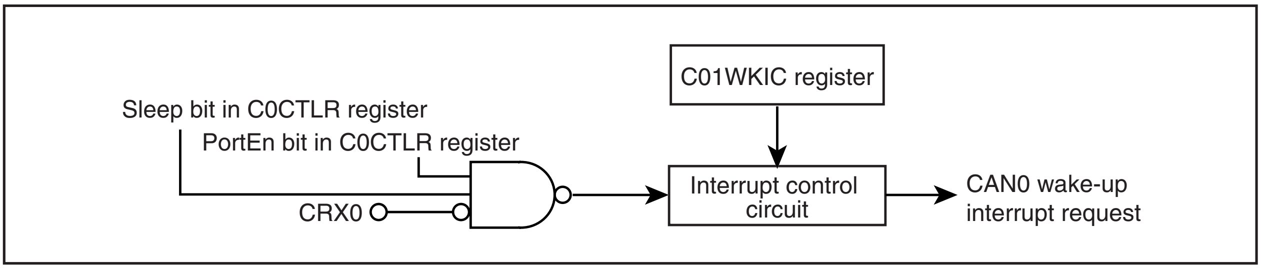

C01WKIC register
Sleep bit in C0CTLR register
PortEn bit in C0CTLR register
CRX0
Interrupt control circuit
CAN0 wake-up interrupt request

Figure 10.15 CAN0 Wake-up Interrupt Block Diagram

10.10 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMADi register (i = 0 to 3). Set the start address of any instruction in the RMADi register. Use bits AIER0 and AIER1 in the AIER register and bits AIER20 and AIER21 in the AIER2 register to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed (refer to **10.5.7 Saving Registers**). (The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

- Rewrite the content of the stack and then use the REIT instruction to return.
- Restore the stack to its previous state before the interrupt request was accepted by using the POP or similar other instruction and then use a jump instruction to return.

Table 10.6 shows the Value of PC that is Saved to Stack Area when Address Match Interrupt Request is Accepted. Table 10.7 shows the Relationship between Address Match Interrupt Sources and Associated Registers.

Note that when using the external bus in 8-bit width, no address match interrupts can be used for external areas.

Figure 10.16 shows Registers AIER, AIER2, and RMAD0 to RMAD3.

Table 10.6 Value of PC that is Saved to Stack Area when Address Match Interrupt Request is Accepted

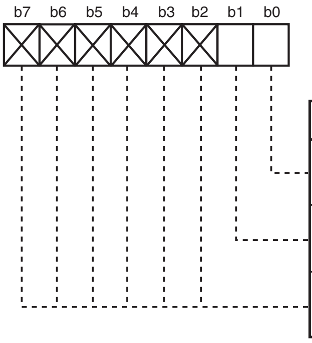
Instruction at Address Indicated by RMADi Register	Value of PC that is Saved to Stack Area
• 16-bit operation code instruction • Instruction shown below among 8-bit operation code instructions ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B:S #IMM8,dest STNZ.B:S #IMM8,dest STZX.B:S #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (However, dest = A0 or A1)	Address indicated by RMADi register + 2
Instructions other than the above	Address indicated by RMADi register + 1

Value of PC that is saved to stack area: Refer to **10.5.7 Saving Registers**.

Table 10.7 Relationship between Address Match Interrupt Sources and Associated Registers

Address Match Interrupt Sources	Address Match Interrupt Enable Bit	Address Match Interrupt Register
Address match interrupt 0	AIER0	RMAD0
Address match interrupt 1	AIER1	RMAD1
Address match interrupt 2	AIER20	RMAD2
Address match interrupt 3	AIER21	RMAD3

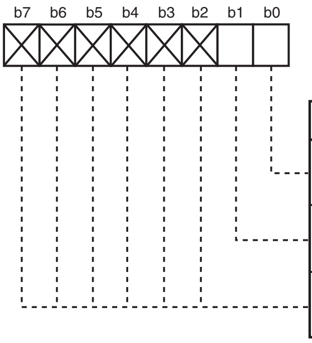
Address Match Interrupt Enable Register



Symbol	Address	After Reset
AIER	0009h	XXXXXX00b

Bit Symbol	Bit Name	Function	RW
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
- (b7-b2)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

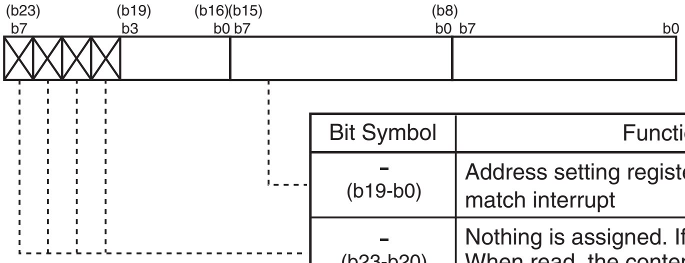
Address Match Interrupt Enable Register 2



Symbol	Address	After Reset
AIER2	01BBh	XXXXXX00b

Bit Symbol	Bit Name	Function	RW
AIER20	Address match interrupt 2 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER21	Address match interrupt 3 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
- (b7-b2)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

Address Match Interrupt Register i (i = 0 to 3)



Symbol	Address	After Reset
RMAD0	0012h to 0010h	X00000h
RMAD1	0016h to 0014h	X00000h
RMAD2	01BAh to 01B8h	X00000h
RMAD3	01BEh to 01BCh	X00000h

Bit Symbol	Function	Setting Range	RW
- (b19-b0)	Address setting register for address match interrupt	00000h to FFFFFh	RW
- (b23-b20)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

Figure 10.16 Registers AIER, AIER2, and RMAD0 to RMAD3

11. Watchdog Timer

The watchdog timer is the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit in the PM1 register. The PM12 bit can only be set to 1 (watchdog timer reset). Once this bit is set to 1, it cannot be set to 0 (watchdog timer interrupt) in a program. Refer to **5.3 Watchdog Timer Reset** for details about watchdog timer reset.

When the main clock, on-chip oscillator clock or PLL clock is selected for CPU clock, the divide-by-n value for the prescaler can be selected to be 16 or 128. If a sub clock is selected for CPU clock, the divide-by-n value for the prescaler is always 2 no matter how the WDC7 bit is set. The period of watchdog timer can be calculated as given below. The period of watchdog timer is, however, subject to an error due to the prescaler.

With main clock, on-chip oscillator clock or PLL clock selected for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (16 or 128)} \times \text{Watchdog timer count (32768)}}{\text{CPU clock}}$$

With sub clock selected for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (2)} \times \text{Watchdog timer count (32768)}}{\text{CPU clock}}$$

For example, when CPU clock = 16 MHz and the divide-by-n value for the prescaler = 16, the watchdog timer period is approx. 32.8 ms.

The watchdog timer is initialized by writing to the WDTS register. The prescaler is initialized after reset. Note that the watchdog timer and the prescaler both are inactive after reset, so that the watchdog timer is activated to start counting by writing to the WDTS register.

In stop mode, wait mode and hold state, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.

Figure 11.1 shows the Watchdog Timer Block Diagram. Figure 11.2 shows Registers WDC and WDTS.

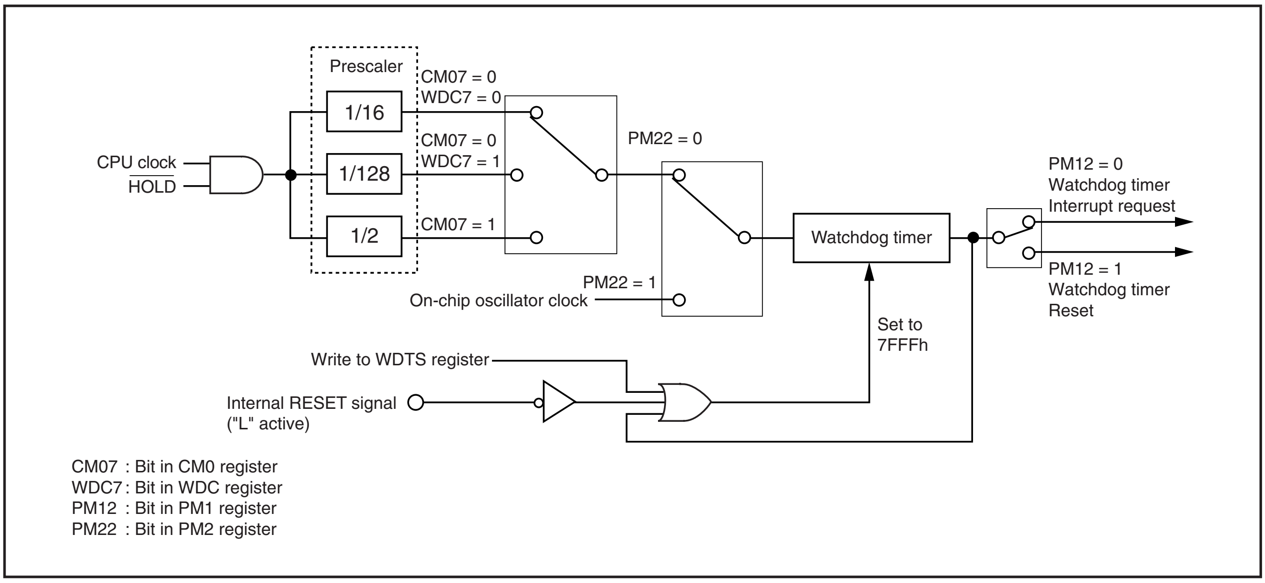
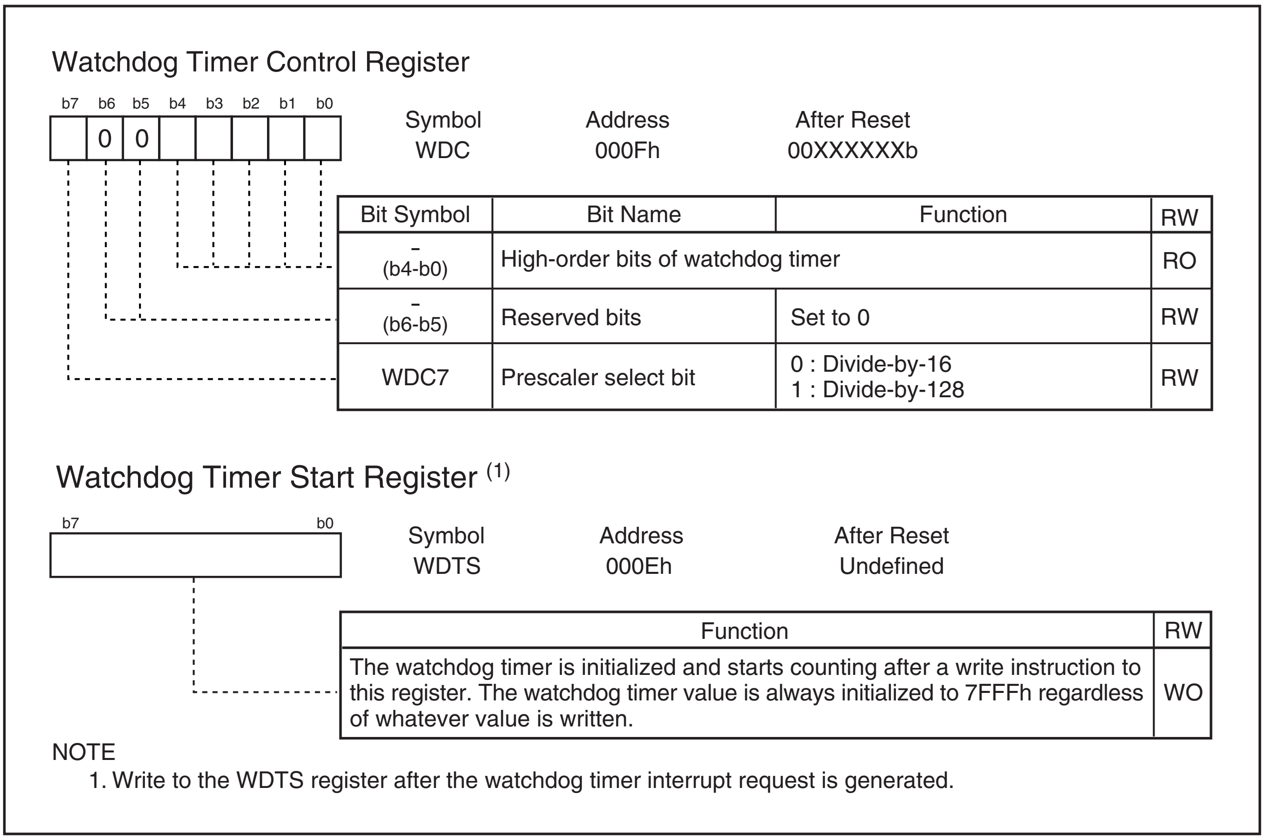


Figure 11.1 Watchdog Timer Block Diagram



Watchdog Timer Control Register

Symbol	Address	After Reset
WDC	000Fh	00XXXXXXb

Bit Symbol	Bit Name	Function	RW
- (b4-b0)	High-order bits of watchdog timer		RO
- (b6-b5)	Reserved bits	Set to 0	RW
WDC7	Prescaler select bit	0 : Divide-by-16 1 : Divide-by-128	RW

Watchdog Timer Start Register (1)

Symbol	Address	After Reset
WDTS	000Eh	Undefined

Function	RW
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to 7FFFh regardless of whatever value is written.	WO

NOTE
1. Write to the WDTS register after the watchdog timer interrupt request is generated.

Figure 11.2 Registers WDC and WDTS

11.1 Count Source Protective Mode

In this mode, a on-chip oscillator clock is used for the watchdog timer count source. The watchdog timer can be kept being clocked even when CPU clock stops as a result of runaway.

Before this mode can be used, the following register settings are required:
(1) Set the PRC1 bit in the PRCR register to 1 (write to registers PM1 and PM2 enabled).
(2) Set the PM12 bit in the PM1 register to 1 (reset when the watchdog timer underflows).
(3) Set the PM22 bit in the PM2 register to 1 (on-chip oscillator clock used for the watchdog timer count source).
(4) Set the PRC1 bit in the PRCR register to 0 (write to registers PM1 and PM2 disabled).
(5) Write to the WDTS register (watchdog timer starts counting).

Setting the PM22 bit to 1 results in the following conditions:

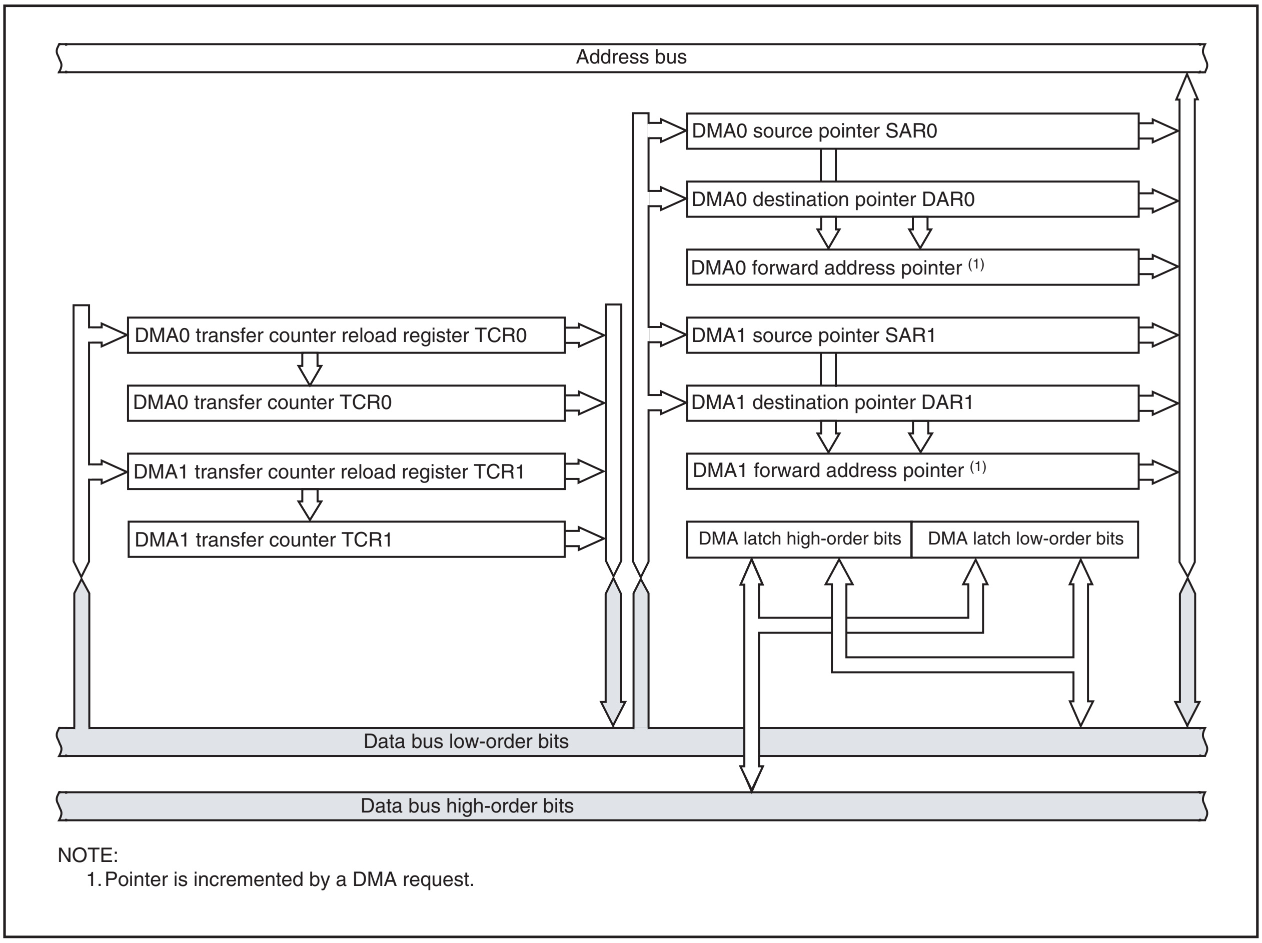
- The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.

$$\text{Watchdog timer period} = \frac{\text{Watchdog timer count (32768)}}{\text{On-chip oscillator clock}}$$

- The CM10 bit in the CM1 register is disabled against write. (Writing a 1 has no effect, nor is stop mode entered.)
- The watchdog timer does not stop when in wait mode or hold state.

12. DMAC

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8- or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. Figure 12.1 shows the DMAC Block Diagram. Table 12.1 lists the DMAC Specifications. Figures 12.2 to 12.4 show the DMAC related-registers.


Address bus
DMA0 source pointer SAR0
DMA0 destination pointer DAR0
DMA0 forward address pointer (1)
DMA1 source pointer SAR1
DMA1 destination pointer DAR1
DMA1 forward address pointer (1)
DMA latch high-order bits
DMA latch low-order bits
DMA0 transfer counter reload register TCR0
DMA0 transfer counter TCR0
DMA1 transfer counter reload register TCR1
DMA1 transfer counter TCR1
Data bus low-order bits
Data bus high-order bits

NOTE:
1. Pointer is incremented by a DMA request.

Figure 12.1 DMAC Block Diagram

A DMA request is generated by a write to the DSR bit in the DMiSL register (i = 0, 1), as well as by an interrupt request which is generated by any function specified by bits DMS, and DSEL3 to DSEL0 in the DMiSL register. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the IR bit in the interrupt control register does not change state due to a DMA transfer.

A data transfer is initiated each time a DMA request is generated when the DMAE bit in the DMiCON register = 1 (DMA enabled). However, if the cycle in which a DMA request is generated is faster than the DMA transfer cycle, the number of transfer requests generated and the number of times data is transferred may not match. For details, refer to **12.4 DMA Request**.

Table 12.1 DMAC Specifications

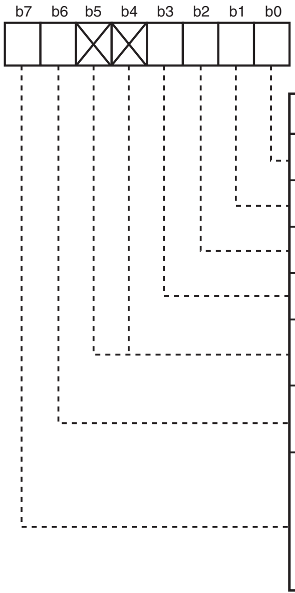
Item		Specification
No. of channels		2 (cycle steal method)
Transfer memory space		• From given address in the 1-Mbyte space to a fixed address • From a fixed address to given address in the 1-Mbyte space • From a fixed address to a fixed address
Maximum no. of bytes transferred		128 Kbytes (with 16-bit transfer) or 64 Kbytes (with 8-bit transfer)
DMA request sources (1) (2)		Falling edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Both edge of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ Timers A0 to A4 interrupt requests Timers B0 to B5 interrupt requests UART0 transmit, UART0 receive interrupt requests UART1 transmit, UART1 receive interrupt requests UART2 transmit, UART2 receive interrupt requests SI/O3, SI/O4 interrupt requests A/D conversion interrupt requests Software triggers
Channel priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer unit		8 bits or 16 bits
Transfer address direction		forward or fixed (The source and destination addresses cannot both be in the forward direction.)
Transfer mode	Single transfer	Transfer is completed when the DMAi transfer counter underflows after reaching the terminal count.
	Repeat transfer	When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is continued with it.
DMA interrupt request generation timing		When the DMAi transfer counter underflowed
DMA start up		Data transfer is initiated each time a DMA request is generated when the The DMAE bit in the DMAiCON register = 1 (enabled).
DMA shutdown	Single transfer	• When the DMAE bit is set to 0 (disabled) • After the DMAi transfer counter underflows
	Repeat transfer	When the DMAE bit is set to 0 (disabled)
Reload timing for forward address pointer and transfer counter		When a data transfer is started after setting the DMAE bit to 1 (enabled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register.
DMA transfer cycles		Minimum 3 cycles between SFR and internal RAM

i = 0, 1

NOTES:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable DMA request sources differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020h to 003Fh) are accessed by the DMAC.

DMA0 Request Source Select Register



Symbol	Address	After Reset
DM0SL	03B8h	00h

Bit Symbol	Bit Name	Function	RW
DSEL0	DMA request source select bits	See **NOTE 1**	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
- (b5-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–
DMS	DMA request source expansion select bit	0 : Basic request source 1 : Extended request source	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to 1 when the DMS bit is 0 (basic source) and bits DSEL3 to DSEL0 are 0001b (software trigger). When read, the content is 0.	RW

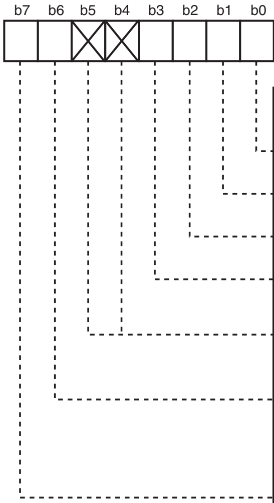
NOTE:

1. The DMA0 request sources can be selected by a combination of the DMS bit and bits DSEL3 to DSEL0 in the manner described below.

Bits DSEL3 to DSEL0	DMS = 0 (basic request source)	DMS = 1 (extended request source)
0000b	Falling edge of $\overline{\text{INT0}}$ pin	–
0001b	Software trigger	–
0010b	Timer A0	–
0011b	Timer A1	–
0100b	Timer A2	–
0101b	Timer A3	–
0110b	Timer A4	Two edges of $\overline{\text{INT0}}$ pin
0111b	Timer B0	Timer B3
1000b	Timer B1	Timer B4
1001b	Timer B2	Timer B5
1010b	UART0 transmit	–
1011b	UART0 receive	–
1100b	UART2 transmit	–
1101b	UART2 receive	–
1110b	A/D conversion	–
1111b	UART1 transmit	–

Figure 12.2 DM0SL Register

DMA1 Request Source Select Register



Symbol	Address	After Reset
DM1SL	03BAh	00h

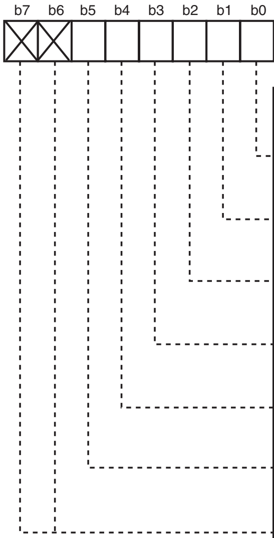
Bit Symbol	Bit Name	Function	RW
DSEL0	DMA request source select bits	See **NOTE 1**	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
- (b5-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–
DMS	DMA request source expansion select bit	0 : Basic request source 1 : Extended request source	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to 1 when the DMS bit is 0 (basic source) and the DSEL3 to DSEL0 bits are 0001b (software trigger). When read, the content is 0.	RW

NOTE:

1. The DMA1 request sources can be selected by a combination of the DMS bit and bits DSEL3 to DSEL0 in the manner described below.

Bits DSEL3 to DSEL0	DMS = 0 (basic request source)	DMS = 1 (extended request source)
0000b	Falling edge of $\overline{\text{INT1}}$ pin	–
0001b	Software trigger	–
0010b	Timer A0	–
0011b	Timer A1	–
0100b	Timer A2	–
0101b	Timer A3	SI/O3
0110b	Timer A4	SI/O4
0111b	Timer B0	Two edges of $\overline{\text{INT1}}$ pin
1000b	Timer B1	–
1001b	Timer B2	–
1010b	UART0 transmit	–
1011b	UART0 receive/ACK0	–
1100b	UART2 transmit	–
1101b	UART2 receive/ACK2	–
1110b	A/D conversion	–
1111b	UART1 receive/ACK1	–

DMAi Control Register (i = 0, 1)



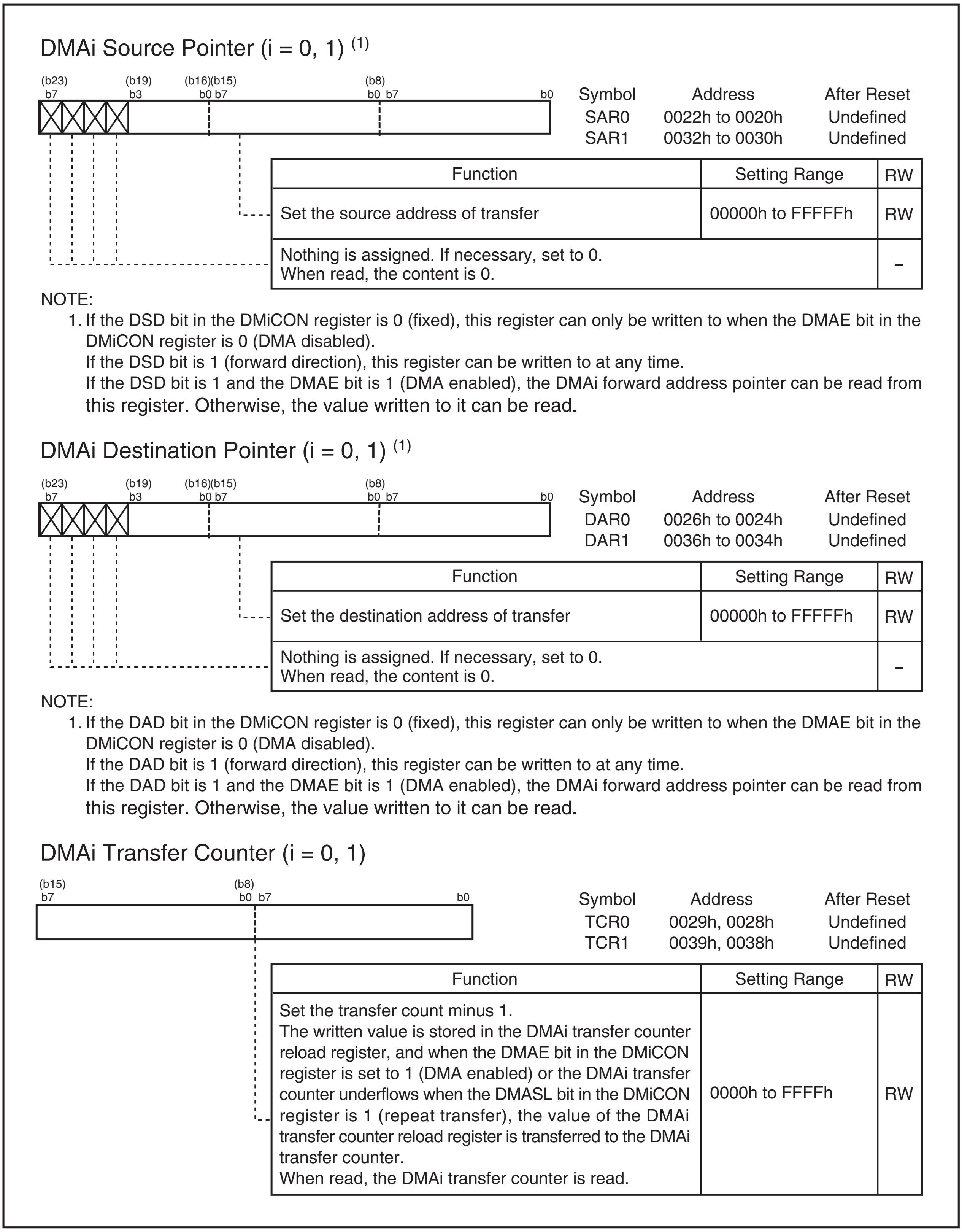
Symbol	Address	After Reset
DM0CON	002Ch	00000X00b
DM1CON	003Ch	00000X00b

Bit Symbol	Bit Name	Function	RW
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	RW
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	RW
DMAS	DMA request bit	0 : DMA not requested 1 : DMA requested	RW (1)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	RW
DSD	Source address direction select bit (2)	0 : Fixed 1 : Forward	RW
DAD	Destination address direction select bit (2)	0 : Fixed 1 : Forward	RW
- (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–

NOTES:

1. The DMAS bit can be set to 0 by writing 0 in a program. (This bit remains unchanged even if 1 is written.)
2. At least one of bits DAD and DSD is set to 0 (address direction fixed).

Figure 12.3 Registers DM1SL, DM0CON, and DM1CON



DMAi Source Pointer (i = 0, 1) (1)

Symbol	Address	After Reset
SAR0	0022h to 0020h	Undefined
SAR1	0032h to 0030h	Undefined

Function	Setting Range	RW
Set the source address of transfer	00000h to FFFFFh	RW
Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–

NOTE:

1. If the DSD bit in the DMiCON register is 0 (fixed), this register can only be written to when the DMAE bit in the DMiCON register is 0 (DMA disabled).
   If the DSD bit is 1 (forward direction), this register can be written to at any time.
   If the DSD bit is 1 and the DMAE bit is 1 (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi Destination Pointer (i = 0, 1) (1)

Symbol	Address	After Reset
DAR0	0026h to 0024h	Undefined
DAR1	0036h to 0034h	Undefined

Function	Setting Range	RW
Set the destination address of transfer	00000h to FFFFFh	RW
Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–

NOTE:

1. If the DAD bit in the DMiCON register is 0 (fixed), this register can only be written to when the DMAE bit in the DMiCON register is 0 (DMA disabled).
   If the DAD bit is 1 (forward direction), this register can be written to at any time.
   If the DAD bit is 1 and the DMAE bit is 1 (DMA enabled), the DMAi forward address pointer can be read from this register. Otherwise, the value written to it can be read.

DMAi Transfer Counter (i = 0, 1)

Symbol	Address	After Reset
TCR0	0029h, 0028h	Undefined
TCR1	0039h, 0038h	Undefined

Function	Setting Range	RW
Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit in the DMiCON register is set to 1 (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit in the DMiCON register is 1 (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read.	0000h to FFFFh	RW

Figure 12.4 Registers SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1

12.1 Transfer Cycle

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. During memory expansion and microprocessor modes, it is also affected by the BYTE pin level. Furthermore, the bus cycle itself is extended by a software wait or $\overline{\text{RDY}}$ signal.

12.1.1 Effect of Source and Destination Addresses

If the transfer unit and data bus both are 16 bits and the source address of transfer begins with an odd address, the source read cycle consists of one more bus cycle than when the source address of transfer begins with an even address.

Similarly, if the transfer unit and data bus both are 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

12.1.2 Effect of BYTE Pin Level

During memory expansion and microprocessor modes, if 16 bits of data are to be transferred on an 8-bit data bus (input on the BYTE pin = high), the operation is accomplished by transferring 8 bits of data twice. Therefore, this operation requires two bus cycles to read data and two bus cycles to write data.

Furthermore, if the DMAC is to access the internal area (internal ROM, internal RAM, or SFR), unlike in the case of the CPU, the DMAC does it through the data bus width selected by the BYTE pin.

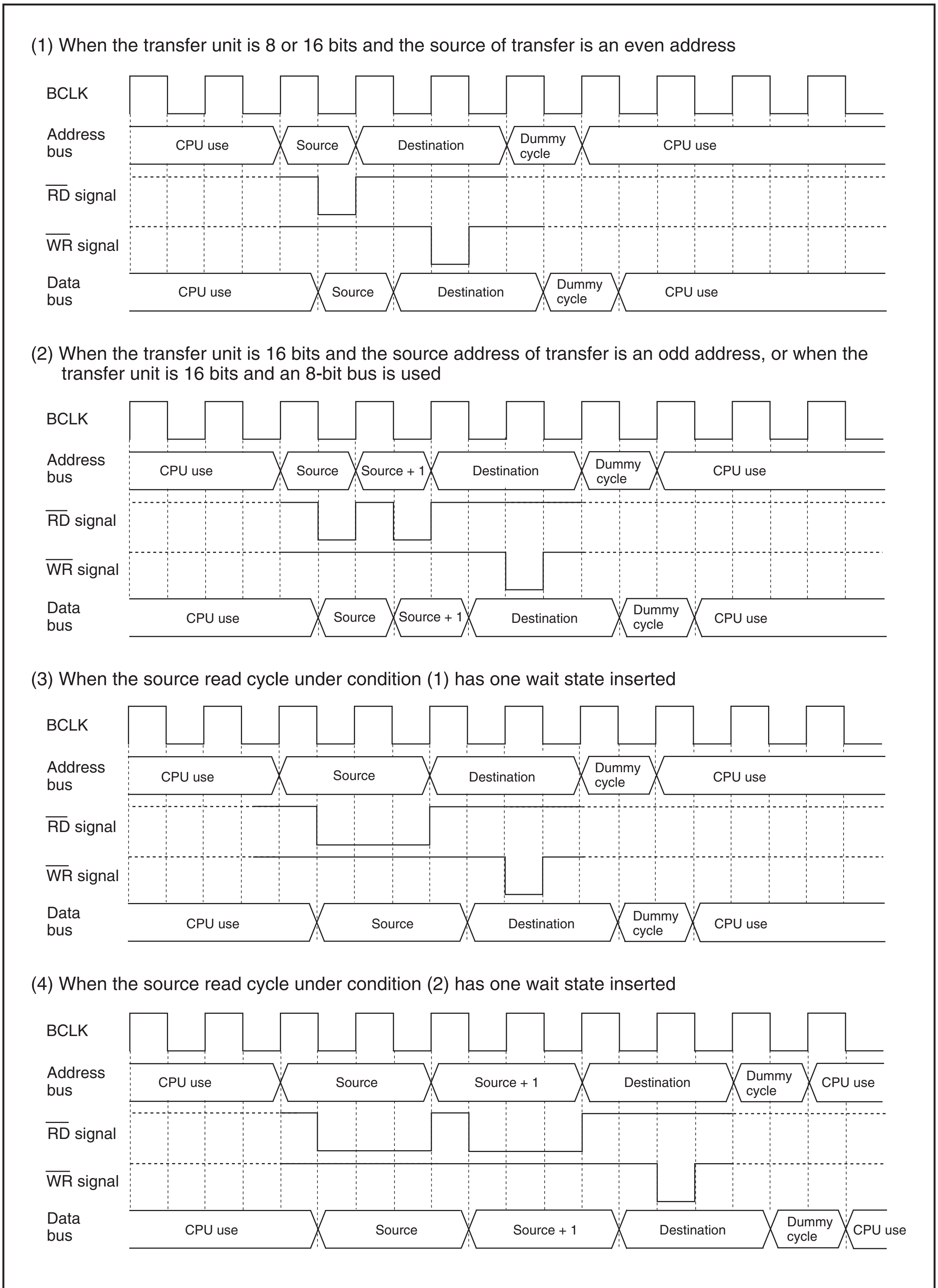
12.1.3 Effect of Software Wait

For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

12.1.4 Effect of $\overline{\text{RDY}}$ Signal

During memory expansion and microprocessor modes, DMA transfers to and from an external area are affected by the $\overline{\text{RDY}}$ signal. Refer to **7.2.6 $\overline{\text{RDY}}$ Signal**.

Figure 12.5 shows the Transfer Cycles for Source Read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16-bit unit using an 8-bit bus ((2) on Figure 12.5), two source read bus cycles and two destination write bus cycles are required.



(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address

(2) When the transfer unit is 16 bits and the source address of transfer is an odd address, or when the transfer unit is 16 bits and an 8-bit bus is used

(3) When the source read cycle under condition (1) has one wait state inserted

(4) When the source read cycle under condition (2) has one wait state inserted

NOTE:

1. The same timing changes occur with the respective conditions at the destination as at the source.

Figure 12.5 Transfer Cycles for Source Read

12.2 DMA Transfer Cycles

Any combination of even or odd transfer read and write addresses is possible.

Table 12.2 lists the DMA Transfer Cycles. Table 12.3 lists the Coefficient j, k.

The number of DMAC transfer cycles can be calculated as follows:

No. of transfer cycles per transfer unit = No. of read cycles × j + No. of write cycles × k

Table 12.2 DMA Transfer Cycles

Transfer Unit	Bus Width	Access Address	Single-chip Mode		Memory Expansion Mode Microprocessor Mode	
			No. of Read Cycles	No. of Write Cycles	No. of Read Cycles	No. of Write Cycles
8-bit transfer (DMBIT =1)	16 bits (BYTE = L)	Even	1	1	1	1
		Odd	1	1	1	1
	8 bits (BYTE= H)	Even	-	-	1	1
		Odd	-	-	1	1
16-bit transfer (DMBIT = 0)	16 bits (BYTE =L)	Even	1	1	1	1
		Odd	2	2	2	2
	8 bits (BYTE = H)	Even	-	-	2	2
		Odd	-	-	2	2

-: This condition does not exist.

Table 12.3 Coefficient j, k

	Internal Area				External Area						
	Internal ROM, RAM		SFR		Separate Bus				Multiplexed Bus		
	No Wait	With Wait	1 Wait [1]	2 Waits [1]	No Wait	With Wait [2]			With Wait [2]		
						1 Wait	2 Waits	3 Waits	1 Wait	2 Waits	3 Waits
j	1	2	2	3	1	2	3	4	3	3	4
k	1	2	2	3	2	2	3	4	3	3	4

NOTES:

1. Depends on the set value of the PM20 bit in the PM2 register.
2. Depends on the set value of the CSE register.

12.3 DMA Enable

When a data transfer starts after setting the DMAE bit in the DMiCON register (i = 0, 1) to 1 (enabled), the DMAC operates as follows:

(1) Reload the forward address pointer with the SARi register value when the DSD bit in the DMiCON register is 1 (forward) or the DARi register value when the DAD bit in the DMiCON register is 1 (forward).

(2) Reload the DMAi transfer counter with the DMAi transfer counter reload register value.

If the DMAE bit is set to 1 again while it remains set, the DMAC performs the above operation.
However, if a DMA request may occur simultaneously when the DMAE bit is being written, follow the steps below.

Step 1: Write 1 to the DMAE bit and DMAS bit in the DMiCON register simultaneously.
Step 2: Make sure that the DMAi is in an initial state as described above (1) and (2) in a program.

If the DMAi is not in an initial state, the above steps should be repeated.

12.4 DMA Request

The DMAC can generate a DMA request as triggered by the request source that is selected with bits DMS, and DSEL3 to DSEL0 in the DMiSL register (i = 0, 1) on either channel.
Table 12.4 lists the Timing at which DMAS Bit Changes State.
Whenever a DMA request is generated, the DMAS bit is set to 1 (DMA requested) regardless of whether or not the DMAE bit is set. If the DMAE bit was set to 1 (enabled) when this occurred, the DMAS bit is set to 0 (DMA not requested) immediately before a data transfer starts. This bit cannot be set to 1 in a program (it can only be set to 0).
The DMAS bit may be set to 1 when the DMS bit or bits DSEL3 to DSEL0 change state. Therefore, always be sure to set the DMAS bit to 0 after changing the DMS bit or bits DSEL3 to DSEL0.
Because if the DMAE bit is 1, a data transfer starts immediately after a DMA request is generated, the DMAS bit in almost all cases is 0 when read in a program. Read the DMAE bit to determine whether the DMAC is enabled.

Table 12.4 Timing at which DMAS Bit Changes State

DMA Source	DMAS Bit in DMiCON Register	
	Timing at which the bit is set to 1	Timing at which the bit is set to 0
Software trigger	When the DSR bit in the DMiSL register is set to 1	• Immediately before a data transfer starts • When set by writing 0 in a program
Peripheral function	When the interrupt control register for the peripheral function that is selected by bits DSEL3 to DSEL0, and DMS in the DMiSL register has its IR bit set to 1.	

i = 0, 1

12.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of BCLK), the DMAS bit on each channel is set to 1 (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1.

The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period.

Figure 12.6 shows an example of DMA Transfer by External Sources.

In Figure 12.6, DMA0 request having priority is received first to start a transfer when a DMA0 request and DMA1 request are generated simultaneously. After one DMA0 transfer is completed, a bus arbitration is returned to the CPU. When the CPU has completed one bus access, a DMA1 transfer starts. After one DMA1 transfer is completed, the bus arbitration is again returned to the CPU.

In addition, DMA requests cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in Figure 12.6, occurs more than one time, the DMAS bit is set to 0 as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

Refer to **7.2.7 $\overline{\text{HOLD}}$ Signal** for details about bus arbitration between the CPU and DMA.

An example where DMA requests for external sources are detected active at the same time, a DMA transfer is executed in the shortest cycle.

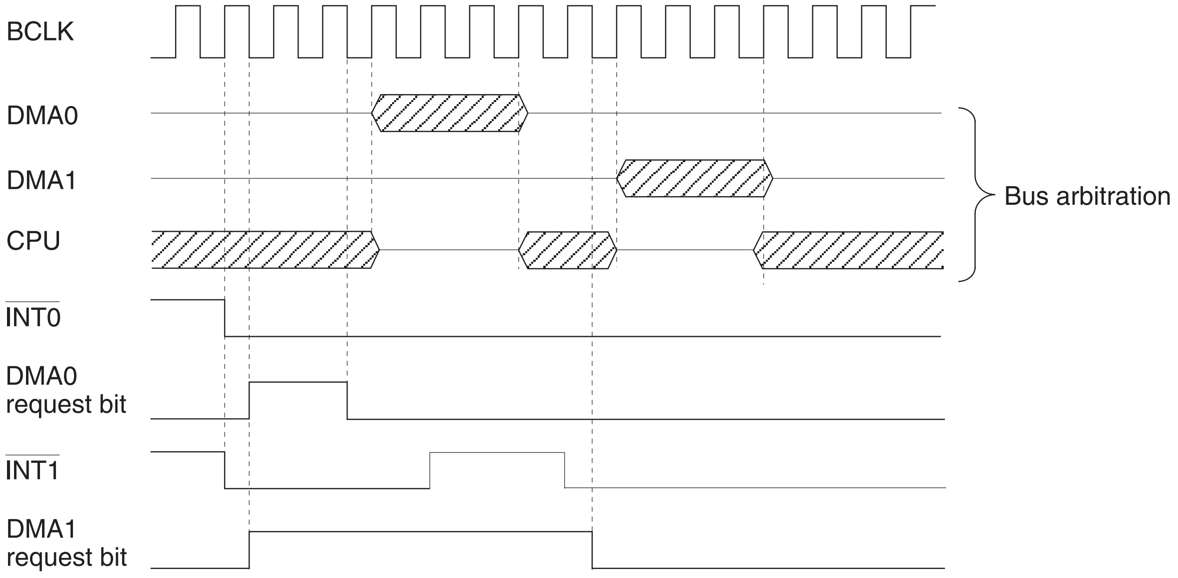
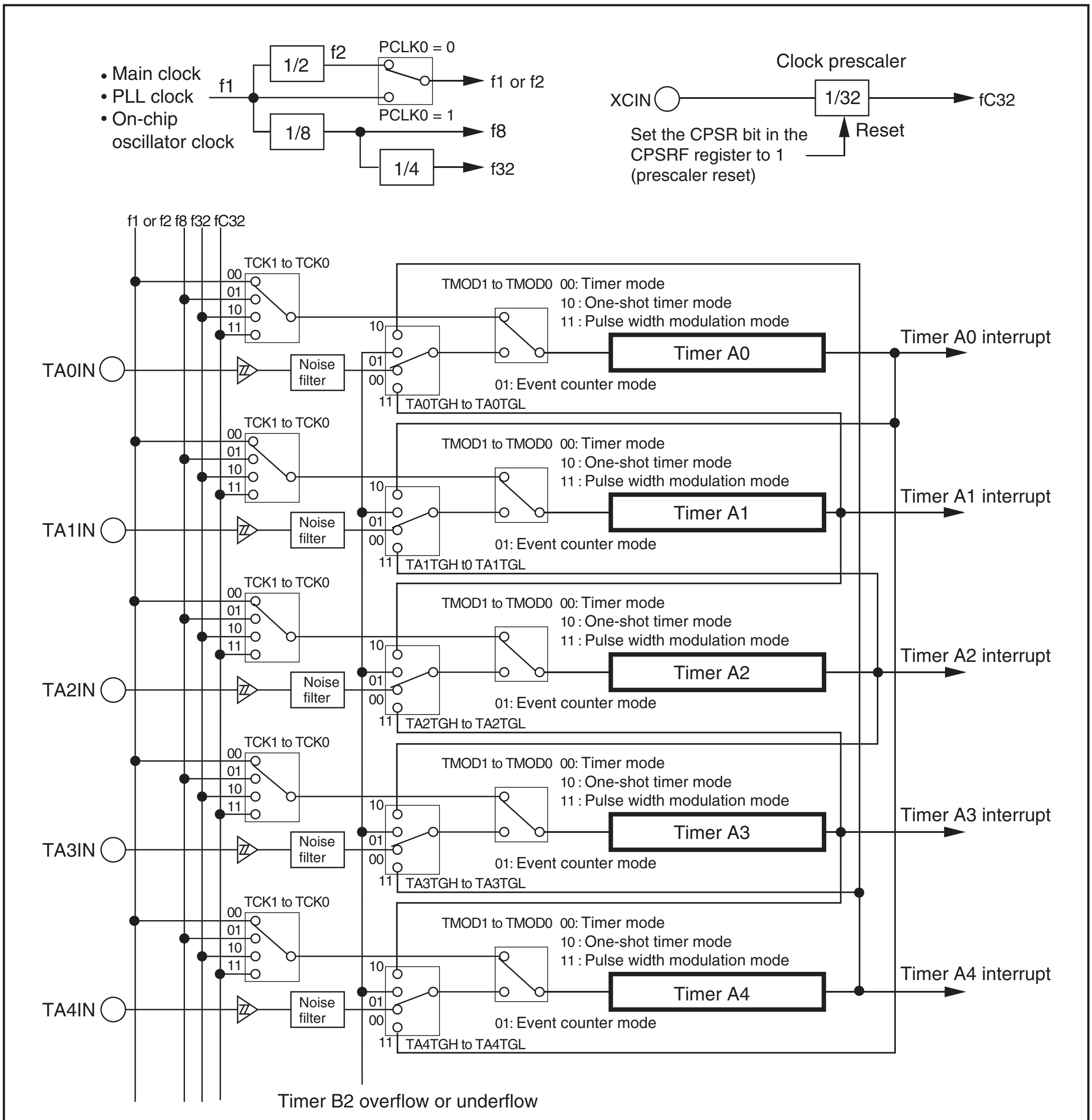


Figure 12.6 DMA Transfer by External Sources

13. Timers

Eleven 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (six). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc.

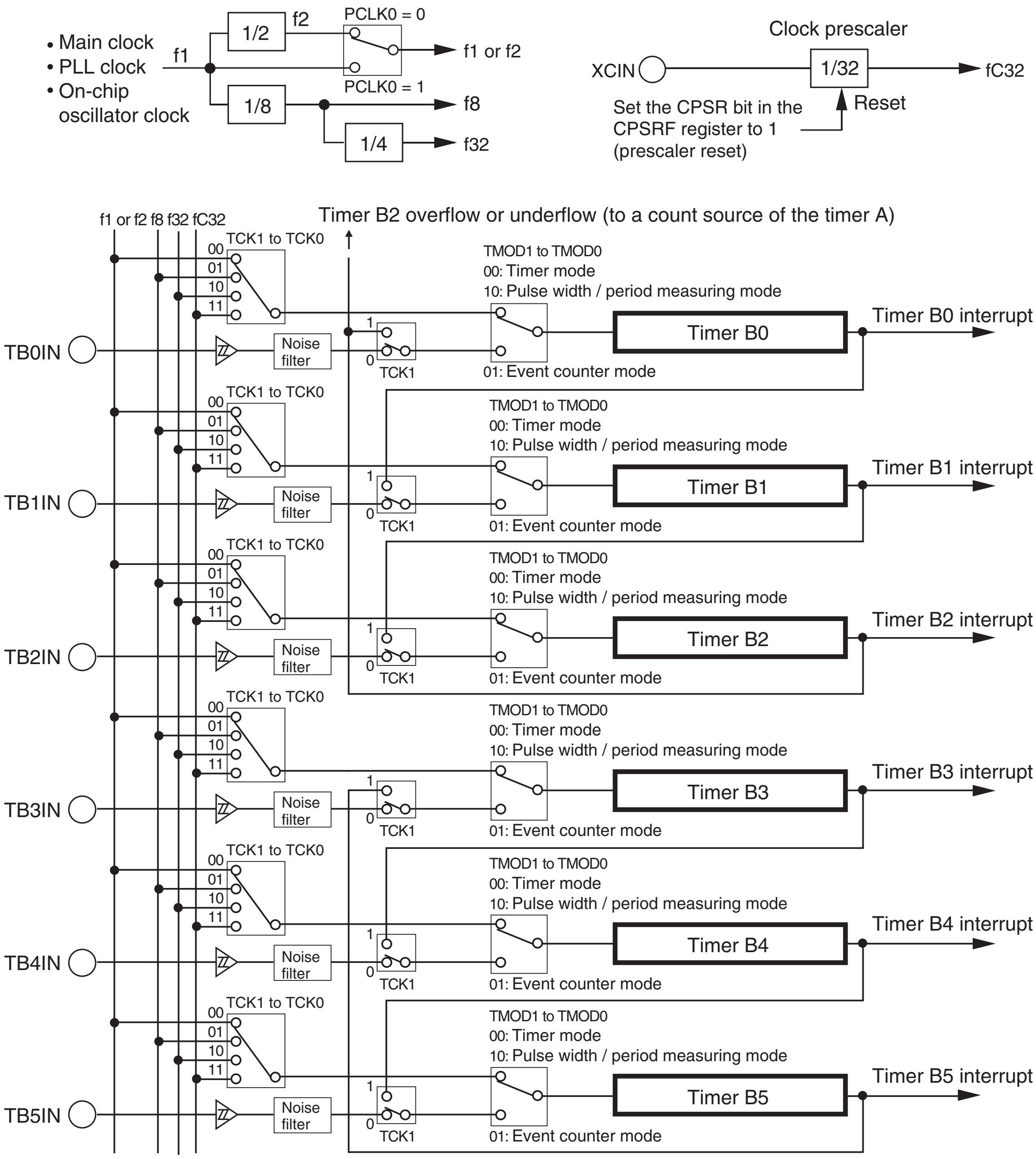
Figures 13.1 and 13.2 show the Timer A and Timer B Configurations.



PCLK0: Bit in PCLKR register
TCK1 to TCK0, TMOD1 to TMOD0: Bits in TAiMR register (i = 0 to 4)
TAiTGH to TAiTGL: Bits in ONSF register or TRGSR register

NOTE:
1. Be aware that TA0IN shares the pin with RXD2, SCL2, and TB5IN.

Figure 13.1 Timer A Configuration



PCLK0: Bit in PCLKR register
TCK1 to TCK0, TMOD1 to TMOD0: Bits in TBiMR register (i = 0 to 5)

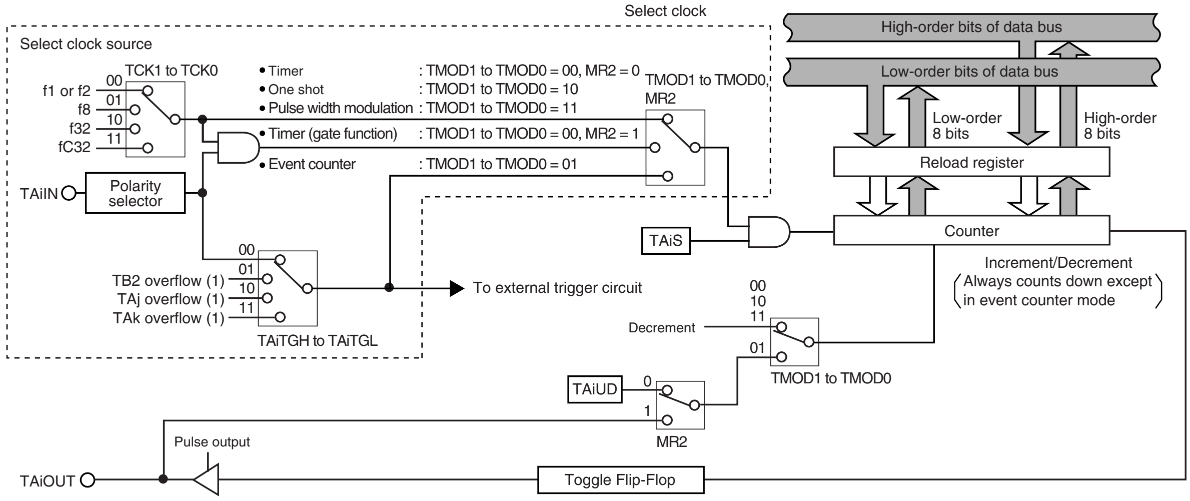
NOTE:
1. Be aware that TB5IN shares the pin with RXD2, SCL2, and TA0IN.

Figure 13.2 Timer B Configuration

13.1 Timer A

Figure 13.3 shows the Timer A Block Diagram. Figures 13.4 to 13.6 show the timer A-related registers. The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use bits TMOD1 to TMOD0 in the TAiMR register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count 0000h.
- Pulse width modulation mode: The timer outputs pulses in a given width successively.


Select clock
Select clock source
TCK1 to TCK0
00
01
10
11
f1 or f2
f8
f32
fC32
• Timer : TMOD1 to TMOD0 = 00, MR2 = 0
• One shot : TMOD1 to TMOD0 = 10
• Pulse width modulation : TMOD1 to TMOD0 = 11
• Timer (gate function) : TMOD1 to TMOD0 = 00, MR2 = 1
• Event counter : TMOD1 to TMOD0 = 01
TMOD1 to TMOD0, MR2
TAiIN
Polarity selector
High-order bits of data bus
Low-order bits of data bus
Low-order 8 bits
High-order 8 bits
Reload register
Counter
TAiS
00
01
10
11
TB2 overflow (1)
TAj overflow (1)
TAk overflow (1)
TAiTGH to TAiTGL
To external trigger circuit
Increment/Decrement
Always counts down except in event counter mode
00
10
11
01
Decrement
TMOD1 to TMOD0
TAiUD
0
1
MR2
Pulse output
TAiOUT
Toggle Flip-Flop

TCK1 to TCK0, TMOD1 to TMOD0, MR2 to MR1: Bits in TAiMR register
TAiTGH to TAiTGL: Bits in ONSF register If i = 0, bits in TRGSR register if i = 1 to 4
TAiS: Bit in TABSR register
TAiUD: Bit in UDF register

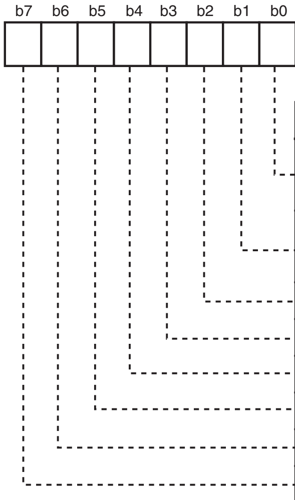
i = 0 to 4
j = i - 1except j = 4 when i = 0
k = i + 1 except k = 0 when i = 4

NOTE:
1. Overflow or underflow

TAi	Addresses	TAj	TAk
Timer A0	0387h - 0386h	Timer A4	Timer A1
Timer A1	0389h - 0388h	Timer A0	Timer A2
Timer A2	038Bh- 038Ah	Timer A1	Timer A3
Timer A3	038Dh- 038Ch	Timer A2	Timer A4
Timer A4	038Fh- 038Eh	Timer A3	Timer A0

Figure 13.3 Timer A Block Diagram

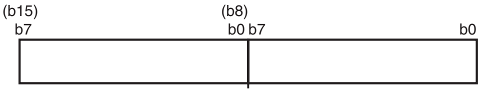
Timer Ai Mode Register (i = 0 to 4)



Symbol	Address	After Reset
TA0MR to TA4MR	0396h to 039Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation mode	RW
TMOD1			RW
MR0	Function varies depending on operating mode		RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count source select bits	Function varies depending on operating mode	RW
TCK1			RW

Timer Ai Register (i = 0 to 4) (1)



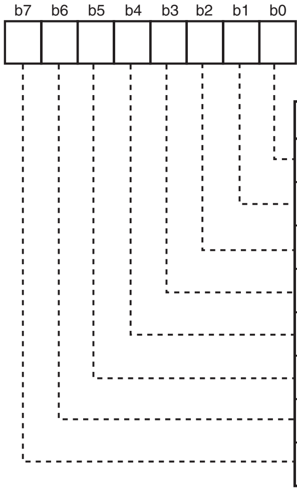
Symbol	Address	After Reset
TA0	0387h, 0386h	Undefined
TA1	0389h, 0388h	Undefined
TA2	038Bh, 038Ah	Undefined
TA3	038Dh, 038Ch	Undefined
TA4	038Fh, 038Eh	Undefined

Mode	Function	Setting Range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000h to FFFFh	RW
Event counter mode	Divide the count source by FFFFh – n + 1 where n = set value when counting up or by n + 1 when counting down (2)	0000h to FFFFh	RW
One-shot timer mode	Divide the count source by n where n = set value and cause the timer to stop	0000h to FFFFh (3) (4)	WO
Pulse width modulation mode (16-bit PWM)	Modify the pulse width as follows: PWM period: $(2^{16} – 1) / fj$ High level PWM pulse width: n / fj where n = set value, fj = count source frequency	0000h to FFFEh (4) (5)	WO
Pulse width modulation mode (8-bit PWM)	Modify the pulse width as follows: PWM period: $(2^{8} – 1) \times (m + 1)/ fj$ High level PWM pulse width: $(m + 1)n / fj$ where n = high-order address set value, m = low-order address set value, fj = count source frequency	00h to FEh (High-order address) 00h to FFh (4) (5) (Low-order address)	WO

NOTES:
1. The register must be accessed in 16-bit unit.
2. The timer counts pulses from an external device or overflows or underflows in other timers.
3. If the TAi register is set to 0000h, the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if "pulse output" is selected, no pulses are output from the TAiOUT pin.
4. Use the MOV instruction to write to the TAi register.
5. If the TAi register is set to 0000h, the pulse width modulator does not work, the output level on the TAiOUT pin remains low, and timer Ai interrupt requests are not generated either.
The same applies when the 8 high-order bits in the TAi register are set to 00h while operating as an 8-bit pulse width modulator.

Figure 13.4 Registers TA0MR to TA4MR, and TA0 to TA4

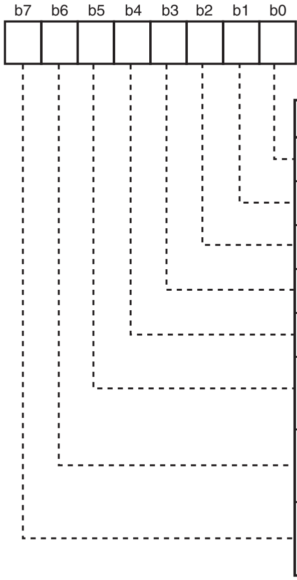
Count Start Flag



Symbol TABSR | Address 0380h | After Reset 00h

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 count start flag	0 : Count stops 1 : Count starts	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Up/Down Flag [1]



Symbol UDF | Address 0384h | After Reset 00h

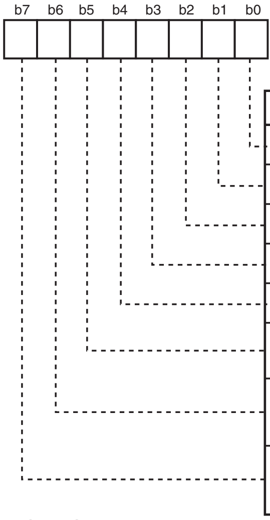
Bit Symbol	Bit Name	Function	RW
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count Enabled by setting the MR2 bit in the TAiMR register to 0 (= switching source in UDF register) during event counter mode.	RW
TA1UD	Timer A1 up/down flag		RW
TA2UD	Timer A2 up/down flag		RW
TA3UD	Timer A3 up/down flag		RW
TA4UD	Timer A4 up/down flag		RW
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : Two-phase pulse signal processing disabled 1 : Two-phase pulse signal processing enabled [2] [3]	WO
TA3P	Timer A3 two-phase pulse signal processing select bit		WO
TA4P	Timer A4 two-phase pulse signal processing select bit		WO

NOTES:

1. Use the MOV instruction to write to this register.
2. Make sure the port direction bits for pins TA2IN to TA4IN, and TA2OUT to TA4OUT are set to 0 (input mode).
3. When not using the two-phase pulse signal processing function, set the corresponding bit to timers A2 to A4 to 0.

Figure 13.5 Registers TABSR and UDF

One-Shot Start Flag



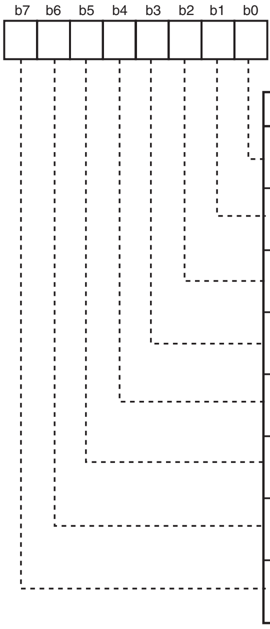
Symbol	Address	After Reset
ONSF	0382h	00h

Bit Symbol	Bit Name	Function	RW
TA0OS	Timer A0 one-shot start flag	The timer starts counting by setting this bit to 1 while bits TMOD1 to TMOD0 in the TAiMR register (i = 0 to 4) = 10b (one-shot timer mode) and the MR2 bit in the TAiMR register = 0 (TAiOS bit enabled). When read, its content is 0.	RW
TA1OS	Timer A1 one-shot start flag		RW
TA2OS	Timer A2 one-shot start flag		RW
TA3OS	Timer A3 one-shot start flag		RW
TA4OS	Timer A4 one-shot start flag		RW
TAZIE	Z-phase input enable bit	0 : Z-phase input disabled 1 : Z-phase input enabled	RW
TA0TGL	Timer A0 event/trigger select bits	b7 b6 0 0 : Input on TA0IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA4 is selected (2) 1 1 : TA1 is selected (2)	RW
TA0TGH			RW

NOTES:

1. Make sure the PD7_1 bit in the PD7 register is set to 0 (input mode).
2. Overflow or underflow.

Trigger Select Register



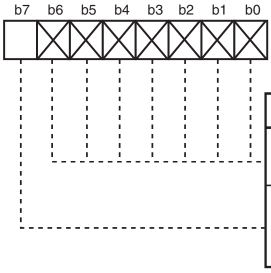
Symbol	Address	After Reset
TRGSR	0383h	00h

Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 event/trigger select bits	b1 b0 0 0 : Input on TA1IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA0 is selected (2) 1 1 : TA2 is selected (2)	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bits	b3 b2 0 0 : Input on TA2IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA1 is selected (2) 1 1 : TA3 is selected (2)	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bits	b5 b4 0 0 : Input on TA3IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA2 is selected (2) 1 1 : TA4 is selected (2)	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bits	b7 b6 0 0 : Input on TA4IN is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA3 is selected (2) 1 1 : TA0 is selected (2)	RW
TA4TGH			RW

NOTES:

1. Make sure the port direction bits for pins TA1IN to TA4IN are set to 0 (input mode).
2. Overflow or underflow.

Clock Prescaler Reset Flag



Symbol	Address	After Reset
CPSRF	0381h	0XXXXXXXb

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—
CPSR	Clock prescaler reset flag	Setting this bit to 1 initializes the prescaler for the timekeeping clock. (When read, the content is 0.)	RW

Figure 13.6 Registers ONSF, TRGSR, and CPSRF

13.1.1 Timer Mode

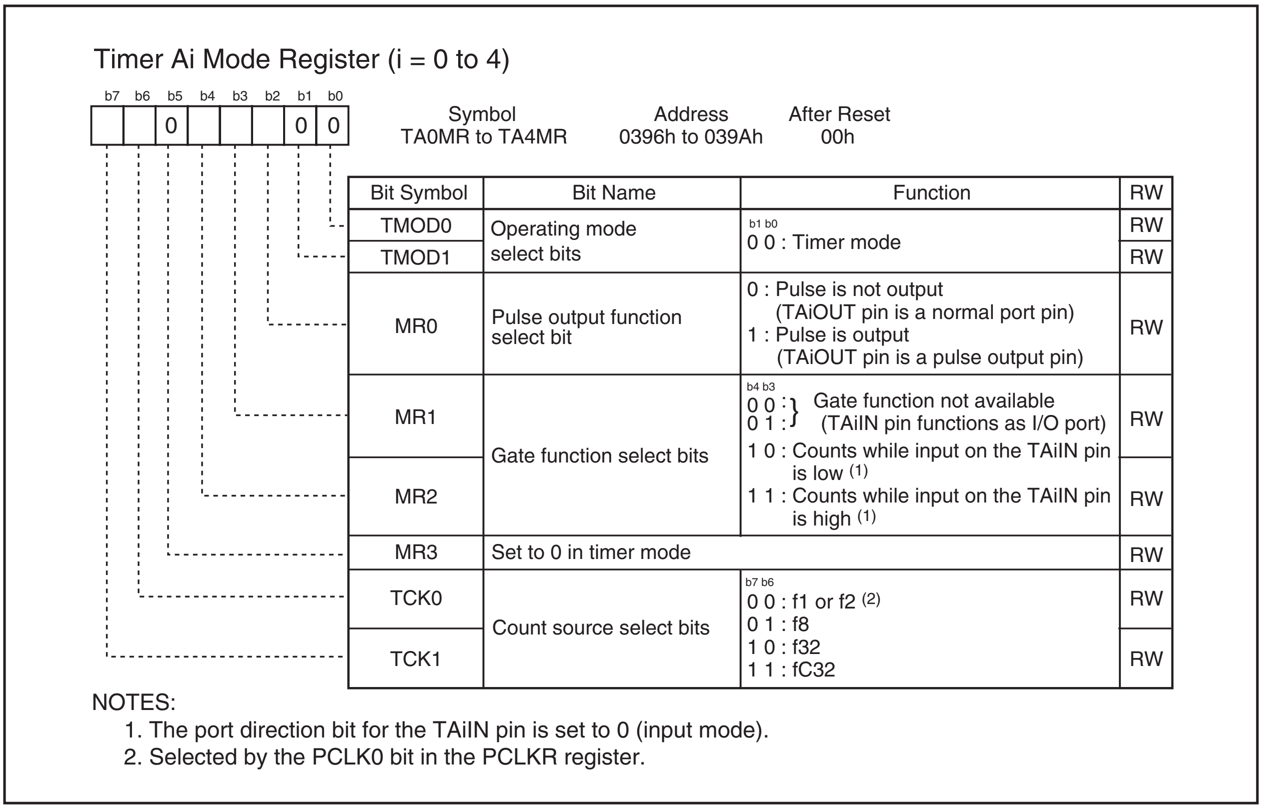
In timer mode, the timer counts a count source generated internally.

Table 13.1 lists the Timer Mode Specifications. Figure 13.7 shows Registers TA0MR to TA4MR in Timer Mode.

Table 13.1 Timer Mode Specifications

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of the TAi register 0000h to FFFFh
Count start condition	Set the TAiS bit in the TABSR register to 1 (count starts)
Count stop condition	Set the TAiS bit to 0 (count stops)
Interrupt request generation timing	Timer underflow
TAiIN pin function	I/O port or gate input
TAiOUT pin function	I/O port or pulse output
Read from timer	Count value can be read by reading the TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to the TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Gate function Counting can be started and stopped by an input signal to TAiIN pin • Pulse output function Whenever the timer underflows, the output polarity of TAiOUT pin is inverted. When TAiS bit is set to 0 (count stops), the pin outputs a low.

i = 0 to 4



Timer Ai Mode Register (i = 0 to 4)

Symbol	Address	After Reset
TA0MR to TA4MR	0396h to 039Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (TAiOUT pin is a pulse output pin)	RW
MR1	Gate function select bits	b4 b3 0 0 : } Gate function not available 0 1 : } (TAiIN pin functions as I/O port) 1 0 : Counts while input on the TAiIN pin is low (1) 1 1 : Counts while input on the TAiIN pin is high (1)	RW
MR2			RW
MR3	Set to 0 in timer mode		RW
TCK0	Count source select bits	b7 b6 0 0 : f1 or f2 (2) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES:
1. The port direction bit for the TAiIN pin is set to 0 (input mode).
2. Selected by the PCLK0 bit in the PCLKR register.

Figure 13.7 Registers TA0MR to TA4MR in Timer Mode

13.1.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3, and A4 can count two-phase external signals. Table 13.2 lists the Event Counter Mode Specifications (when not using two-phase pulse signal processing). Figure 13.8 shows TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing). Table 13.3 lists the Event Counter Mode Specifications (when using two-phase pulse signal processing with timers A2, A3, and A4). Figure 13.9 shows Registers TA2MR to TA4MR in Event Counter Mode (when using two-phase pulse signa processing with timers A2, A3, and A4).

Table 13.2 Event Counter Mode Specifications (when not using two-phase pulse signal processing)

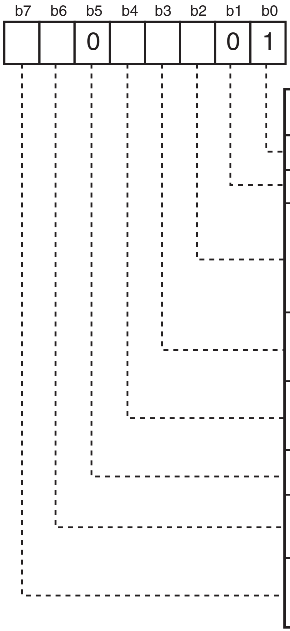
Item	Specification
Count source	• External signals input to TAiIN pin (effective edge can be selected in program) • Timer B2 overflows or underflows, Timer Aj overflows or underflows, Timer Ak overflows or underflows
Count operation	• Up-count or down-count can be selected by external signal or program • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided ratio	1/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of the TAi register 0000h to FFFFh
Count start condition	Set the TAiS bit in the TABSR register to 1 (count starts)
Count stop condition	Set the TAiS bit to 0 (count stops)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	I/O port or count source input
TAiOUT pin function	I/O port, pulse output, or up/down-count select input
Read from timer	Count value can be read by reading the TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to the TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Whenever the timer underflows or underflows, the output polarity of TAiOUT pin is inverted. When TAiS bit is set to 0 (count stops), the pin outputs a low.

i = 0 to 4

j = i - 1, except j = 4 if i = 0

k = i + 1, except k = 0 if i = 4

Timer Ai Mode Register (i = 0 to 4)
(When not using two-phase pulse signal processing)



Symbol	Address	After Reset
TA0MR to TA4MR	0396h to 039Ah	00h

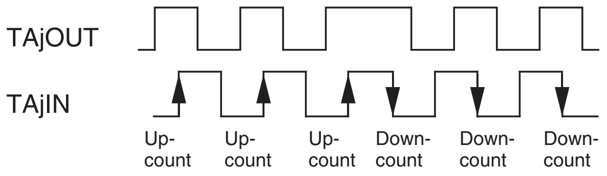
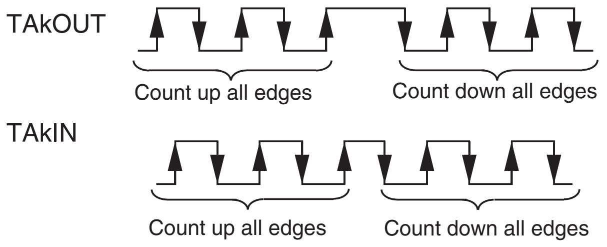
Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 1 : Event counter mode (1)	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (TAiOUT pin functions as pulse output pin)	RW
MR1	Count polarity select bit (2)	0 : Counts falling edge of external signal 1 : Counts rising edge of external signal	RW
MR2	Up/down switching source select bit	0 : UDF register 1 : Input signal to TAiOUT pin (3)	RW
MR3	Set to 0 in event counter mode		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Can be 0 or 1 when not using two-phase pulse signal processing.		RW

NOTES:

1. During event counter mode, the count source can be selected using registers ONSF and TRGSR.
2. Effective when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are 00b (TAiIN pin input).
3. Count down when input on TAiOUT pin is low or count up when input on that pin is high. The port direction bit for TAiOUT pin is set to 0 (input mode).

Figure 13.8 Registers TA0MR to TA4MR in Event Counter Mode (when not using two-phase pulse signal processing)

Table 13.3 Event Counter Mode Specifications (when using two-phase pulse signal processing with timers A2, A3, and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN or TAiOUT pins
Count operation	• Up-count or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide ratio	1/ (FFFFh - n + 1) for up-count 1/ (n + 1) for down-count n : set value of the TAi register 0000h to FFFFh
Count start condition	Set the TAiS bit in the TABSR register to 1 (count starts)
Count stop condition	Set the TAiS bit to 0 (count stops)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read by reading the TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to reload register (Transferred to counter when reloaded next)
Select function [(1)]	• Normal processing operation (timers A2 and A3) The timer counts up rising edges or counts down falling edges on TAjIN pin when input signals on TAjOUT pin is "H".  TAjOUT TAjIN Up-count Up-count Up-count Down-count Down-count Down-count • Multiply-by-4 processing operation (timers A3 and A4) If the phase relationship is such that TAkIN pin goes "H" when the input signal on TAkOUT pin is "H", the timer counts up rising and falling edges on pins TAkOUT and TAkIN. If the phase relationship is such that TAkIN pin goes "L" when the input signal on TAkOUT pin is "H", the timer counts down rising and falling edges on pins TAkOUT and TAkIN.  TAkOUT Count up all edges Count down all edges TAkIN Count up all edges Count down all edges • Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.

i = 2 to 4
j = 2, 3
k = 3, 4

NOTE:

1. Only timer A3 is selectable. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

Timer Ai Mode Register (i = 2 to 4)
(When using two-phase pulse signal processing)

b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	0	1

Symbol	Address	After Reset
TA2MR to TA4MR	0398h to 039Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	To use two-phase pulse signal processing, set this bit to 0.		RW
MR1			RW
MR2	To use two-phase pulse signal processing, set this bit to 1.		RW
MR3	To use two-phase pulse signal processing, set this bit to 0.		RW
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	RW
TCK1	Two-phase pulse signal processing operation select bit [1] [2]	0 : Normal processing operation 1 : Multiply-by-4 processing operation	RW

NOTES:

1. The TCK1 bit is valid for the TA3MR register. No matter how this bit is set, timers A2 and A4 always operate in normal processing mode and x4 processing mode, respectively.
2. If two-phase pulse signal processing is desired, following register settings are required:
   - Set the TAiP bit in the UDF register to 1 (two-phase pulse signal processing function enabled).
   - Set bits TAiTGH and TAiTGL in the TRGSR register to 00b (TAiIN pin input).
   - Set the port direction bits for TAiIN and TAiOUT to 0 (input mode).

Figure 13.9 Registers TA2MR to TA4MR in Event Counter Mode (when using two-phase pulse signal processing with timers A2, A3, and A4)

13.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing

This function initializes the timer count value to 0 by Z-phase (counter initialization) input during two-phase pulse signal processing.

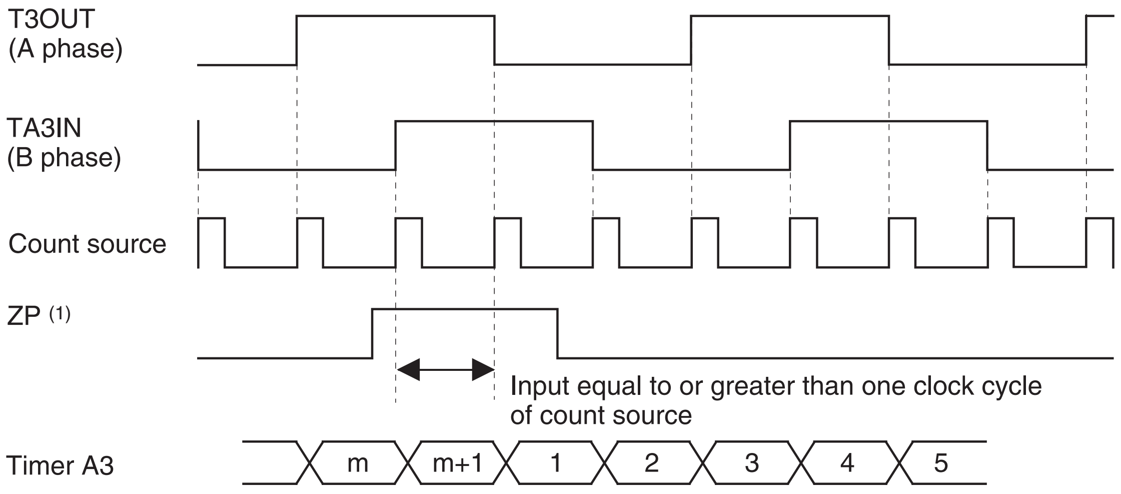
This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the ZP pin.

Counter initialization by Z-phase input is enabled by writing 0000h to the TA3 register and setting the TAZIE bit in the ONSF register to 1 (Z-phase input enabled).

Counter initialization is accomplished by detecting Z-phase input edge. The active edge can be selected to be the rising or falling edge by using the POL bit in the INT2IC register. The Z-phase pulse width applied to the $\overline{\text{INT2}}$ pin must be equal to or greater than one clock cycle of the timer A3 count source.

The counter is initialized at the next count timing after recognizing Z-phase input. Figure 13.10 shows the relationship between the two-phase pulse (A phase and B phase) and the Z-phase.

If timer A3 overflow or underflow coincides with the counter initialization by Z-phase input, a timer A3 interrupt request is generated twice in succession. Do not use the timer A3 interrupt when using this function.



NOTE:

1. This timing diagram is for the case where the POL bit in the INT2IC register = 1 (rising edge).

Figure 13.10 Two-phase Pulse (A Phase and B Phase) and Z Phase

13.1.3 One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. When the trigger occurs, the timer starts up and continues operating for a given period. Table 13.4 lists the One-shot Timer Mode Specifications. Figure 13.11 shows Registers TA0MR to TA4MR in One-shot Timer Mode.

Table 13.4 One-shot Timer Mode Specifications

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	• Down-count • When the counter reaches 0000h, it stops counting after reloading a new value • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : set value of the TAi register 0000h to FFFFh However, the counter does not work if the divide-by-n value is set to 0000h.
Count start condition	The TAiS bit in the TABSR register = 1 (count starts) and one of the following triggers occurs. • External trigger input from the TAiIN pin • Timer B2 overflow or underflow, Timer Aj overflow or underflow, Timer Ak overflow or underflow • The TAiOS bit in the ONSF register is set to 1 (timer starts)
Count stop condition	• When the counter is reloaded after reaching 0000h • TAiS bit is set to 0 (count stops)
Interrupt request generation timing	When the counter reaches 0000h
TAiIN pin function	I/O port or trigger input
TAiOUT pin function	I/O port or pulse output
Read from timer	An undefined value is read by reading the TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to the TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Pulse output function The timer outputs a low when not counting and a high when counting.

i = 0 to 4
j = i - 1, except j = 4 if i = 0
k = i + 1, except k = 0 if i = 4

Timer Ai Mode Register (i = 0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol	Address	After Reset
TA0MR to TA4MR	0396h to 039Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 1 0 : One-shot timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin functions as I/O port) 1 : Pulse is output (TAiOUT pin functions as a pulse output pin)	RW
MR1	External trigger select bit (1)	0 : Falling edge of input signal to TAiIN pin (2) 1 : Rising edge of input signal to TAiIN pin (2)	RW
MR2	Trigger select bit	0 : TAiOS bit is enabled 1 : Selected by bits TAiTGH to TAiTGL	RW
MR3	Set to 0 in one-shot timer mode		RW
TCK0	Count source select bits	b7 b6 0 0 : f1 or f2 (3) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES:
1. Effective when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are 00b (TAiIN pin input).
2. The port direction bit for the TAiIN pin is set to 0 (input mode).
3. Selected by the PCLK0 bit in the PCLKR register.

Figure 13.11 Registers TA0MR to TA4MR in One-shot Timer Mode

13.1.4 Pulse Width Modulation (PWM) Mode

In Pulse Width Modulation mode, the timer outputs pulses of a given width in succession. The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator.

Table 13.5 lists the Pulse Width Modulation Mode Specifications. Figure 13.12 shows Registers TA0MR to TA4MR in Pulse Width Modulation Mode. Figures 13.13 and 13.14 show an Example of 16-bit Pulse Width Modulator Operation and 8-bit Pulse Width Modulator Operation.

Table 13.5 Pulse Width Modulation Mode Specifications

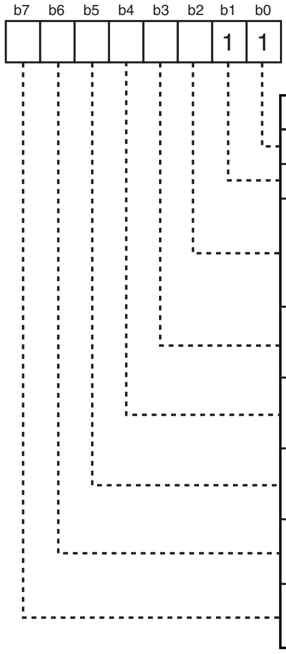
Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	• Down-count (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	• High level width n / fj n : set value of the TAi register • Cycle time $(2^{16}-1)$ / fj fixed fj : count source frequency (f1, f2, f8, f32, fC32)
8-bit PWM	• High level width n × (m+1) / fj n : set value of the TAi register high-order address • Cycle time $(2^{8}-1)$ × (m+1) / fj m : set value of the TAi register low-order address
Count start condition	• The TAiS bit in the TABSR register is set to 1 (count starts) • The TAiS bit = 1 and external trigger input from the TAiIN pin • The TAiS bit = 1 and one of the following external triggers occurs Timer B2 overflow or underflow, Timer Aj overflow or underflow, Timer Ak overflow or underflow
Count stop condition	The TAiS bit is set to 0 (count stops)
Interrupt request generation timing	On the falling edge of the PWM pulse
TAiIN pin function	I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	An undefined value is read by reading the TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to the TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TAi register is written to only reload register (Transferred to counter when reloaded next)

i = 0 to 4

j = i - 1, except j = 4 if i = 0

k = i + 1, except k = 0 if i = 4

Timer Ai Mode Register (i = 0 to 4)

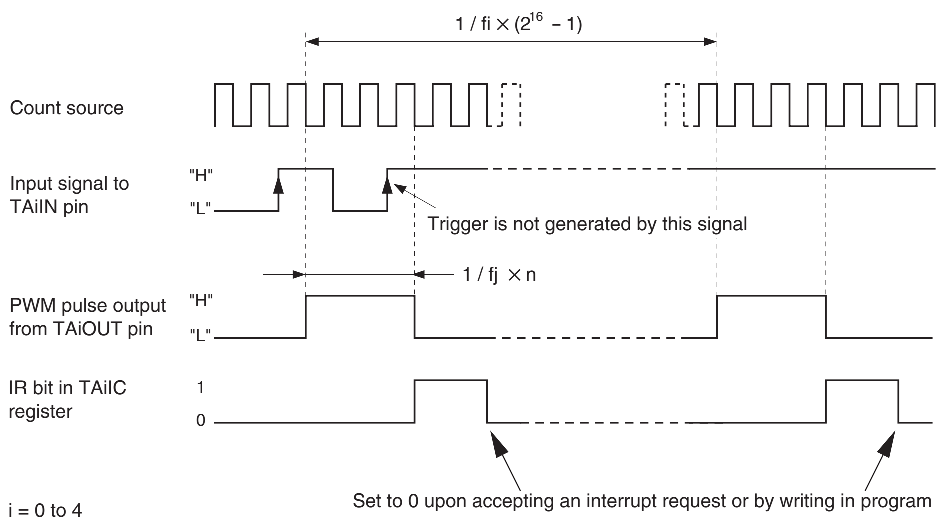


Symbol	Address	After Reset
TA0MR to TA4MR	0396h to 039Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 1 1 : Pulse width modulation mode	RW
TMOD1			RW
MR0	Pulse output function select bit (3)	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (TAiOUT pin is a pulse output pin)	RW
MR1	External trigger select bit (1)	0 : Falling edge of input signal to TAiIN pin (2) 1 : Rising edge of input signal to TAiIN pin (2)	RW
MR2	Trigger select bit	0 : Write 1 to TAiS bit in the TABSR register 1 : Selected by bits TAiTGH to TAiTGL	RW
MR3	16/8-Bit PWM mode select bit	0 : Functions as a 16-bit pulse width modulator 1 : Functions as an 8-bit pulse width modulator	RW
TCK0	Count source select bits	b7 b6 0 0 : f1 or f2 (4) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES:
1. Effective when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are 00b (TAiIN pin input).
2. The port direction bit for the TAiIN pin is set to 0 (input mode).
3. Set to 1 (pulse is output), PWM pulse is output.
4. Selected by the PCLK0 bit in the PCLKR register.

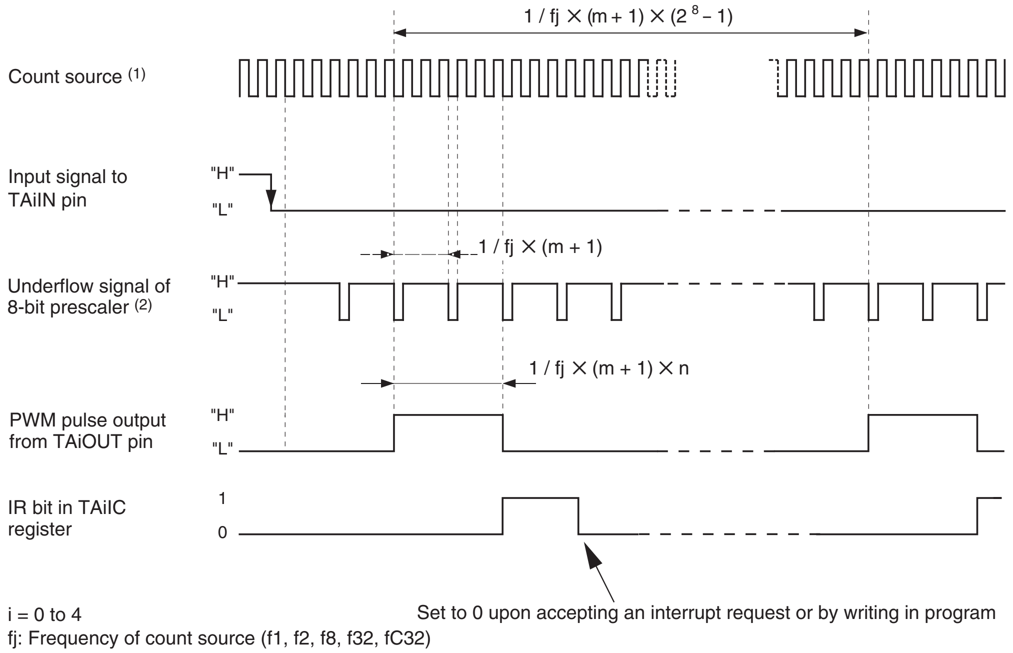
Figure 13.12 Registers TA0MR to TA4MR in Pulse Width Modulation Mode


1 / fi × (2^{16} – 1)
Count source
Input signal to TAiIN pin
"H"
"L"
Trigger is not generated by this signal
1 / fj × n
PWM pulse output from TAiOUT pin
"H"
"L"
IR bit in TAiIC register
1
0
Set to 0 upon accepting an interrupt request or by writing in program

i = 0 to 4
fj: Frequency of count source (f1, f2, f8, f32, fC32)

NOTES:
1. n = 0000h to FFFEh.
2. This timing diagram is the following case.
   - TAi register = 0003h
   - Bits TAiTGH and TAiTGL in the ONSF or TRGSR register = 00b (TAiIN pin input)
   - The MR1 bit in the TAiMR register = 1 (rising edge)
   - The MR2 bit in the TAiMR register = 1 (trigger selected by bits TAiTGH and TAiTGL)

Figure 13.13 Example of 16-bit Pulse Width Modulator Operation


1 / fj × (m + 1) × (2^8 – 1)
Count source (1)
Input signal to TAiIN pin
"H"
"L"
1 / fj × (m + 1)
Underflow signal of 8-bit prescaler (2)
"H"
"L"
1 / fj × (m + 1) × n
PWM pulse output from TAiOUT pin
"H"
"L"
IR bit in TAiIC register
1
0
Set to 0 upon accepting an interrupt request or by writing in program

i = 0 to 4
fj: Frequency of count source (f1, f2, f8, f32, fC32)

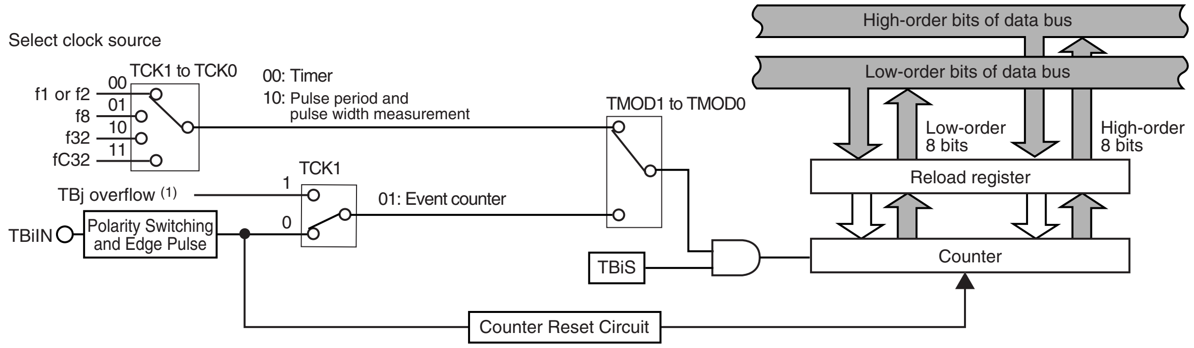
NOTES:
1. The 8-bit prescaler counts the count source.
2. The 8-bit pulse width modulator counts the output from the 8-bit prescaler underflow signal.
3. m = 00h to FFh; n = 00h to FEh.
4. This timing diagram is the following case.
   - TAi register = 0202h
   - Bits TAiTGH and TAiTGL in the ONSF or TRGSR register = 00b (TAiIN pin input)
   - The MR1 bit in the TAiMR register = 0 (falling edge)
   - The MR2 bit in the TAiMR register = 1 (trigger selected by bits TAiTGH and TAiTGL)

Figure 13.14 Example of 8-bit Pulse Width Modulator Operation

13.2 Timer B

Figure 13.15 shows a Timer B Block Diagram. Figures 13.16 and 13.17 show the timer B-related registers. Timer B supports the following three modes. Use bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 5) to select the desired mode.

- Timer mode : The timer counts an internal count source.
- Event counter mode : The timer counts pulses from an external device or over flows or underflows of other timers.
- Pulse period/pulse width measuring mode : The timer measures pulse period or pulse width of an external signal.



TCK1 to TCK0, TMOD1 to TMOD0: Bits in TBiMR register
TBiS: Bit in TABSR register or TBSR register

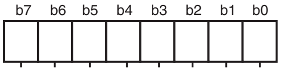
i = 0 to 5
j = i - 1 except j = 2 when i = 0, j = 5 when i = 3

NOTE:
1. Overflow or underflow

TBi	Addresses	TBj
Timer B0	0391h - 0390h	Timer B2
Timer B1	0393h - 0392h	Timer B0
Timer B2	0395h- 0394h	Timer B1
Timer B3	01D1h- 01D0h	Timer B5
Timer B4	01D3h- 01D2h	Timer B3
Timer B5	01D5h- 01D4h	Timer B4

Figure 13.15 Timer B Block Diagram

Timer Bi Mode Register (i = 0 to 5)



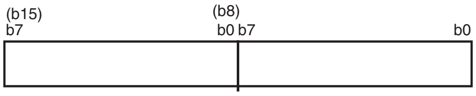
Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b
TB3MR to TB5MR	01DBh to 01DDh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period measurement mode, pulse width measurement mode 1 1 : Do not set a value	RW
TMOD1			RW
MR0	Function varies depending on operating mode		RW
MR1			RW
MR2			RW[1] –[2]
MR3			RO
TCK0	Count source select bits	Function varies depending on operating mode	RW
TCK1			RW

NOTES:

1. Timers B0 and B3.
2. Timers B1, B2, B4, and B5.

Timer Bi Register (i = 0 to 5)[1]



Symbol	Address	After Reset
TB0	0391h, 0390h	Undefined
TB1	0393h, 0392h	Undefined
TB2	0395h, 0394h	Undefined
TB3	01D1h, 01D0h	Undefined
TB4	01D3h, 01D2h	Undefined
TB5	01D5h, 01D4h	Undefined

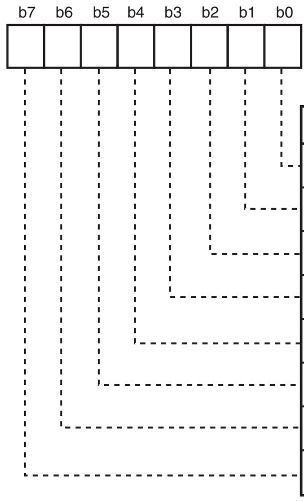
Mode	Function	Setting Range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000h to FFFFh	RW
Event counter mode	Divide the count source by n + 1 where n = set value[2]	0000h to FFFFh	RW
Pulse period modulation mode, Pulse width modulation mode	Measures a pulse period or width	——	RO

NOTES:

1. The register must be accessed in 16-bit unit.
2. The timer counts pulses from an external device or overflows or underflows of other timers.

Figure 13.16 Registers TB0MR to TB5MR, and TB0 to TB5

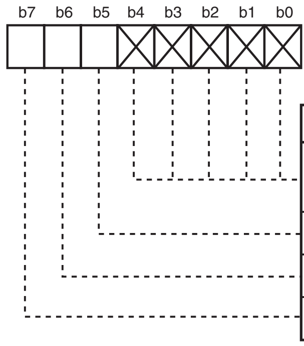
Count Start Flag



Symbol	Address	After Reset
TABSR	0380h	00h

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 count start flag	0 : Count stops 1 : Count starts	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

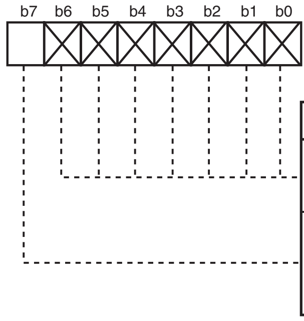
Timer B3, B4, B5 Count Start Flag



Symbol	Address	After Reset
TBSR	01C0h	000XXXXXb

Bit Symbol	Bit Name	Function	RW
- (b4-b0)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
TB3S	Timer B3 count start flag	0 : Count stops 1 : Count starts	RW
TB4S	Timer B4 count start flag		RW
TB5S	Timer B5 count start flag		RW

Clock Prescaler Reset Flag



Symbol	Address	After Reset
CPSRF	0381h	0XXXXXXXb

Bit Symbol	Bit Name	Function	RW
- (b6-b0)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
CPSR	Clock prescaler reset flag	Setting this bit to 1 initializes the prescaler for the timekeeping clock. (When read, the content is 0.)	RW

Figure 13.17 Registers TABSR, TBSR, and CPSRF

13.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally.

Table 13.6 lists the Timer Mode Specifications. Figure 13.18 shows Registers TB0MR to TB5MR in Timer Mode.

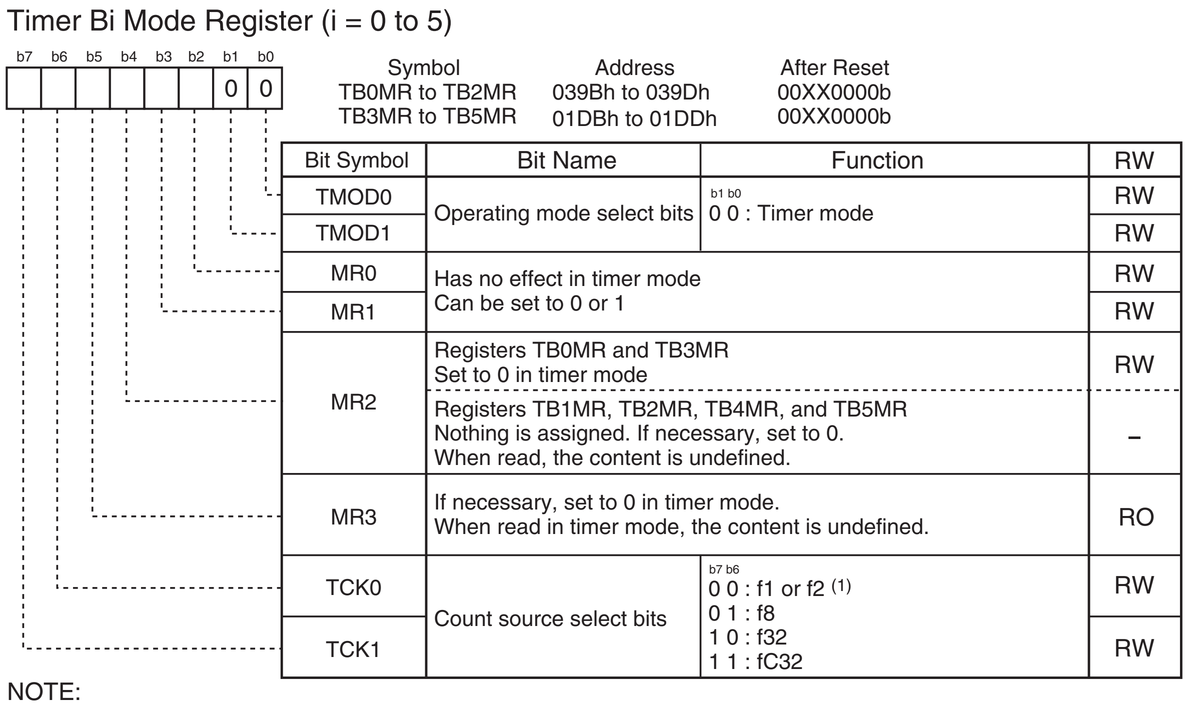
Table 13.6 Timer Mode Specifications

Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of the TBi register 0000h to FFFFh
Count start condition	Set the TBiS bit [1] to 1 (count starts)
Count stop condition	Set the TBiS bit to 0 (count stops)
Interrupt request generation timing	Timer underflow
TBiIN pin function	I/O port
Read from timer	Count value can be read by reading the TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to the TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)

i = 0 to 5

NOTE:

1. Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.



Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b
TB3MR to TB5MR	01DBh to 01DDh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Has no effect in timer mode Can be set to 0 or 1		RW
MR1			RW
MR2	Registers TB0MR and TB3MR Set to 0 in timer mode		RW
	Registers TB1MR, TB2MR, TB4MR, and TB5MR Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
MR3	If necessary, set to 0 in timer mode. When read in timer mode, the content is undefined.		RO
TCK0	Count source select bits	b7 b6 0 0 : f1 or f2 [1] 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTE:

1. Selected by the PCLK0 bit in the PCLKR register.

Figure 13.18 Registers TB0MR to TB5MR in Timer Mode

13.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Table 13.7 lists the Event Counter Mode Specifications. Figure 13.19 shows Registers TB0MR to TB5MR in Event Counter Mode.

Table 13.7 Event Counter Mode Specifications

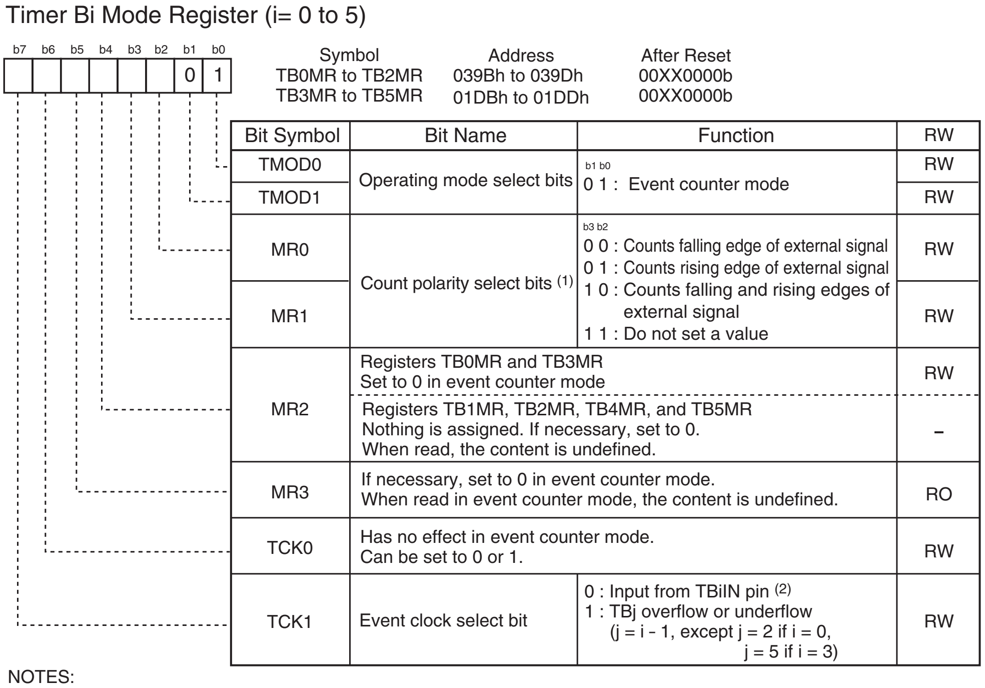
Item	Specification
Count source	• External signals input to TBiIN pin (effective edge can be selected in program) • Timer Bj overflow or underflow
Count operation	• Down-count • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of the TBi register 0000h to FFFFh
Count start condition	Set TBiS bit [1] to 1 (count starts)
Count stop condition	Set TBiS bit to 0 (count stops)
Interrupt request generation timing	Timer underflow
TBiIN pin function	Count source input
Read from timer	Count value can be read by reading the TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to the TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to the TBi register is written to only reload register (Transferred to counter when reloaded next)

i = 0 to 5

j = i - 1, except j = 2 if i = 0, j = 5 if i = 3

NOTE:

1. Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.



Timer Bi Mode Register (i= 0 to 5)

Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b
TB3MR to TB5MR	01DBh to 01DDh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	Count polarity select bits [1]	b3 b2 0 0 : Counts falling edge of external signal 0 1 : Counts rising edge of external signal 1 0 : Counts falling and rising edges of external signal 1 1 : Do not set a value	RW
MR1			RW
MR2	Registers TB0MR and TB3MR Set to 0 in event counter mode		RW
	Registers TB1MR, TB2MR, TB4MR, and TB5MR Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
MR3	If necessary, set to 0 in event counter mode. When read in event counter mode, the content is undefined.		RO
TCK0	Has no effect in event counter mode. Can be set to 0 or 1.		RW
TCK1	Event clock select bit	0 : Input from TBiIN pin [2] 1 : TBj overflow or underflow (j = i - 1, except j = 2 if i = 0, j = 5 if i = 3)	RW

NOTES:

1. Effective when the TCK1 bit = 0 (input from TBiIN pin). If the TCK1 bit = 1 (TBj overflow or underflow), these bits can be set to 0 or 1.
2. The port direction bit for the TBiIN pin is set to 0 (input mode).

Figure 13.19 Registers TB0MR to TB5MR in Event Counter Mode

13.2.3 Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal. Table 13.8 lists the Pulse Period and Pulse Width Measurement Mode Specifications. Figure 13.20 shows Registers TB0MR to TB5MR in Pulse Period and Pulse Width Measurement mode. Figure 13.21 shows the Operation Timing when Measuring Pulse Period. Figure 13.22 shows the Operation Timing when Measuring Pulse Width.

Table 13.8 Pulse Period and Pulse Width Measurement Mode Specifications

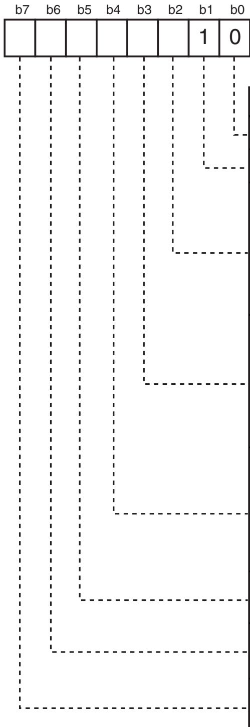
Item	Specification
Count source	f1, f2, f8, f32, fC32
Count operation	• Up-count • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to 0000h to continue counting.
Count start condition	Set the TBiS bit (1) to 1 (count starts)
Count stop condition	Set the TBiS bit to 0 (count stops)
Interrupt request generation timing	• When an effective edge of measurement pulse is input (2) • Timer overflow. If an overflow occurs, the MR3 bit in the TBiMR register is set to 1 (overflow) simultaneously. The MR3 bit is set to 0 (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to 1. At this time, make sure the TBiS bit is set to 1 (count starts).
TBiIN pin function	Measurement pulse input
Read from timer	Contents of the reload register (measurement result) can be read by reading TBi register (3)
Write to timer	Value written to the TBi register is written to neither reload register nor counter

i = 0 to 5

NOTES:

1. Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.
2. Interrupt request is not generated when the first effective edge is input after the timer started counting.
3. Value read from the TBi register is undefined until the second valid edge is input after the timer starts counting.

Timer Bi Mode Register (i = 0 to 5)



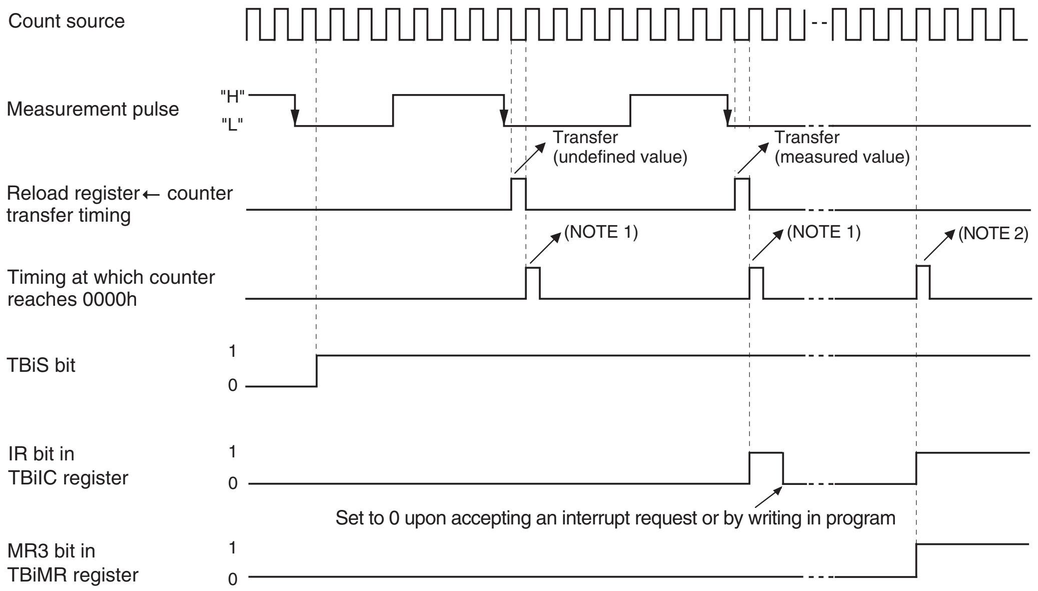
Symbol	Address	After Reset
TB0MR to TB2MR	039Bh to 039Dh	00XX0000b
TB3MR to TB5MR	01DBh to 01DDh	00XX0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 1 0 : Pulse period / pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement mode select bits	b3 b2 0 0 : Pulse period measurement (Measurement between a falling edge and the next falling edge of measured pulse) 0 1 : Pulse period measurement (Measurement between a rising edge and the next rising edge of measured pulse) 1 0 : Pulse width measurement (Measurement between a falling edge and the next rising edge of measured pulse and between a rising edge and the next falling edge) 1 1 : Do not set a value	RW
MR1			RW
MR2	Registers TB0MR and TB3MR Set to 0 in pulse period and pulse width measurement mode		RW
	Registers TB1MR, TB2MR, TB4MR, and TB5MR Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
MR3	Timer Bi overflow flag (1)	0 : Timer did not overflow 1 : Timer has overflowed	RO
TCK0	Count source select bits	b7 b6 0 0 : f1 or f2 (2) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTES:

1. This flag is undefined after reset. When the TBiS bit = 1 (count starts), the MR3 bit is set to 0 (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to 1 (overflow). The MR3 bit cannot be set to 1 in a program. Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.
2. Selected by the PCLK0 bit in the PCLKR register.

Figure 13.20 Registers TB0MR to TB5MR in Pulse Period and Pulse Width Measurement Mode

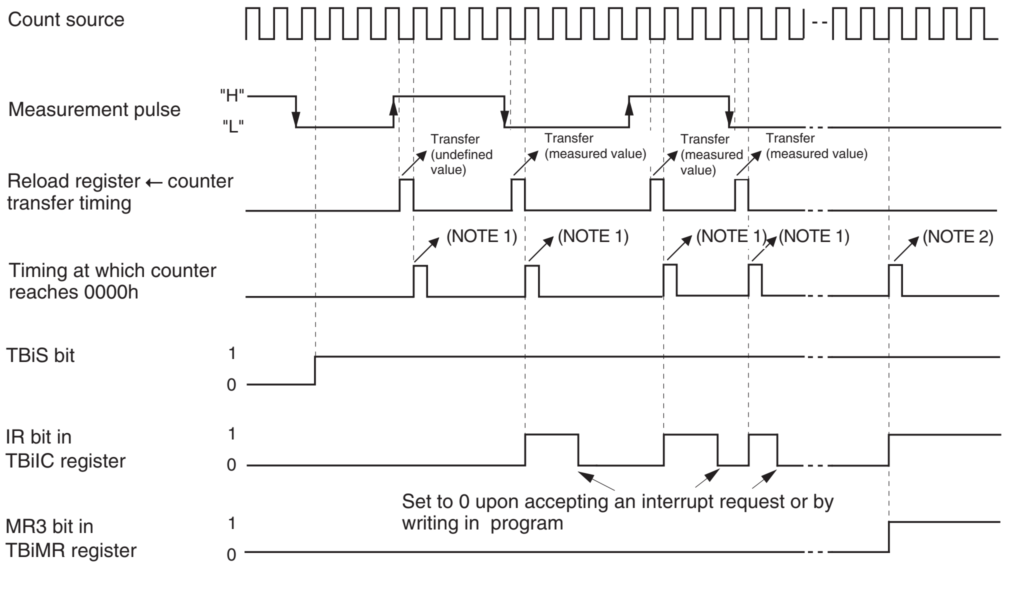

Count source
Measurement pulse
"H"
"L"
Transfer (undefined value)
Transfer (measured value)
Reload register ← counter transfer timing
(NOTE 1)
(NOTE 1)
(NOTE 2)
Timing at which counter reaches 0000h
TBiS bit
1
0
IR bit in TBiIC register
1
0
Set to 0 upon accepting an interrupt request or by writing in program
MR3 bit in TBiMR register
1
0

Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.

i = 0 to 5

NOTES:
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where bits MR1 to MR0 in the TBiMR register are 00b (measure the interval from falling edge to falling edge of the measurement pulse).

Figure 13.21 Operation Timing When Measuring Pulse Period


Count source
Measurement pulse
"H"
"L"
Transfer (undefined value)
Transfer (measured value)
Transfer (measured value)
Transfer (measured value)
Reload register ← counter transfer timing
(NOTE 1)
(NOTE 1)
(NOTE 1)
(NOTE 1)
(NOTE 2)
Timing at which counter reaches 0000h
TBiS bit
1
0
IR bit in TBiIC register
1
0
Set to 0 upon accepting an interrupt request or by writing in program
MR3 bit in TBiMR register
1
0

Bits TB0S to TB2S are assigned to bits 5 to 7 in the TABSR register, and bits TB3S to TB5S are assigned to bits 5 to 7 in the TBSR register.

i = 0 to 5

NOTES:
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where the MR1 to MR0 bits in the TBiMR register are 10b (measure the interval from a falling edge to the next rising edge and the interval from a rising edge to the next falling edge of the measurement pulse).

Figure 13.22 Operation Timing When Measuring Pulse Width

14. Three-Phase Motor Control Timer Function

Timers A1, A2, A4, and B2 can be used to output three-phase motor drive waveforms. Table 14.1 lists the Three-phase Motor Control Timer Function Specifications. Figure 14.1 shows the Three-phase Motor Control Timer Function Block Diagram. Figures 14.2 to 14.8 shows the Three-phase Motor Control Timer Function related registers.

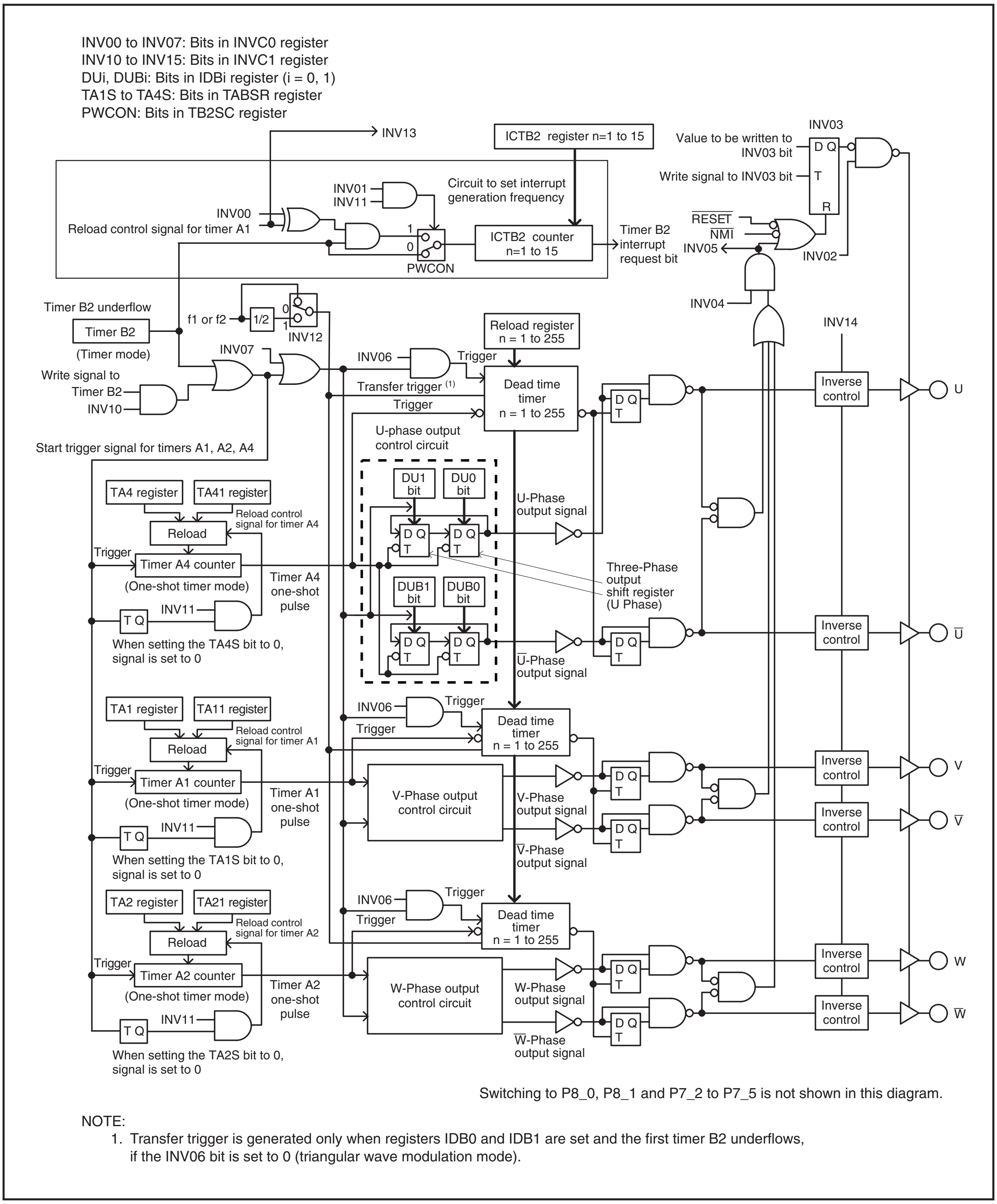
Table 14.1 Three-Phase Motor Control Timer Function Specifications

Item	Specification
Three-Phase waveform output pin	Six pins (U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$)
Forced cutoff input [1]	Input “L” to $\overline{NMI}$ pin
Used timers	Timer A4, A1, A2 (used in the one-shot timer mode) • Timer A4: U- and $\overline{U}$-phase waveform control • Timer A1: V- and $\overline{V}$-phase waveform control • Timer A2: W- and $\overline{W}$-phase waveform control Timer B2 (used in the timer mode) • Carrier wave cycle control Dead time timer (3 eight-bit timer and shared reload register) • Dead time control
Output waveform	Triangular wave modulation, Sawtooth wave modification • Enable to output “H” or “L” for one cycle • Enable to set positive-phase level and negative-phase level respectively
Carrier wave cycle	Triangular wave modulation: count source × (m+1) × 2 Sawtooth wave modulation: count source × (m+1) m: Setting value of the TB2 register, 0000h to FFFFh Count source: f1, f2, f8, f32, fC32
Three-Phase PWM output width	Triangular wave modulation: count source × n × 2 Sawtooth wave modulation: count source × n n: Setting value of registers TA4, TA1, and TA2 (of registers TA4, TA41, TA1, TA11, TA2, and TA21 when setting the INV11 bit to 1), 0001h to FFFFh Count source: f1, f2, f8, f32, fC32
Dead time	Count source × p, or no dead time p: Setting value of the DTT register, 01h to FFh Count source: f1, f2, f1 divided by 2, f2 divided by 2
Active level	Enable to select “H” or “L”
Positive and negative-phase concurrent active disable function	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function
Interrupt frequency	For timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis

NOTE:

1. Forced cutoff with $\overline{NMI}$ input is effective when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by $\overline{NMI}$ input enabled). If an “L” signal is applied to the $\overline{NMI}$ pin when the IVPCR1 bit is 1, the related pins go to a high-impedance state regardless of which functions of those pins are being used.

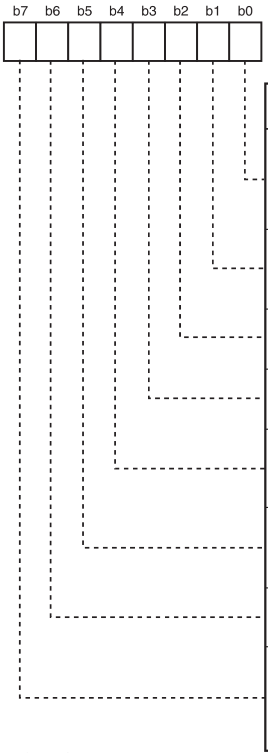
Related pins:
- P7_2/CLK2/TA1OUT/V
- P7_3/$\overline{CTS2}$/$\overline{RTS2}$/TA1IN/$\overline{V}$
- P7_4/TA2OUT/W/(CLK4)
- P7_5/TA2IN/$\overline{W}$/(SOUT4)
- P8_0/TA4OUT/U(SIN4)
- P8_1/TA4IN/$\overline{U}$


INV00 to INV07: Bits in INVC0 register
INV10 to INV15: Bits in INVC1 register
DUi, DUBi: Bits in IDBi register (i = 0, 1)
TA1S to TA4S: Bits in TABSR register
PWCON: Bits in TB2SC register
INV13
ICTB2 register n=1 to 15
INV03
Value to be written to INV03 bit
Write signal to INV03 bit
D Q
T
R
INV01
INV11
Circuit to set interrupt generation frequency
INV00
Reload control signal for timer A1
1
0
PWCON
ICTB2 counter n=1 to 15
Timer B2 interrupt request bit
RESET
NMI
INV05
INV02
INV04
Timer B2 underflow
f1 or f2
1/2
0
1
INV12
Timer B2
(Timer mode)
Write signal to Timer B2
INV10
INV07
INV06
Trigger
Reload register n = 1 to 255
INV14
Transfer trigger (1)
Dead time timer n = 1 to 255
Trigger
U-phase output control circuit
Inverse control
U
Start trigger signal for timers A1, A2, A4
TA4 register
TA41 register
DU1 bit
DU0 bit
Reload control signal for timer A4
Reload
U-Phase output signal
Trigger
Timer A4 counter
(One-shot timer mode)
Timer A4 one-shot pulse
DUB1 bit
DUB0 bit
Three-Phase output shift register (U Phase)
INV11
T Q
When setting the TA4S bit to 0, signal is set to 0
Ū
Ū-Phase output signal
TA1 register
TA11 register
INV06
Trigger
Reload control signal for timer A1
Dead time timer n = 1 to 255
Reload
Trigger
Timer A1 counter
(One-shot timer mode)
V-Phase output control circuit
V-Phase output signal
V
V̄
Timer A1 one-shot pulse
INV11
When setting the TA1S bit to 0, signal is set to 0
V̄-Phase output signal
TA2 register
TA21 register
Reload control signal for timer A2
Reload
Timer A2 counter
(One-shot timer mode)
Timer A2 one-shot pulse
W-Phase output control circuit
W-Phase output signal
W
W̄
W̄-Phase output signal
INV11
When setting the TA2S bit to 0, signal is set to 0
Switching to P8_0, P8_1 and P7_2 to P7_5 is not shown in this diagram.

NOTE:
1. Transfer trigger is generated only when registers IDB0 and IDB1 are set and the first timer B2 underflows, if the INV06 bit is set to 0 (triangular wave modulation mode).

Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram

Three-Phase PWM Control Register 0 (1)



Symbol	Address	After Reset
INVC0	01C8h	00h

Bit Symbol	Bit Name	Function	RW
INV00	Interrupt enable output polarity select bit	0: The ICTB2 counter is incremented by one on the rising edge of the timer A1 reload control signal 1: The ICTB2 counter is incremented by one on the falling edge of the timer A1 reload control signal (2)	RW
INV01	Interrupt enable output specification bit (3)	0: ICTB2 counter is incremented by one when timer B2 underflows 1: Selected by the INV00 bit (2)	RW
INV02	Mode select bit (4)	0: No three-phase control timer functions 1: Three-phase control timer function (5)	RW
INV03	Output control bit	0: Three-phase control timer output disabled (5) 1: Three-phase control timer output enabled (6)	RW
INV04	Positive and negative-phases concurrent active disable function enable bit	0: Concurrent active output enabled 1: Concurrent active output disabled	RW
INV05	Positive and negative-phases concurrent active output detect flag	0: Not detected 1: Detected (7)	RW
INV06	Modulation mode select bit (8)	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode (9)	RW
INV07	Software trigger select bit	Transfer trigger is generated when the INV07 bit is set to 1. Trigger to the dead time timer is also generated when setting the INV06 bit to 1. Its value is 0 when read.	RW

NOTES:

1. Set the INVC0 register after the PRC1 bit in the PRCR register is set to 1 (write enabled). Rewrite bits INV00 to INV02, and INV06 when the timers A1, A2, A4, and B2 stop.
2. Bits INV00 and INV01 are enabled only when the INV11 bit is set to 1 (three-phase mode 1). The ICTB2 counter is incremented by one every time the timer B2 underflows, regardless of INV00 and INV01 bit settings, when the INV11 bit is set to 0 (three-phase mode 0).
   When setting the INV01 bit to 1, set the timer A1 count start flag before the first timer B2 underflow.
   When the INV00 bit is set to 1, the first interrupt is generated when the timer B2 underflows *n-1* times, if *n* is the value set in the ICTB2 counter. Subsequent interrupts are generated every *n* times the timer B2 underflows.
3. Set the INV01 bit to 1 after setting the ICTB2 register .
4. Set the INV02 bit to 1 to operate the dead time timer, U-, V-, and W-phase output control circuits and ICTB2 counter.
5. When the INV03 bit is set to 1, the pins applied to U/V/W output three-phase PWM.
   Pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$, including pins shared with other output functions, are all placed in high-impedance states when the following conditions are all met.
   - The INV02 bit is set to 1 (three-phase control timer function)
   - The INV03 bit is set to 0 (three-phase control timer output disabled)
   - Direction registers of each port are set to 0 (input mode)
6. The INV03 bit is set to 0 when the following conditions are all met.
   - Reset
   - A concurrent active state occurs while INV04 bit is set to 1
   - The INV03 bit is set to 0 by program
   - A signal applied to the $\overline{NMI}$ pin changes "H" to "L"

   When both the INV04 and INV05 bits are set to 1, the INV03 bit is set to 0.
7. The INV05 bit cannot be set to 1 by program. Set the INV04 bit to 0, as well, when setting the INV05 bit to 0.
8. The following table describes how the INV06 bit works.

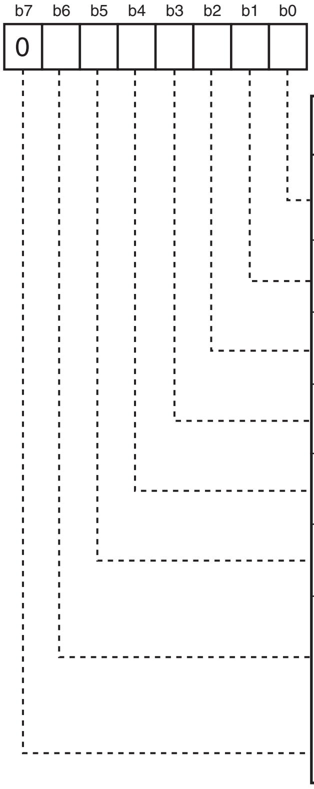
Item	INV06 = 0	INV06 = 1
Mode	Triangular wave modulation mode	Sawtooth wave modulation mode
Timing to transfer from registers IDB0 and IDB1 to three-phase output shift register	Transferred once by generating a transfer trigger after setting registers IDB0 and IDB1	Transferred every time a transfer trigger is generated
Timing to trigger the dead time timer when the INV16 bit=0	On the falling edge of a one-shot pulse of timer A1, A2, or A4	By a transfer trigger, or the falling edge of a one-shot pulse of timer A1, A2, or A4
INV13 bit	Enabled when the INV11 bit=1 and the INV06 bit=0	Disabled

Transfer trigger : Timer B2 underflows and write to the INV07 bit, or write to the TB2 register when INV10 = 1

9. When the INV06 bit is set to 1, set the INV11 bit to 0 (three-phase mode 0) and the PWCON bit in the TB2SC register to 0 (reload timer B2 with timer B2 underflow).

Figure 14.2 INVC0 Register

Three-Phase PWM Control Register 1[1]



Symbol	Address	After Reset
INVC1	01C9h	00h

Bit Symbol	Bit Name	Function	RW
INV10	Timers A1, A2, and A4 start trigger select bit	0: Timer B2 underflow 1: Timer B2 underflow and write to timer B2	RW
INV11	Timer A1-1, A2-1, A4-1 control bit [2]	0: Three-phase mode 0 [3] 1: Three-phase mode 1	RW
INV12	Dead time timer count source select bit	0 : f1 or f2 [6] 1 : f1 divided-by-2 or f2 divided-by-2	RW
INV13	Carrier wave detect flag [4]	0: Timer A1 reload control signal is 0 1: Timer A1 reload control signal is 1	RO
INV14	Output polarity control bit	0 : Active "L" of an output waveform 1 : Active "H" of an output waveform	RW
INV15	Dead time disable bit	0: Dead time enabled 1: Dead time disabled	RW
INV16	Dead time timer trigger select bit	0: Falling edge of a one-shot pulse of timers A1, A2, and A4 [5] 1: Rising edge of the three-phase output shift register (U-, V-, W-phase)	RW
- (b7)	Reserved bit	Set to 0	RW

NOTES:

1. Rewrite the INVC1 register after the PRC1 bit in the PRCR register is set to 1 (write enabled). Timers A1, A2, A4, and B2 must be stopped during rewrite.
2. The following table lists how the INV11 bit works.

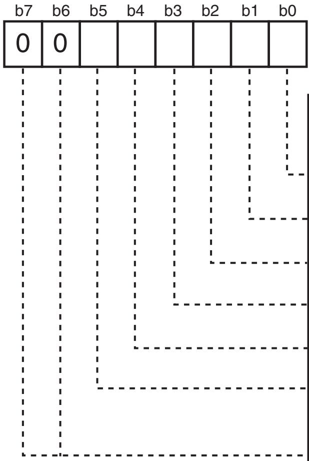
Item	INV11 = 0	INV11 = 1
Mode	Three-phase mode 0	Three-phase mode 1
Registers TA11, TA21, and TA41	Not used	Used
Bits INV00 and INV01	Disabled. The ICTB2 counter is incremented whenever the timer B2 underflows	Enabled
INV13 bit	Disabled	Enabled when INV11=1 and INV06=0

3. When the INV06 bit is set to 1 (sawtooth wave modulation mode), set the INV11 bit to 0 (three-phase mode 0). Also, when the INV11 bit is set to 0, set the PWCON bit in the TB2SC register to 0 (timer B2 is reloaded when the timer B2 underflows).
4. The INV13 bit is enabled only when the INV06 bit is set to 0 (Triangular wave modulation mode) and the INV11 bit to 1 (three-phase mode 1).
5. If the following conditions are all met, set the INV16 bit to 1 (rising edge of the three-phase output shift register).
   - The INV15 bit is set to 0 (dead time timer enabled)
   - The Dij bit (i=U, V or W, j=0, 1) and DiBj bit always have different values when the INV03 bit is set to 1. (The positive-phase and negative-phase always output opposite level signals.)

   If above conditions are not met, set the INV16 bit to 0 (falling edge of a one-shot pulse of timers A1, A2, and A4).
6. Selected by the PCLK0 bit in the PCLKR register.

Figure 14.3 INVC1 Register

Three-Phase Output Buffer Register i (i = 0, 1) (1)



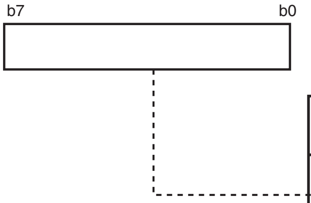
Symbol	Address	After Reset
IDB0, IDB1	01CAh, 01CBh	00111111b

Bit Symbol	Bit Name	Function	RW
DUi	U-phase output buffer i	Write output level 0: Active level 1: Inactive level When read, the value of the three-phase shift register is read.	RW
DUBi	$\overline{U}$-phase output buffer i		RW
DVi	V-phase output buffer i		RW
DVBi	$\overline{V}$-phase output buffer i		RW
DWi	W-phase output buffer i		RW
DWBi	$\overline{W}$-phase output buffer i		RW
- (b7-b6)	Reserved bits	Set to 0	RO

NOTE:

1. The values of registers IDB0 and IDB1 are transferred to the three-phase output shift register by a transfer trigger.
   After the transfer trigger occurs, the values written in the IDB0 register determine each phase output signal first. Then the value written in the IDB1 register on the falling edge of timers A1, A2, and A4 one-shot pulse determines each phase output signal.

Dead Time Timer (1) (2)



Symbol	Address	After Reset
DTT	01CCh	Undefined

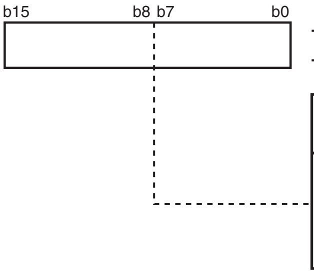
Function	Setting Range	RW
If setting value is *n*, the timer stops when counting *n* times a count source selected by the INV12 bit in the INVC1 register after start trigger occurs. Positive or negative phase, which changes from inactive level to active level, shifts when the dead time timer stops.	1 to 255	WO

NOTES:

1. Use the MOV instruction to set the DTT register.
2. The DTT register is enabled when the INV15 bit in the INVC1 register is set to 0 (dead time enabled). No dead time can be set when the INV15 bit is set to 1 (dead time disabled). The INV06 bit in the INVC0 register determines start trigger of the DTT register.

Figure 14.4 Registers IDB0, IDB1, and DTT

Timer Ai, Ai-1 Register (i = 1, 2, 4) (1) (2) (3) (4) (5) (6)



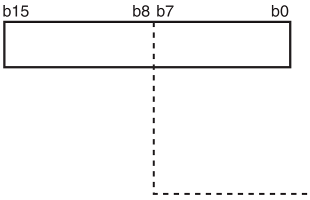
Symbol	Address	After Reset
TA1, TA2, TA4	0389h - 0388h, 038Bh - 038Ah, 038Fh - 038Eh	Undefined
TA11, TA21, TA41 (7)	01C3h - 01C2h, 01C5h - 01C4h, 01C7h - 01C6h	Undefined

Function	Setting Range	RW
If setting value is n, the timer stops when the nth count source is counted after a start trigger is generated. Positive phase changes to negative phase, and vice versa, when timers A1, A2, and A4 stop.	0000h to FFFFh	WO

NOTES:

1. Use a 16-bit data for read and write.
2. If the TAi or TAi1 register is set to 0000h, no counters start and no timer Ai interrupt is generated.
3. Use the MOV instruction to set registers TAi and TAi1.
4. When the INV15 bit in the INVC1 register is set to 0 (dead timer enabled), phase switches from an inactive level to an active level when the dead time timer stops.
5. When the INV11 bit in the INVC1 register is set to 0 (three-phase mode 0), the value of the TAi register is transferred to the reload register by a timer Ai start trigger.
   When the INV11 bit is set to 1 (three-phase mode 1), the value of the TAi1 register is first transferred to the reload register by a timer Ai start trigger. Then, the value of the TAi register is transferred by the next trigger. The values of registers TAi1 and TAi are transferred alternately to the reload register with every timer Ai start trigger.
6. Do not write to these registers when the timer B2 underflows.
7. Follow the procedure below to set the TAi1 register.
   (a) Write value to the TAi1 register,
   (b) Wait one timer Ai count source cycle, and
   (c) Write the same value as (a) to the TAi1 register.

Timer B2 Register (1)



Symbol	Address	After Reset
TB2	0395h - 0394h	Undefined

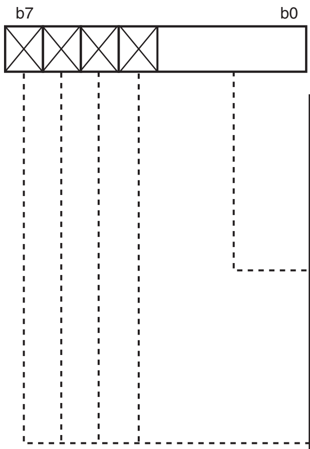
Function	Setting Range	RW
If setting value is *n*, count source is divided by *n*+1. Timers A1, A2, and A4 start every time an underflow occurs.	0000h to FFFFh	RW

NOTE:

1. Use a 16-bit data for read and write.

Figure 14.5 Registers TA1, TA2, TA4, TA11, TA21, TA41, and TB2

Timer B2 Interrupt Generation Frequency Set Counter [1] [2] [3]

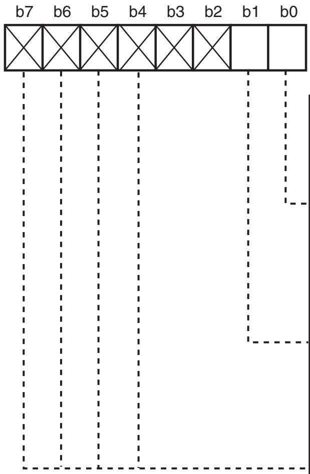


Symbol	Address	After Reset
ICTB2	01CDh	Undefined

Function	Setting Range	RW
When the INV01 bit in the INVC0 register is set to 0 (the ICTB2 counter increments whenever the timer B2 underflows) and the setting value is *n*, the timer B2 interrupt is generated every *n*th time timer B2 underflow occurs. When the INV01 bit is set to 1 (the INV00 bit selects count timing of the ICTB2 counter) and setting value is *n*, the timer B2 interrupt is generated every *n*th time timer B2 underflow meeting the condition selected in the INV00 bit occurs.	1 to 15	WO
Nothing is assigned. If necessary, set to 0.		—

NOTES:
1. Use the MOV instruction to set the ICTB2 register.
2. If the INV01 bit is set to 1, set the ICTB2 register when the TB2S bit is set to 0 (timer B2 count stops), If the INV01 bit is set to 0 and the TB2S bit to 1 (timer B2 count starts), do not set the ICTB2 register when the timer B2 underflows.
3. If the INV00 bit is set to 1, the first interrupt is generated when the timer B2 underflows *n-1* times, *n* being the value set in the ICTB2 counter. Subsequent interrupts are generated every *n* times the timer B2 underflows.

Timer B2 Special Mode Register [1]



Symbol	Address	After Reset
TB2SC	039Eh	XXXXXX00b

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 reload timing switching bit	0 : Timer B2 underflow 1 : Timer A output at odd-numbered occurrences [2]	RW
IVPCR1	Three-phase output port $\overline{\text{NMI}}$ control bit 1 [3]	0 : Three-phase output forcible cutoff by $\overline{\text{NMI}}$ input (high-impedance) disabled 1 : Three-phase output forcible cutoff by $\overline{\text{NMI}}$ input (high-impedance) enabled	RW
- (b7-b2)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		-

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. If the INV11 bit in the INVC1 register is 0 (three-phase mode 0) or the INV06 bit in the INVC0 register is 1 (sawtooth wave modulation mode), set this bit to 0 (timer B2 underflow).
3. Related pins are U(P8_0/TA4OUT/(SIN4)), $\overline{\text{U}}$(P8_1/TA4IN), V(P7_2/CLK2/TA1OUT), $\overline{\text{V}}$(P7_3/$\overline{\text{CTS2}}$/$\overline{\text{RTS2}}$/TA1IN), W(P7_4/TA2OUT/(CLK4)), $\overline{\text{W}}$(P7_5/TA2IN/(SOUT4)).
If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit = 1, the target pins go to a high-impedance state regardless of which functions of those pins are being used.
After forced interrupt (cutoff), input "H" to the $\overline{\text{NMI}}$ pin and set the IVPCR1 bit to 0: this forced cutoff will be reset.

Figure 14.6 Registers ICTB2 and TB2SC

Trigger Select Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TRGSR	0383h	00h

Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 event/trigger select bits	Set to 01b (TB2 underflow) before using a V-phase output control circuit	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bits	Set to 01b (TB2 underflow) before using a W-phase output control circuit	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bits	b5 b4 0 0: Input on TA3IN pin is selected [1] 0 1: TB2 is selected [2] 1 0: TA2 is selected [2] 1 1: TA4 is selected [2]	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bits	Set to 01b (TB2 underflow) before using a U-phase output control circuit	RW
TA4TGH			RW

NOTES:

1. Set the corresponding port direction bit to 0 (input mode).
2. Overflow or underflow.

Count Start Flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TABSR	0380h	00h

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 count start flag	0 : Count stops 1 : Count starts	RW
TA1S	Timer A1 count start flag		RW
TA2S	Timer A2 count start flag		RW
TA3S	Timer A3 count start flag		RW
TA4S	Timer A4 count start flag		RW
TB0S	Timer B0 count start flag		RW
TB1S	Timer B1 count start flag		RW
TB2S	Timer B2 count start flag		RW

Figure 14.7 Registers TRGSR and TRBSR

Timer Ai Mode Register (i = 1, 2, 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	1	0

Symbol	Address	After Reset
TA1MR, TA2MR, TA4MR	0397h, 0398h, 039Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	Set to 10b (one-shot timer mode) when using the three-phase motor control timer function	RW
TMOD1			RW
MR0	Pulse output function select bit	Set to 0 when using the three-phase motor control timer function	RW
MR1	External trigger select bit	Set to 0 when using the three-phase motor control timer function	RW
MR2	Trigger select bit	Set to 1 (selected by the TRGSR register) when using the three-phase motor control timer function	RW
MR3	Set to 0 when using the three-phase motor control timer function		RW
TCK0	Count source select bits	b7 b6 0 0 : f1 or f2 (1) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTE:

1. Selected by the PCLK0 bit in the PCLKR register.

Timer B2 Mode Register

b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

Symbol	Address	After Reset
TB2MR	039Dh	00XX0000b

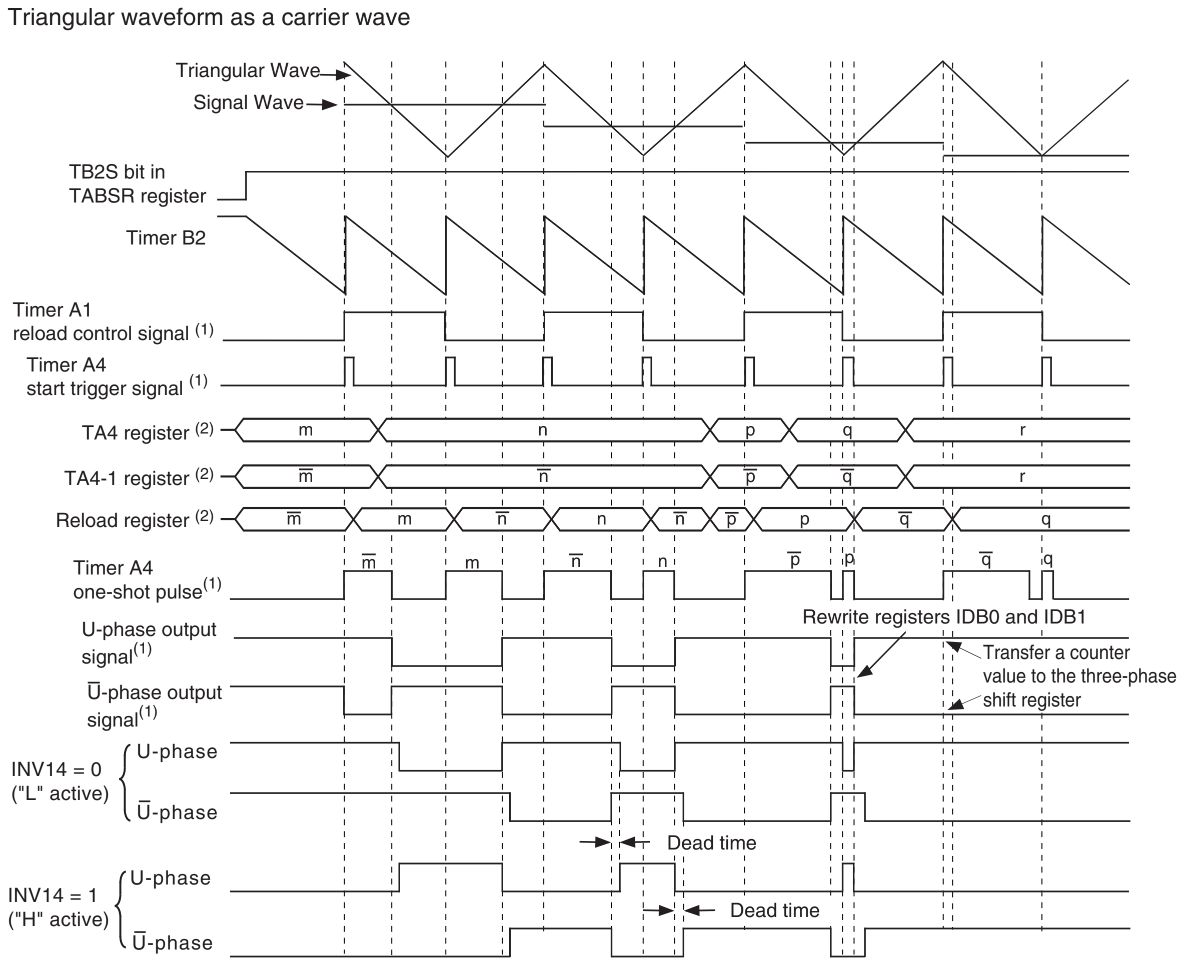
Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	Set to 00b (timer mode) when using the three-phase motor control timer function	RW
TMOD1			RW
MR0	Disabled when using the three-phase motor control timer function. If necessary, set to 0. When read, the content is undefined.		RW
MR1			RW
MR2	Set to 0 when using the three-phase motor control timer function		RW
MR3	If necessary, set to 0 when using the three-phase motor control timer function. When read when using the three-phase motor control timer function, the content is undefined.		RO
TCK0	Count source select bits	b7 b6 0 0 : f1 or f2 (1) 0 1 : f8 1 0 : f32 1 1 : fC32	RW
TCK1			RW

NOTE:

1. Selected by the PCLK0 bit in the PCLKR register.

Figure 14.8 Registers TA1MR, TA2MR, TA4MR, and TB2MR

The three-phase motor control timer function is enabled by setting the INV02 bit in the INVC0 register to 1. When this function is selected, timer B2 is used to control the carrier wave, and timers A4, A1, and A2 are used to control three-phase PWM outputs (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$). The dead time is controlled by a dedicated dead-time timer. Figure 14.9 shows an Example of Triangular Wave Modulation Opertation and Figure 14.10 shows an Example of Sawtooth Wave Modulation Operation.



INV00, INV01: Bits in the INVC0 register
INV11, INV14: Bits in the INVC1 register

NOTES:

1. Internal signals. See **Figure 14.1 Three-Phase Motor Control Timer Functions Block Diagram**.
2. Applies only when the INV11 bit is set to 1 (three-phase mode).

The above applies to INVC0 = 00XX11XXb and INVC1 = 010XXXX0b (X varies depending on each system.)
Examples of PWM output change are

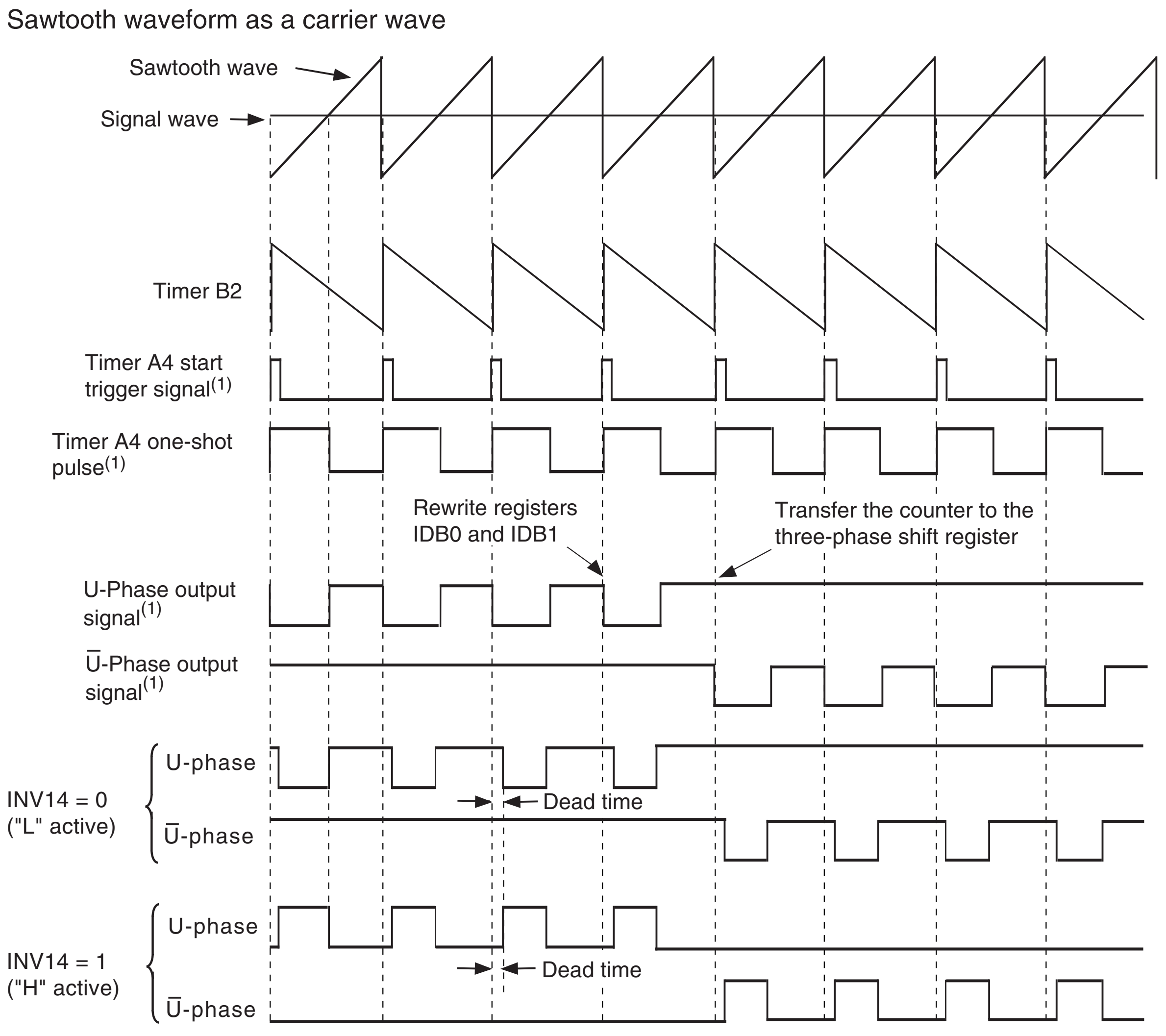
(a) When INV11=1 (three-phase mode 1)

- INV01=0 and ICTB2=2h (The timer B2 interrupt is generated with every second timer B2 underflow) or INV01= 1, INV00=1 and ICTB2=1h (The timer B2 interrupt is generated on the falling edge of the timer A reload control signal)
- Default value of the timer: TA41=$\overline{m}$, TA4=m
  Registers TA4 and TA41 are changed whenever the timer B2 interrupt is generated.
  First time: TA41=$\overline{n}$, TA4=n.
  Second time: TA41=$\overline{p}$, TA4=p.
- Default value of registers IDB0 and IDB1
  DU0=1, DUB0=0, DU1=0, DUB1=1
  They are changed to DU0=1, DUB0=0, DU1=1, DUB1=0 by the third timer B2 interrupt.

(b) When INV11=0 (three-phase mode 0)

- INV01=0, ICTB2=1h (The timer B2 interrupt is generated whenever the timer B2 underflows)
- Default value of the timer: TA4=$\overline{m}$
  The TA4 register is changed whenever the timer B2 interrupt is generated.
  First time: TA4=m. Second time: TA4=$\overline{n}$.
  Third time: TA4=n. Fourth time: TA=$\overline{p}$.
  Fifth time: TA4=p.
- Default value of registers IDB0 and IDB1:
  DU0=1, DUB0=0, DU1=0, DUB1=1
  They are changed to DU0=1, DUB0=0, DU1=1, DUB1=0 by the sixth timer B2 interrupt.

Figure 14.9 Triangular Wave Modulation Operation


Sawtooth waveform as a carrier wave
Sawtooth wave
Signal wave
Timer B2
Timer A4 start trigger signal(1)
Timer A4 one-shot pulse(1)
Rewrite registers IDB0 and IDB1
Transfer the counter to the three-phase shift register
U-Phase output signal(1)
U̅-Phase output signal(1)
INV14 = 0 ("L" active)
U-phase
U̅-phase
Dead time
INV14 = 1 ("H" active)
U-phase
U̅-phase
Dead time

INV14: Bits in the INVC1 register

NOTES:

1. Internal signals. See **Figure 14.1 Three-Phase Motor Control Timer Functions Block Diagram**.

The above applies to INVC0 = 01XX110Xb and INVC1 = 010XXX00b (X varies depending on each system.)
The examples of PWM output change are
- Default value of registers IDB0 and IDB1: DU0=0, DUB0=1, DU1=1, DUB1=1
  They are changed to DU0=1, DUB0=0, DU1=1, DUB1=1 by the timer B2 interrupt.

Figure 14.10 Sawtooth Wave Modulation Operation

15. Serial Interface

Serial interface is configured with 7 channels: UART0 to UART2 and SI/O3 to SI/O6 [1].

NOTE:

1. 100-pin version supports 5 channels; UART0 to UART2, SI/O3, SI/O4
   128-pin version supports 7 channels; UART0 to UART2, SI/O3 to SI/O6

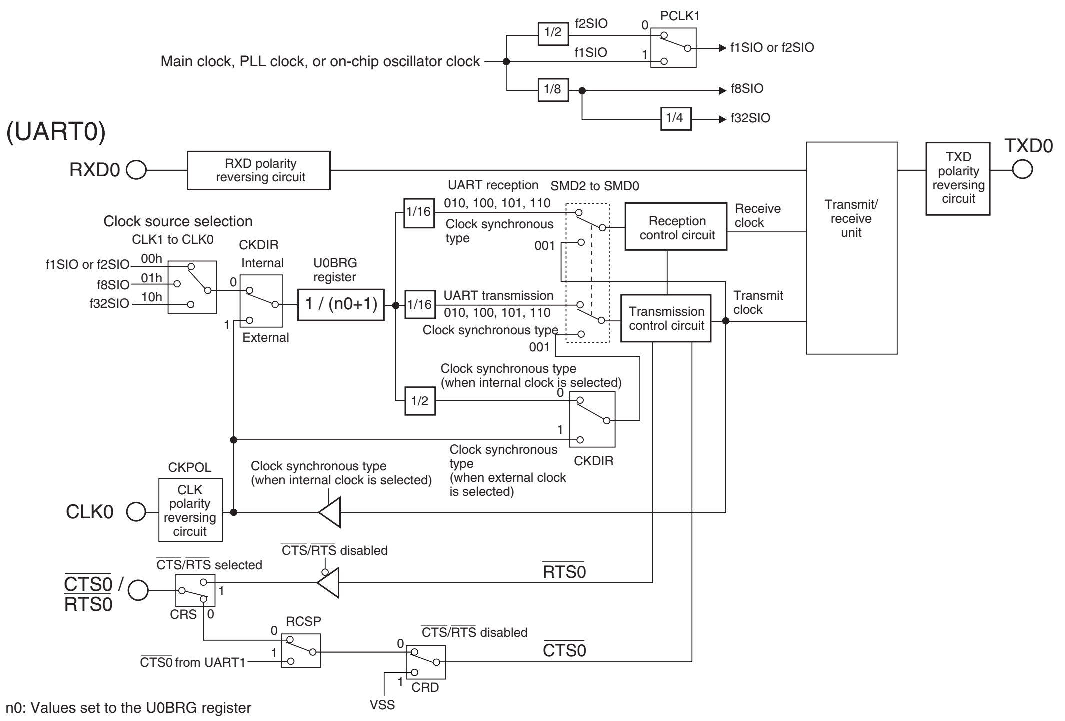
15.1 UARTi (i = 0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other. Figures 15.1 to 15.3 show the UARTi Block Diagram. Figure 15.4 shows the UARTi Transmit/Receive Unit.

UARTi has the following modes:

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode).
- Special mode 1 (I^{2}C mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode)
- Special mode 4 (SIM mode) : UART2

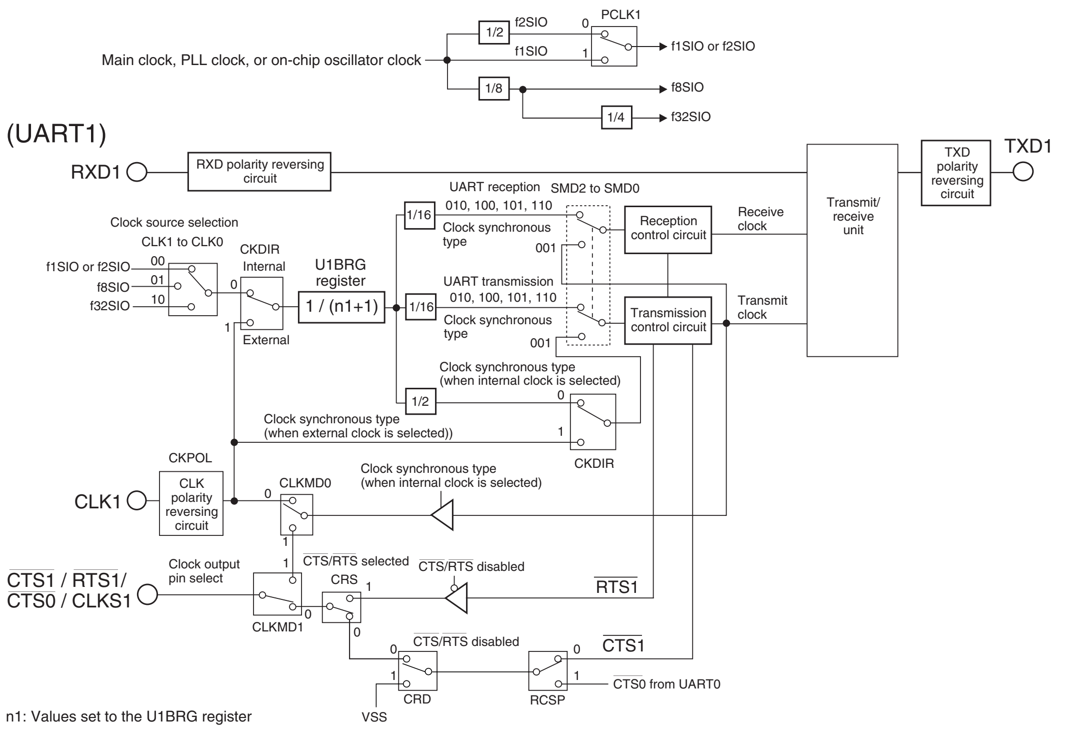
Figures 15.5 to 15.10 show the UARTi-related registers.
Refer to tables listing each mode for register setting.



n0: Values set to the U0BRG register

PCLK1: Bit in PCLKR register
SMD2 to SMD0, CKDIR: Bits in U0MR register
CLK1 to CLK0, CKPOL, CRD, CRS: Bits in U0C0 register
RCSP: Bit in UCON register

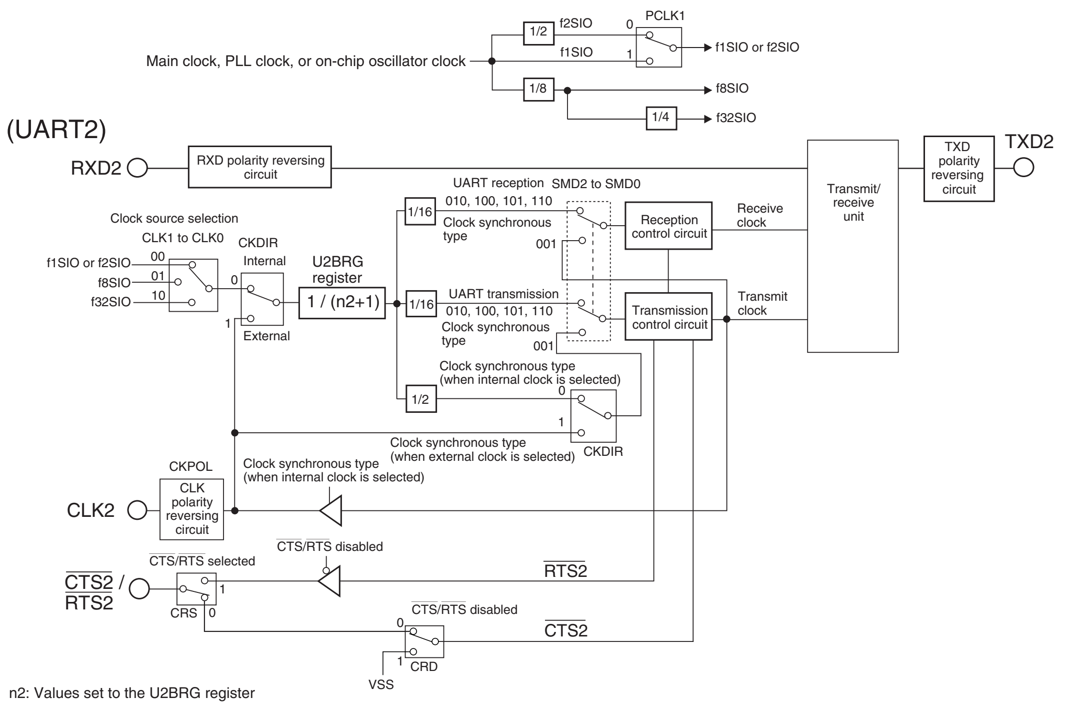
Figure 15.1 UART0 Block Diagram



n1: Values set to the U1BRG register

PCLK1: Bit in PCLKR register
SMD2 to SMD0, CKDIR: Bits in U1MR register
CLK1 to CLK0, CKPOL, CRD, CRS: Bits in U1C0 register
CLKMD0, CLKMD1, RCSP: Bits in UCON register

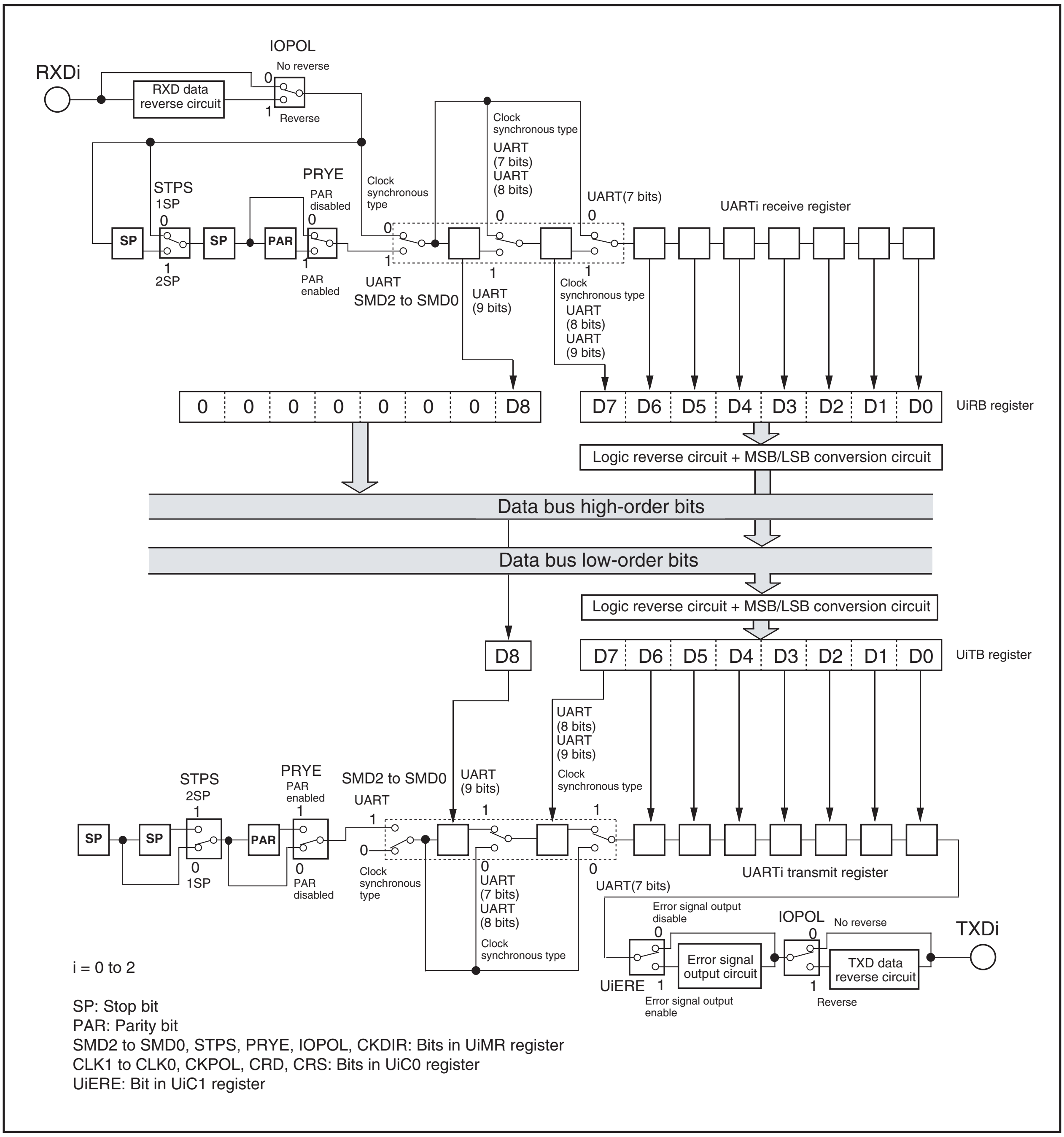
Figure 15.2 UART1 Block Diagram



n2: Values set to the U2BRG register

PCLK1: Bit in PCLKR register
SMD2 to SMD0, CKDIR: Bits in U2MR register
CLK1 to CLK0, CKPOL, CRD, CRS: Bits in U2C0 register

Figure 15.3 UART2 Block Diagram

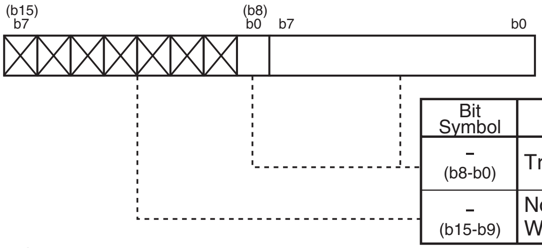


i = 0 to 2

SP: Stop bit
PAR: Parity bit
SMD2 to SMD0, STPS, PRYE, IOPOL, CKDIR: Bits in UiMR register
CLK1 to CLK0, CKPOL, CRD, CRS: Bits in UiC0 register
UiERE: Bit in UiC1 register

Figure 15.4 UARTi Transmit/Receive Unit

UARTi Transmit Buffer Register (i = 0 to 2) [1]



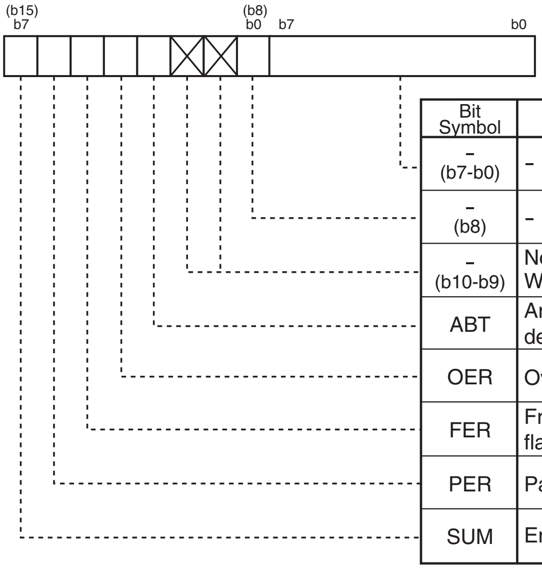
Symbol	Address	After Reset
U0TB	03A3h to 03A2h	Undefined
U1TB	03ABh to 03AAh	Undefined
U2TB	01FBh to 01FAh	Undefined

Bit Symbol	Function	RW
- (b8-b0)	Transmit data	WO
- (b15-b9)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.	-

NOTE:

1. Use the MOV instruction to write to this register.

UARTi Receive Buffer Register (i = 0 to 2)



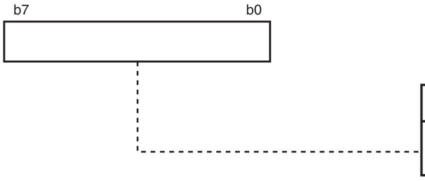
Symbol	Address	After Reset
U0RB	03A7h to 03A6h	Undefined
U1RB	03AFh to 03AEh	Undefined
U2RB	01FFh to 01FEh	Undefined

Bit Symbol	Bit Name	Function	RW
- (b7-b0)	-	Receive data (D7 to D0)	RO
- (b8)	-	Receive data (D8)	RO
- (b10-b9)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		-
ABT	Arbitration lost detecting flag [1]	0 : Not detected 1 : Detected	RW
OER	Overrun error flag [2]	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag [2] [3]	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag [2] [3]	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag [2] [3]	0 : No error 1 : Error found	RO

NOTES:

1. The ABT bit is set to 0 by writing 0 in a program. (Writing 1 has no effect.)
2. When bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled) or the RE bit in the UiC1 register = 0 (reception disabled), all of bits SUM, PER, FER, and OER are set to 0 (no error). The SUM bit is set to 0 (no error) when all of the PER, FER and OER bits are = 0 (no error).
   Also, the PER and FER bits are set to 0 by reading the lower byte of the UiRB register.
3. These error flags are disabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous serial I/O mode) or to 010b (I^{2}C mode). When read, the content is undefined.

UARTi Bit Rate Register (i = 0 to 2) [1] [2] [3]



Symbol	Address	After Reset
U0BRG	03A1h	Undefined
U1BRG	03A9h	Undefined
U2BRG	01F9h	Undefined

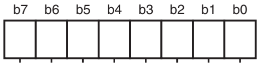
Bit Symbol	Function	Setting Range	RW
- (b7-b0)	Assuming that set value = n, UiBRG divides the count source by n + 1	00h to FFh	WO

NOTES:

1. Write to this register while serial interface is neither transmitting nor receiving.
2. Use the MOV instruction to write to this register.
3. Write to this register after setting bits CLK1 to CLK0 in the UiC0 register.

Figure 15.5 Registers U0TB to U2TB, U0RB to U2RB, and U0BRG to U2BRG

UARTi Transmit/Receive Mode Register (i = 0 to 2)

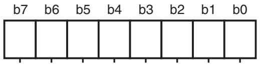


Symbol	Address	After Reset
U0MR to U2MR	03A0h, 03A8h, 01F8h	00h

Bit Symbol	Bit Name	Function	RW
SMD0	Serial I/O mode select bits (1)	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : I^2C mode (2) 1 0 0 : UART mode transfer data 7-bit long 1 0 1 : UART mode transfer data 8-bit long 1 1 0 : UART mode transfer data 9-bit long Do not set a value except above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (3)	RW
STPS	Stop bit length select bit	0 : 1 stop bit 1 : 2 stop bits	RW
PRY	Odd/even parity select bit	Effective when the PRYE bit = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
IOPOL	TXD, RXD I/O polarity reverse bit	0 : No reverse 1 : Reverse	RW

NOTES:
1. To receive data, set the corresponding port direction bit for each RXDi pin to 0 (input mode).
2. Set the corresponding port direction bit for pins SCL and SDA to 0 (input mode).
3. Set the corresponding port direction bit for each CLKi pin to 0 (input mode).

UARTi Transmit/Receive Control Register 0 (i = 0 to 2)



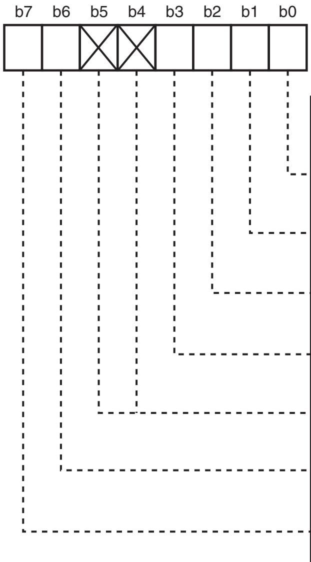
Symbol	Address	After Reset
U0C0 to U2C0	03A4h, 03ACh, 01FCh	00001000b

Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG count source select bits (5)	b1 b0 0 0 : f1SIO or f2SIO is selected (6) 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Do not set a value	RW
CLK1			RW
CRS	$\overline{CTS}/\overline{RTS}$ function select bit (1)	Effective when CRD = 0 0 : $\overline{CTS}$ function is selected (2) 1 : $\overline{RTS}$ function is selected	RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (P6_0, P6_4, P7_3 can be used as I/O ports)	RW
NCH	Data output select bit (3)	0 : Pins TXDi/SDAi and SCLi are CMOS output 1 : Pins TXDi/SDAi and SCLi are N channel open-drain output	RW
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	RW
UFORM	Transfer format select bit (4)	0 : LSB first 1 : MSB first	RW

NOTES:
1. $\overline{CTS1}/\overline{RTS1}$ can be used when the CLKMD1 bit in the UCON register = 0 (only CLK1 output) and the RCSP bit in the UCON register = 0 ($\overline{CTS0}/\overline{RTS0}$ not separated).
2. Set the corresponding port direction bit for each $\overline{CTSi}$ pin to 0 (input mode).
3. SCL2/P7_1 is N channel open-drain output. The NCH bit in the U2C0 register is N channel open-drain output regardless of the NCH bit.
4. The UFORM bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous serial I/O mode), or 101b (UART mode, 8-bit transfer data).
   Set this bit to 1 when bits SMD2 to SMD0 are set to 010b (I^2C mode), and to 0 when bits SMD2 to SMD0 are set to 100b (UART mode, 7-bit transfer data) or 110b (UART mode, 9-bit transfer data).
5. When changing bits CLK1 to CLK0, set the UiBRG register.
6. Selected by the PCLK1 bit in the PCLKR register.

Figure 15.6 Registers U0MR to U2MR and U0C0 to U2C0

UARTj Transmit/Receive Control Register 1 (j = 0, 1)



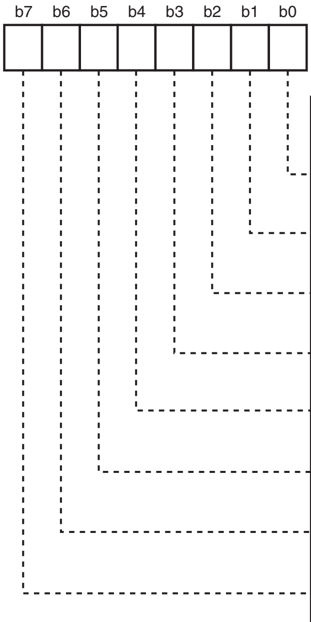
Symbol	Address	After Reset
U0C1, U1C1	03A5h, 03ADh	00XX0010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in the UjTB register 1 : No data present in the UjTB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in the UjRB register 1 : Data present in the UjRB register	RO
- (b5-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
UjLCH	Data logic select bit [(1)]	0 : No reverse 1 : Reverse	RW
UjERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

NOTE:

1. The UjLCH bit is enabled when bits SMD2 to SMD0 in the UjMR register are set to 001b (clock synchronous serial I/O mode), 100b (UART mode, 7-bit transfer data) or 101b (UART mode, 8-bit transfer data). Set this bit to 0 when bits SMD2 to SMD0 are set to 010b (I^2C mode) or 110b (UART mode, 9-bit transfer data).

UART2 Transmit/Receive Control Register 1



Symbol	Address	After Reset
U2C1	01FDh	00000010b

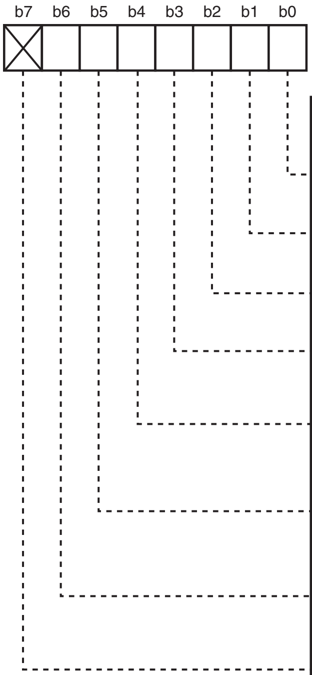
Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in U2TB register 1 : No data present in U2TB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in U2RB register 1 : Data present in U2RB register	RO
U2IRS	UART2 transmit interrupt source select bit	0 : Transmit buffer empty (TI bit = 1) 1 : Transmission completed (TXEPT bit = 1)	RW
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
U2LCH	Data logic select bit [(1)]	0 : No reverse 1 : Reverse	RW
U2ERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

NOTE:

1. The U2LCH bit is enabled when bits SMD2 to SMD0 in the U2MR register are set to 001b (clock synchronous serial I/O mode), 100b (UART mode, 7-bit transfer data) or 101b (UART mode, 8-bit transfer data). Set this bit to 0 when bits SMD2 to SMD0 are set to 010b (I^2C mode) or 110b (UART mode, 9-bit transfer data).

Figure 15.7 Registers U0C1, U1C1, and U2C1

UART Transmit/Receive Control Register 2



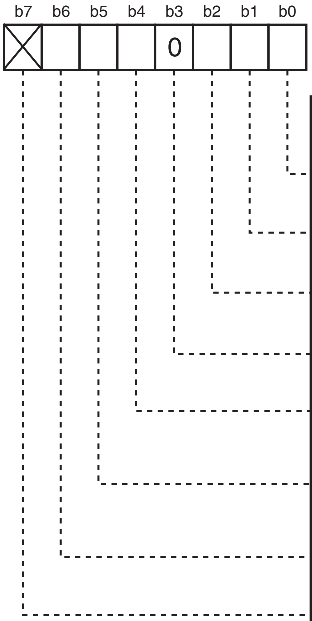
Symbol	Address	After Reset
UCON	03B0h	X0000000b

Bit Symbol	Bit Name	Function	RW
U0IRS	UART0 transmit interrupt source select bit	0 : Transmit buffer empty (TI bit = 1) 1 : Transmission completed (TXEPT bit = 1)	RW
U1IRS	UART1 transmit interrupt source select bit	0 : Transmit buffer empty (TI bit = 1) 1 : Transmission completed (TXEPT bit = 1)	RW
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
CLKMD0	UART1 CLK/CLKS select bit 0	Effective when the CLKMD1 bit = 1 0 : Clock output from CLK1 1 : Clock output from CLKS1	RW
CLKMD1	UART1 CLK/CLKS select bit 1 [1]	0 : CLK output is only CLK1 1 : Transfer clock output from multiple pins function selected	RW
RCSP	Separate UART0 $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ bit	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ shared pin 1 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ separated ($\overline{\text{CTS0}}$ supplied from the P6_4 pin)	RW
- (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTE:

1. When using multiple transfer clock output pins, make sure the following conditions are met:
   - The CKDIR bit in the U1MR register = 0 (internal clock)

UARTi Special Mode Register (i = 0 to 2)



Symbol	Address	After Reset
U0SMR to U2SMR	01EFh, 01F3h, 01F7h	X0000000b

Bit Symbol	Bit Name	Function	RW
IICM	I^{2}C mode select bit	0 : Other than I^{2}C mode 1 : I^{2}C mode	RW
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	RW
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected (busy)	RW [1]
- (b3)	Reserved bit	Set to 0	RW
ABSCS	Bus collision detect sampling clock select bit	0 : Rising edge of transfer clock 1 : Underflow signal of timer Aj [2]	RW
ACSE	Auto clear function select bit of transmit enable bit	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	RW
SSS	Transmit start condition select bit	0 : Not synchronized to RXDi 1 : Synchronized to RXDi [3]	RW
- (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTES:

1. The BBS bit is set to 0 by writing 0 in a program (writing 1 has no effect).
2. Underflow signal of timer A3 in UART0, underflow signal of timer A4 in UART1, underflow signal of timer A0 in UART2.
3. When a transfer begins, the SSS bit is set to 0 (not synchronized to RXDi).

Figure 15.8 Registers UCON, and U0SMR to U2SMR

UARTi Special Mode Register 2 (i = 0 to 2)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
U0SMR2 to U2SMR2	01EEh, 01F2h, 01F6h	X0000000b

Bit Symbol	Bit Name	Function	RW
IICM2	I^2C mode select bit 2	See **Table 15.12 I^2C Mode Functions**	RW
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	RW
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	RW
ALS	SDA output stop bit	0 : Disabled 1 : Enabled	RW
STAC	UARTi initialization bit	0 : Disabled 1 : Enabled	RW
SWC2	SCL wait output bit 2	0: Transfer clock 1: "L" output	RW
SDHI	SDA output disable bit	0: Enabled 1: Disabled (high-impedance)	RW
- (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

UARTi Special Mode Register 3 (i = 0 to 2)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
U0SMR3 to U2SMR3	01EDh, 01F1h, 01F5h	000X0X0Xb

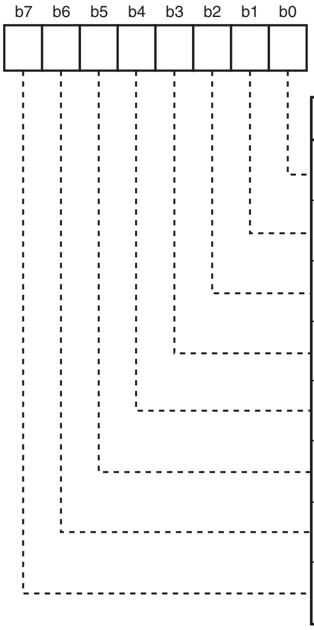
Bit Symbol	Bit Name	Function	RW
- (b0)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
CKPH	Clock phase set bit	0 : Without clock delay 1 : With clock delay	RW
- (b2)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
NODC	Clock output select bit	0 : CLKi is CMOS output 1 : CLKi is N channel open-drain output	RW
- (b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
DL0	SDAi digital delay setup bits (1) (2)	b7 b6 b5 0 0 0 : Without delay 0 0 1 : 1 to 2 cycle(s) of UiBRG count source 0 1 0 : 2 to 3 cycles of UiBRG count source 0 1 1 : 3 to 4 cycles of UiBRG count source 1 0 0 : 4 to 5 cycles of UiBRG count source 1 0 1 : 5 to 6 cycles of UiBRG count source 1 1 0 : 6 to 7 cycles of UiBRG count source 1 1 1 : 7 to 8 cycles of UiBRG count source	RW
DL1			RW
DL2			RW

NOTES:

1. Bits DL2 to DL0 are used to generate a delay in SDAi output by digital means during I^2C mode. In other than I^2C mode, set these bits to 000b (no delay).
2. The amount of delay varies with the load on pins SCLi and SDAi. Also, when using an external clock, the amount of delay increases by about 100 ns.

Figure 15.9 Registers U0SMR2 to U2SMR2 and U0SMR3 to U2SMR3

UARTi Special Mode Register 4 (i = 0 to 2)



Symbol	Address	After Reset
U0SMR4 to U2SMR4	01ECh, 01F0h, 01F4h	00h

Bit Symbol	Bit Name	Function	RW
STAREQ	Start condition generate bit [1]	0 : Clear 1 : Start	RW
RSTAREQ	Restart condition generate bit [1]	0 : Clear 1 : Start	RW
STPREQ	Stop condition generate bit [1]	0 : Clear 1 : Start	RW
STSPSEL	SCL,SDA output select bit	0 : Start and stop conditions not output 1 : Start and stop conditions output	RW
ACKD	ACK data bit	0 : ACK 1 : NACK	RW
ACKC	ACK data output enable bit	0 : Serial interface data output 1 : ACK data output	RW
SCLHI	SCL output stop enable bit	0 : Disabled 1 : Enabled	RW
SWC9	SCL wait bit 3	0 : SCL "L" hold disabled 1 : SCL "L" hold enabled	RW

NOTE:

1. Set to 0 when each condition is generated.

Figure 15.10 Registers U0SMR4 to U2SMR4

15.1.1 Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data.

Table 15.1 lists the Clock Synchronous Serial I/O Mode Specifications. Table 15.2 lists the Registers to be Used in and Setting in Clock Synchronous Serial I/O Mode.

Table 15.1 Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	Transfer data length: 8 bits
Transfer clock	The CKDIR bit in the UiMR register = 0 (internal clock) : fj/(2(n+1)) • fj = f1SIO, f2SIO, f8SIO, f32SIO. n: Setting value of the UiBRG register 00h to FFh The CKDIR bit = 1 (external clock) : Input from CLKi pin
Transmit/receive control	Selectable from $\overline{\text{CTS}}$ function, $\overline{\text{RTS}}$ function or $\overline{\text{CTS}}/\overline{\text{RTS}}$ function disabled
Transmit start condition	Before transmission can start, meet the following requirements [1] • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register) • If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTS}}$i pin = L
Receive start condition	Before reception can start, meet the following requirements [1] • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt request generation timing	For transmission, one of the following conditions can be selected • The UiIRS bit [2] = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit =1 (transmission completed): when the serial interface finished transmitting data from the UARTi transmit register For reception • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	Overrun error [3] This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select function	• CLK polarity selection Transfer data input/output can be selected to occur synchronously with the rising or the falling edge of the transfer clock • LSB first, MSB first selection Whether to start transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected • Continuous receive mode selection Reception is enabled immediately by reading the UiRB register • Switching serial data logic This function reverses the logic value of the transmit/receive data • Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set • Separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ pins (UART0) $\overline{\text{CTS0}}$ and $\overline{\text{RTS0}}$ are input/output from separate pins

i = 0 to 2

NOTES:

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. Bits U0IRS and U1IRS are bits 0 and 1 in the UCON register; the U2IRS bit is bit 4 in the U2C1 register.
3. If an overrun error occurs, the receive data of UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.

Table 15.2 Registers to be Used and Settings in Clock Synchronous Serial I/O Mode

Register	Bit	Function
UiTB (1)	0 to 7	Set transmit data
UiRB (1)	0 to 7	Receive data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a bit rate
UiMR (1)	SMD2 to SMD0	Set to 001b
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to 0
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ to use
	TXEPT	Transmit register empty flag
	CRD	Select $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled or disabled
	NCH	Select TXDi pin output mode
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS (2)	Select the UART2 transmit interrupt source
	U2RRM (2)	Set this bit to 1 to use continuous receive mode
	UiLCH	Set this bit to 1 to use inverted data logic
	UiERE	Set to 0
UiSMR	0 to 7	Set to 0
UiSMR2	0 to 7	Set to 0
UiSMR3	0 to 2	Set to 0
	NODC	Select clock output mode
	4 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
UCON	U0IRS, U1IRS	Select the UART0/UART1 transmit interrupt source
	U0RRM, U1RRM	Set this bit to 1 to use continuous receive mode
	CLKMD0	Select the transfer clock output pin when the CLKMD1 bit = 1
	CLKMD1	Set this bit to 1 to output UART1 transfer clock from two pins
	RCSP	Set this bit to 1 to accept as input the $\overline{\text{CTS0}}$ signal of the UART0 from the P6_4 pin
	7	Set to 0

i = 0 to 2

NOTES:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in clock synchronous serial I/O mode.
2. Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.

Table 15.3 lists the I/O Pin Functions (when not select multiple transfer clock output pin select function) in clock synchronous serial I/O mode. Table 15.4 lists the P6_4 Pin Functions in clock synchronous serial I/O mode.

Note that for a period from when the UARTi operating mode is selected to when transfer starts, the TXDi pin outputs an “H”.

Figure 15.11 shows the Transmit/Receive Operation during clock synchronous serial I/O mode.

Table 15.3 I/O Pin Functions (when not select multiple transfer clock output pin select function)

Pin Name	Function	Method of Selection
TXDi (P6_3, P6_7, P7_0)	Serial data output	(Outputs dummy data when performing reception only)
RXDi (P6_2, P6_6, P7_1)	Serial data input	Bits PD6_2 and PD6_6 in PD6 register = 0 PD7_1 bit in PD7 register = 0 (Can be used as an input port when performing transmission only)
CLKi (P6_1, P6_5, P7_2)	Transfer clock output	CKDIR bit in UiMR register = 0
	Transfer clock input	CKDIR bit = 1 Bits PD6_1 and PD6_5 in PD6 register = 0 PD7_2 bit in PD7 register = 0
CTSi/RTSi (P6_0, P6_4, P7_3)	CTS input	CRD bit in UiC0 register = 0 CRS bit in UiC0 register = 0 Bits PD6_0 and PD6_4 in PD6 register = 0 PD7_3 bit in PD7 register = 0
	RTS output	CRD bit = 0 CRS bit = 1
	I/O port	CRD bit = 1

i = 0 to 2

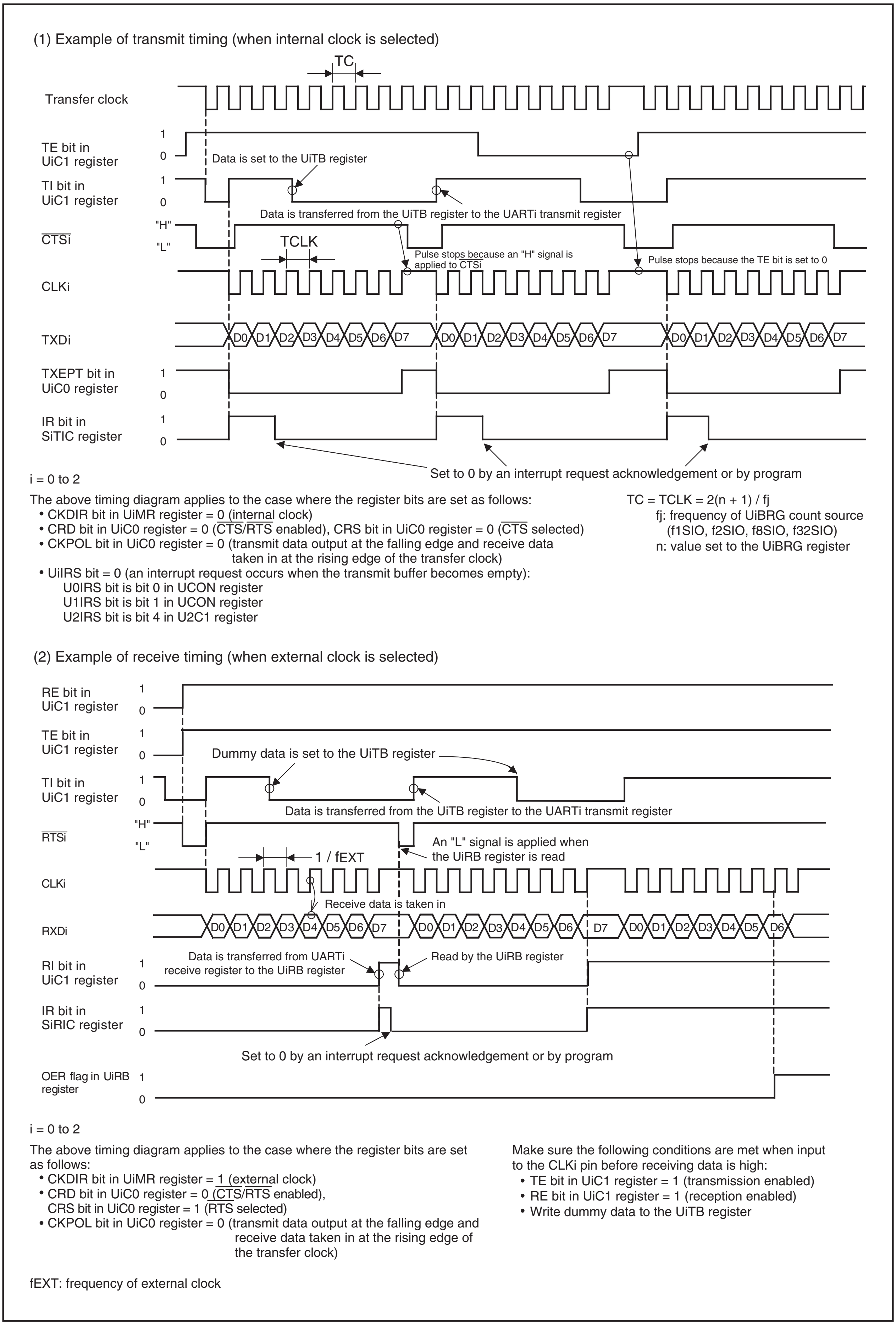
Table 15.4 P6_4 Pin Functions

Pin Function	Bit set Value					
	U1C0 Register		UCON Register			PD6 Register
	CRD bit	CRS bit	RCSP bit	CLKMD1 bit	CLKMD0 bit	PD6_4 bit
P6_4	1	-	0	0	-	Input: 0, Output: 1
CTS1	0	0	0	0	-	0
RTS1	0	1	0	0	-	-
CTS0 [1]	0	0	1	0	-	0
CLKS1	-	-	-	1 [2]	1	-

-: 0 or 1

NOTES:

1. In addition to this, set the CRD bit in the U0C0 register to 0 (CTS0/RTS0 enabled) and the CRS bit in the U0C0 register to 1 (RTS0 selected).
2. When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:
   - High if the CLKPOL bit in the U1C0 register = 0
   - Low if the CLKPOL bit = 1



(1) Example of transmit timing (when internal clock is selected)

TC
Transfer clock
TE bit in UiC1 register
1
0
Data is set to the UiTB register
TI bit in UiC1 register
Data is transferred from the UiTB register to the UARTi transmit register
"H"
"L"
CTSi
TCLK
Pulse stops because an "H" signal is applied to CTSi
Pulse stops because the TE bit is set to 0
CLKi
TXDi
D0 D1 D2 D3 D4 D5 D6 D7
TXEPT bit in UiC0 register
IR bit in SiTIC register
Set to 0 by an interrupt request acknowledgement or by program

i = 0 to 2

The above timing diagram applies to the case where the register bits are set as follows:
- CKDIR bit in UiMR register = 0 (internal clock)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled), CRS bit in UiC0 register = 0 (CTS selected)
- CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- UiIRS bit = 0 (an interrupt request occurs when the transmit buffer becomes empty):
  U0IRS bit is bit 0 in UCON register
  U1IRS bit is bit 1 in UCON register
  U2IRS bit is bit 4 in U2C1 register

TC = TCLK = 2(n + 1) / fj
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
n: value set to the UiBRG register

(2) Example of receive timing (when external clock is selected)

RE bit in UiC1 register
1
0
TE bit in UiC1 register
TI bit in UiC1 register
Dummy data is set to the UiTB register
Data is transferred from the UiTB register to the UARTi transmit register
"H"
"L"
RTSi
An "L" signal is applied when the UiRB register is read
1 / fEXT
CLKi
Receive data is taken in
RXDi
D0 D1 D2 D3 D4 D5 D6 D7
RI bit in UiC1 register
Data is transferred from UARTi receive register to the UiRB register
Read by the UiRB register
IR bit in SiRIC register
Set to 0 by an interrupt request acknowledgement or by program
OER flag in UiRB register

i = 0 to 2

The above timing diagram applies to the case where the register bits are set as follows:
- CKDIR bit in UiMR register = 1 (external clock)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled), CRS bit in UiC0 register = 1 (RTS selected)
- CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

Make sure the following conditions are met when input to the CLKi pin before receiving data is high:
- TE bit in UiC1 register = 1 (transmission enabled)
- RE bit in UiC1 register = 1 (reception enabled)
- Write dummy data to the UiTB register

fEXT: frequency of external clock

Figure 15.11 Transmit and Receive Operation

15.1.1.1 Counter Measure for Communication Error Occurs

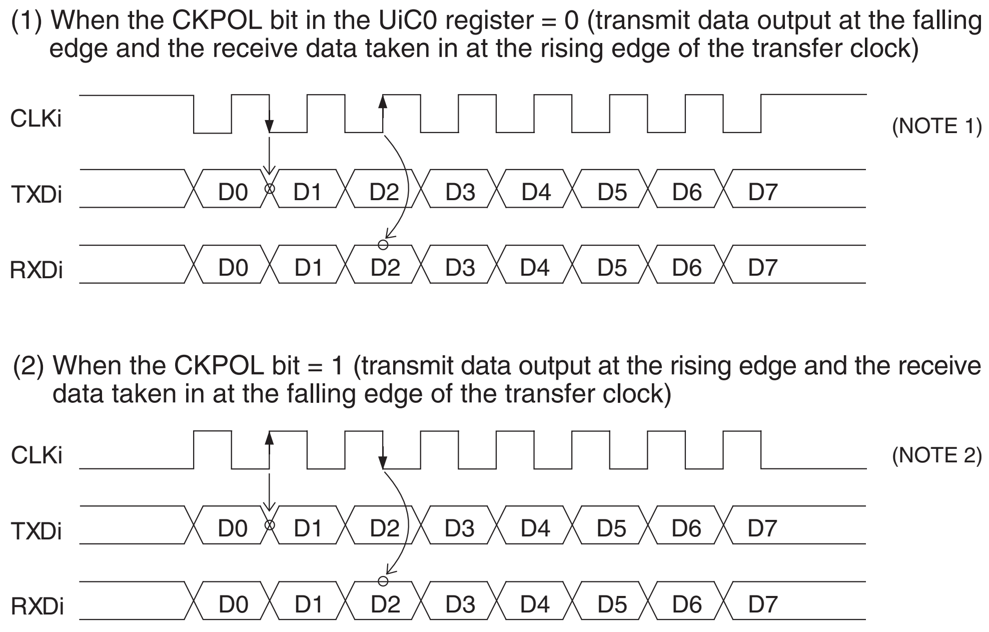
If a communication error occurs while transmitting or receiving in clock synchronous serial I/O mode, follow the procedures below.

- Resetting the UiRB register (i = 0 to 2)
  (1) Set the RE bit in the UiC1 register to 0 (reception disabled)
  (2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled)
  (3) Set bits SMD2 to SMD0 in the UiMR register to 001b (clock synchronous serial I/O mode)
  (4) Set the RE bit in the UiC1 register to 1 (reception enabled)

- Resetting the UiTB register (i = 0 to 2)
  (1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled)
  (2) Set bits SMD2 to SMD0 in the UiMR register to 001b (clock synchronous serial I/O mode)
  (3) 1 (transmission enabled) is written to the TE bit in the UiC1 register, regardless of the TE bit

15.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i = 0 to 2) to select the transfer clock polarity. Figure 15.12 shows the Transfer Clock Polarity.



(1) When the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock)

(2) When the CKPOL bit = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock)

i = 0 to 2

* This applies to the case where the UFORM bit in the UiC0 register = 0 (LSB first) and the UiLCH bit in the UiC1 register = 0 (no reverse).

NOTES:
1. When not transferring, the CLKi pin outputs a high signal.
2. When not transferring, the CLKi pin outputs a low signal.

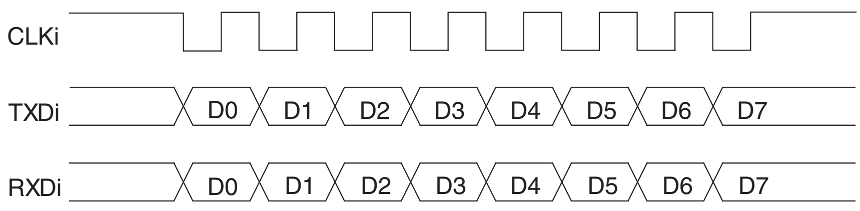
Figure 15.12 Transfer Clock Polarity

15.1.1.3 LSB First/MSB First Select Function

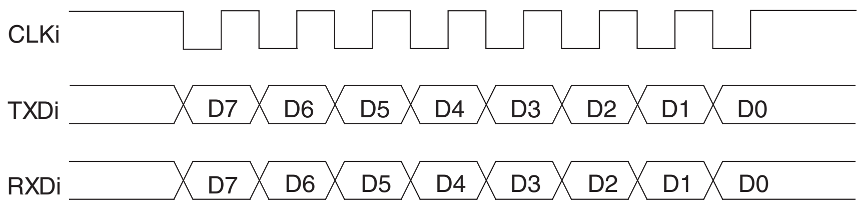
Use the UFORM bit in the UiC0 register (i = 0 to 2) to select the transfer format.

Figure 15.13 shows the Transfer Format.

(1) When the UFORM bit in the UiC0 register = 0 (LSB first)



(2) When the UFORM bit = 1 (MSB first)



i = 0 to 2

* This applies to the case where the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock) and the UiLCH bit in the UiC1 register = 0 (no reverse).

Figure 15.13 Transfer Format

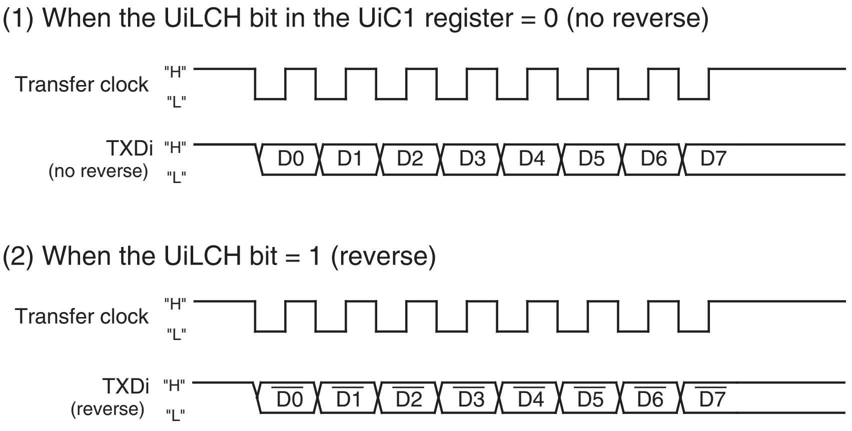
15.1.1.4 Continuous Receive Mode

In continuous receive mode, receive operation becomes enable when the receive buffer register is read. It is not necessary to write dummy data into the transmit buffer register to enable receive operation in this mode. However, a dummy read of the receive buffer register is required when starting the operating mode.

When the UiRRM bit (i = 0 to 2) = 1 (continuous receive mode), the TI bit in the UiC1 register is set to 0 (data present in UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit = 1, do not write dummy data to the UiTB register in a program. Bits U0RRM and U1RRM are bits 2 and 3 in the UCON register, respectively, and the U2RRM bit is the bit 5 in the U2C1 register.

15.1.1.5 Serial Data Logic Switching Function

When the UiLCH bit in the UiC1 register (i = 0 to 2) = 1 (reverse), the data written to the UiTB register has its logic reversed before being transmitted. Similarly, the receive data has its logic reversed when read from the UiRB register. Figure 15.14 shows the Serial Data Logic Switching.


(1) When the UiLCH bit in the UiC1 register = 0 (no reverse)
Transfer clock
"H"
"L"
TXDi
(no reverse)
"H"
"L"
D0 D1 D2 D3 D4 D5 D6 D7
(2) When the UiLCH bit = 1 (reverse)
Transfer clock
"H"
"L"
TXDi
(reverse)
"H"
"L"
$\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$

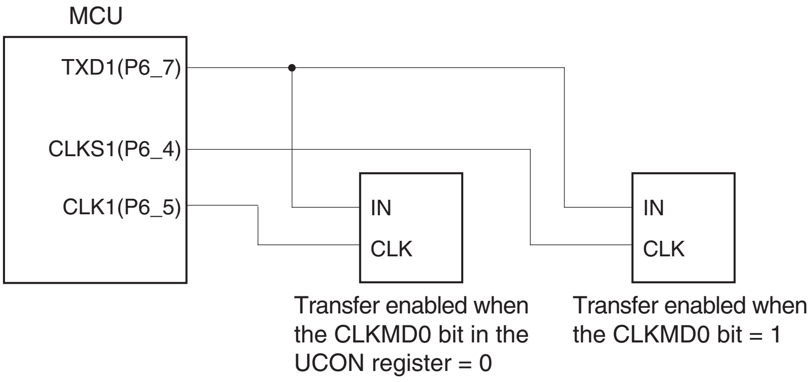
i = 0 to 2

* This applies to the case where the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock) and the UFORM bit = 0 (LSB first).

Figure 15.14 Serial Data Logic Switching

15.1.1.6 Transfer Clock Output From Multiple Pins (UART1)

Use bits CLKMD1 to CLKMD0 in the UCON register to select one of the two transfer clock output pins. Figure 15.15 shows the Transfer Clock Output from Multiple Pins. This function can be used when the selected transfer clock for UART1 is an internal clock.


MCU
TXD1(P6_7)
CLKS1(P6_4)
CLK1(P6_5)
IN
CLK
IN
CLK
Transfer enabled when the CLKMD0 bit in the UCON register = 0
Transfer enabled when the CLKMD0 bit = 1

* This applies to the case where the CKDIR bit in the U1MR register = 0 (internal clock) and the CLKMD1 bit in the UCON register = 1 (transfer clock output from multiple pins).

Figure 15.15 Transfer Clock Output from Multiple Pins

15.1.1.7 CTS/RTS Function

When the CTS function is used transmit and receive operation start when “L” is applied to the CTSi/RTSi (i = 0 to 2) pin. Transmit and receive operation begins when the CTSi/RTSi pin is held “L”. If the “L” signal is switched to “H” during a transmit or receive operation, the operation stops before the next data.

When the RTS function is used, the CTSi/RTSi pin outputs on “L” signal when the MCU is ready to receive. The output level becomes “H” on the first falling edge of the CLKi pin.

- CRD bit in UiC0 register = 1 (CTS/RTS function disabled) CTSi/RTSi pin is programmable I/O function
- CRD bit = 0, CRS bit in UiC0 register = 0 (CTS function is selected)
  CTSi/RTSi pin is CTS function
- CRD bit = 0, CRS bit = 1 (RTS function is selected) CTSi/RTSi pin is RTS function

15.1.1.8 CTS/RTS Separate Function (UART0)

This function separates CTS0/RTS0, outputs RTS0 from the P6_0 pin, and accepts as input the CTS0 from the P6_4 pin. To use this function, set the register bits as shown below.

- CRD bit in U0C0 register = 0 (CTS/RTS of UART0 enabled)
- CRS bit in U0C0 register = 1 (output RTS of UART0)
- CRD bit in U1C0 register = 0 (CTS/RTS of UART1 enabled)
- CRS bit in U1C0 register = 0 (input CTS of UART1)
- RCSP bit in UCON register = 1 (input CTS0 from the P6_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

Note that when using the CTS/RTS separate function, CTS/RTS of UART1 separate function cannot be used.

Figure 15.16 shows the CTS/RTS Separate Function.

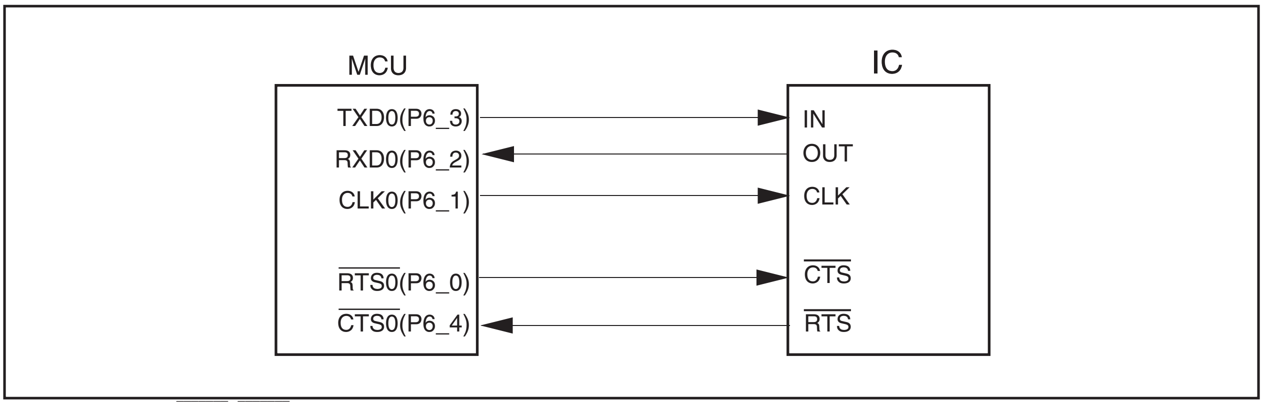


Figure 15.16 CTS/RTS Separate Function

15.1.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired bit rate and transfer data format. Table 15.5 lists the UART Mode Specifications. Table 15.6 lists the Registers to be Used and Setting in UART Mode.

Table 15.5 UART Mode Specifications

Item	Specification
Transfer data format	• Character bit (transfer data): Selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: Selectable from odd, even, or none • Stop bit: Selectable from 1 or 2 bits
Transfer clock	• CKDIR bit in UiMR register = 0 (internal clock) : fj/(16(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO. n: Setting value of the UiBRG register 00h to FFh • The CKDIR bit = 1 (external clock) : fEXT/(16(n+1)) fEXT: Input from CLKi pin. n :Setting value of the UiBRG register 00h to FFh
Transmit/receive control	Selectable from $\overline{\text{CTS}}$ function, $\overline{\text{RTS}}$ function or $\overline{\text{CTS}}/\overline{\text{RTS}}$ function disabled
Transmit start condition	Before transmission can start, meet the following requirements • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in UiTB register) • If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTS}}$i pin = L
Receive start condition	Before reception can start, meet the following requirements • The RE bit in the UiC1 register = 1 (reception enabled) • Start bit detection
Interrupt request generation timing	For transmission, one of the following conditions can be selected • The UiIRS bit (1) = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit =1 (transmission completed): when the serial interface finished transmitting data from the UARTi transmit register For reception • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (2) This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the bit one before the last stop bit of the next data • Framing error (3) This error occurs when the number of stop bits set is not detected • Parity error (3) This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set • Error sum flag This flag is set to 1 when any of the overrun, framing, or parity errors occur
Select function	• LSB first, MSB first selection Whether to start transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected • Serial data logic switch This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. • TXD, RXD I/O polarity switch This function reverses the polarities of the TXD pin output and RXD pin input. The logic levels of all I/O data is reversed. • Separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ pins (UART0) $\overline{\text{CTS0}}$ and $\overline{\text{RTS0}}$ are input/output from separate pins

i = 0 to 2

NOTES:

1. Bits U0IRS and U1IRS are bits 0 and 1 in the UCON register. The U2IRS bit is bit 4 in the U2C1 register.
2. If an overrun error occurs, the receive data of UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

Table 15.6 Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmit data (1)
UiRB	0 to 8	Receive data can be read (1)
	OER,FER,PER,SUM	Error flag
UiBRG	0 to 7	Set a bit rate
UiMR	SMD2 to SMD0	Set these bits to 100b when transfer data is 7-bit long Set these bits to 101b when transfer data is 8-bit long Set these bits to 110b when transfer data is 9-bit long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL	Select the TXD/RXD input/output polarity
UiC0	CLK0 to CLK1	Select the count source for the UiBRG register
	CRS	Select $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ to use
	TXEPT	Transmit register empty flag
	CRD	Select $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled or disabled
	NCH	Select TXDi pin output mode
	CKPOL	Set to 0
	UFORM	LSB first or MSB first can be selected when transfer data is 8-bit long. Set this bit to 0 when transfer data is 7- or 9-bit long.
UiC1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS (2)	Select the UART2 transmit interrupt source
	U2RRM (2)	Set to 0
	UiLCH	Set this bit to 1 to use inverted data logic
	UiERE	Set to 0
UiSMR	0 to 7	Set to 0
UiSMR2	0 to 7	Set to 0
UiSMR3	0 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
UCON	U0IRS, U1IRS	Select the UART0/UART1 transmit interrupt source
	U0RRM, U1RRM	Set to 0
	CLKMD0	Invalid because the CLKMD1 bit = 0
	CLKMD1	Set to 0
	RCSP	Set this bit to 1 to accept as input the $\overline{\text{CTS0}}$ of UART0 signal from the P6_4 pin
	7	Set to 0

i = 0 to 2

NOTES:

1. The bits used for transmit/receive data are as follows:
   - Bits 0 to 6 when transfer data is 7-bit long
   - Bits 0 to 7 when transfer data is 8-bit long
   - Bits 0 to 8 when transfer data is 9-bit long.
2. Set bits 4 to 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are included in the UCON register.

Table 15.7 lists the I/O Pins Functions in UART mode. Table 15.8 lists the P6_4 Pin Functions in UART mode. Note that for a period from when the UARTi operating mode is selected to when transfer starts, the TXDi pin outputs an “H”.

Figure 15.17 shows the Transmit Operation in UART mode. Figure 15.18 shows the Receive Operation in UART mode.

Table 15.7 I/O Pin Functions

Pin Name	Function	Method of Selection
TXDi (P6_3, P6_7, P7_0)	Serial data output	(Outputs “H” when performing reception only)
RXDi (P6_2, P6_6, P7_1)	Serial data input	Bits PD6_2 and PD6_6 in PD6 register = 0 PD7_1 bit in PD7 register = 0 (Can be used as an input port when performing transmission only)
CLKi (P6_1, P6_5, P7_2)	I/O port	CKDIR bit in UiMR register = 0
	Transfer clock input	CKDIR bit in UiMR register = 1 Bits PD6_1 and PD6_5 in PD6 register = 0 PD7_2 bit in PD7 register = 0
CTSi/RTSi (P6_0, P6_4, P7_3)	CTS input	CRD bit in UiC0 register = 0 CRS bit in UiC0 register = 0 Bits PD6_0 and PD6_4 in PD6 register = 0 PD7_3 bit in PD7 register = 0
	RTS output	CRD bit = 0 CRS bit = 1
	I/O port	CRD bit = 1

i = 0 to 2

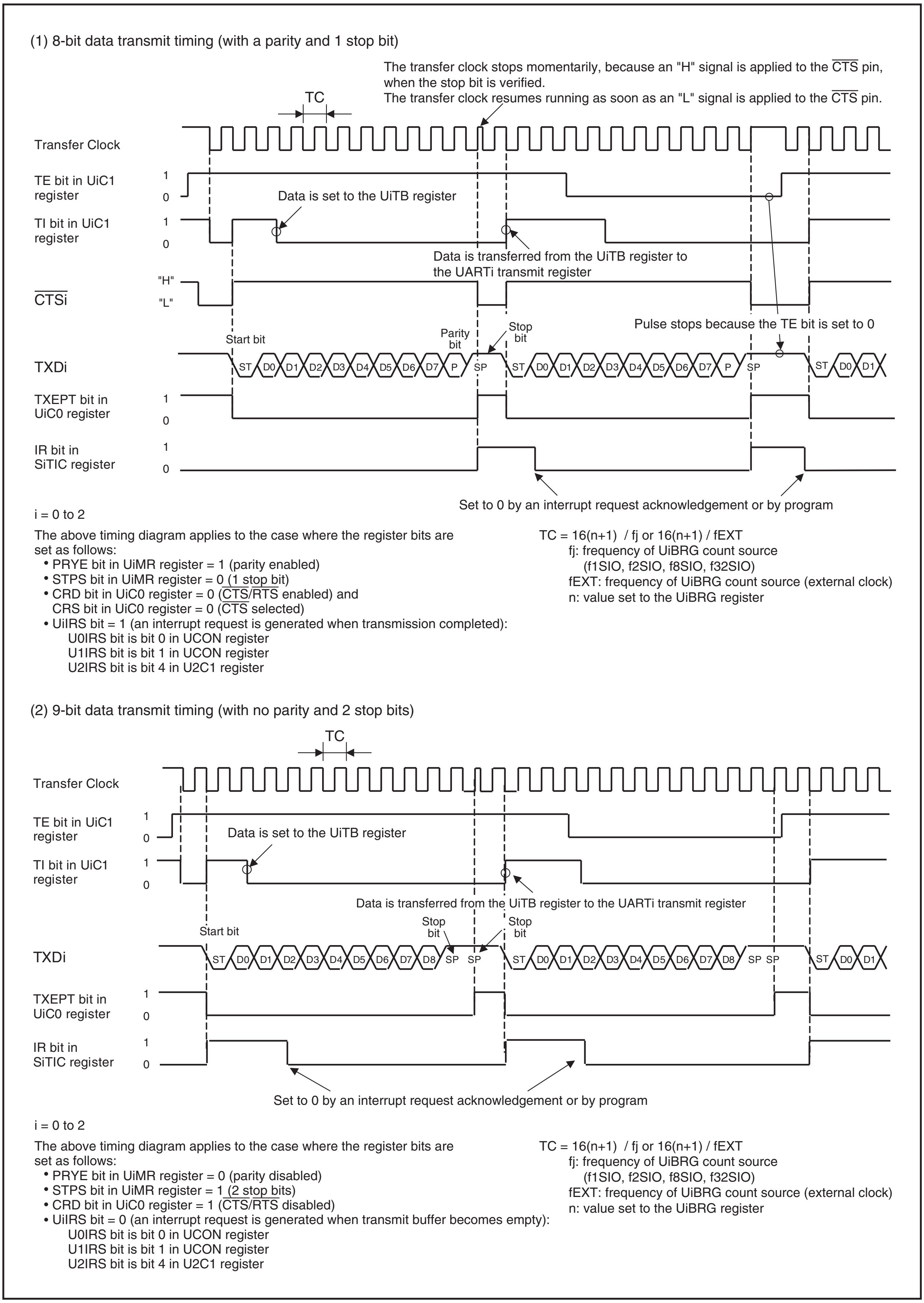
Table 15.8 P6_4 Pin Functions

Pin Function	Bit set Value				
	U1C0 Register		UCON Register		PD6 Register
	CRD bit	CRS bit	RCSP bit	CLKMD1 bit	PD6_4 bit
P6_4	1	-	0	0	Input: 0, Output: 1
CTS1	0	0	0	0	0
RTS1	0	1	0	0	-
CTS0 [1]	0	0	1	0	0

-: 0 or 1

NOTE:

1. In addition to this, set the CRD bit in the U0C0 register to 0 (CTS0/RTS0 enabled) and the CRS bit in the U0C0 register to 1 (RTS0 selected).



(1) 8-bit data transmit timing (with a parity and 1 stop bit)

The transfer clock stops momentarily, because an "H" signal is applied to the CTS pin, when the stop bit is verified.
The transfer clock resumes running as soon as an "L" signal is applied to the CTS pin.
TC
Transfer Clock
TE bit in UiC1 register
TI bit in UiC1 register
CTSi
TXDi
TXEPT bit in UiC0 register
IR bit in SiTIC register
1
0
"H"
"L"
Data is set to the UiTB register
Data is transferred from the UiTB register to the UARTi transmit register
Start bit
Parity bit
Stop bit
Pulse stops because the TE bit is set to 0
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
Set to 0 by an interrupt request acknowledgement or by program

i = 0 to 2

The above timing diagram applies to the case where the register bits are set as follows:

- PRYE bit in UiMR register = 1 (parity enabled)
- STPS bit in UiMR register = 0 (1 stop bit)
- CRD bit in UiC0 register = 0 (CTS/RTS enabled) and CRS bit in UiC0 register = 0 (CTS selected)
- UiIRS bit = 1 (an interrupt request is generated when transmission completed):
  U0IRS bit is bit 0 in UCON register
  U1IRS bit is bit 1 in UCON register
  U2IRS bit is bit 4 in U2C1 register

TC = 16(n+1) / fj or 16(n+1) / fEXT
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT: frequency of UiBRG count source (external clock)
n: value set to the UiBRG register

(2) 9-bit data transmit timing (with no parity and 2 stop bits)

TC
Transfer Clock
TE bit in UiC1 register
TI bit in UiC1 register
TXDi
TXEPT bit in UiC0 register
IR bit in SiTIC register
1
0
Data is set to the UiTB register
Data is transferred from the UiTB register to the UARTi transmit register
Start bit
Stop bit
Stop bit
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP
Set to 0 by an interrupt request acknowledgement or by program

i = 0 to 2

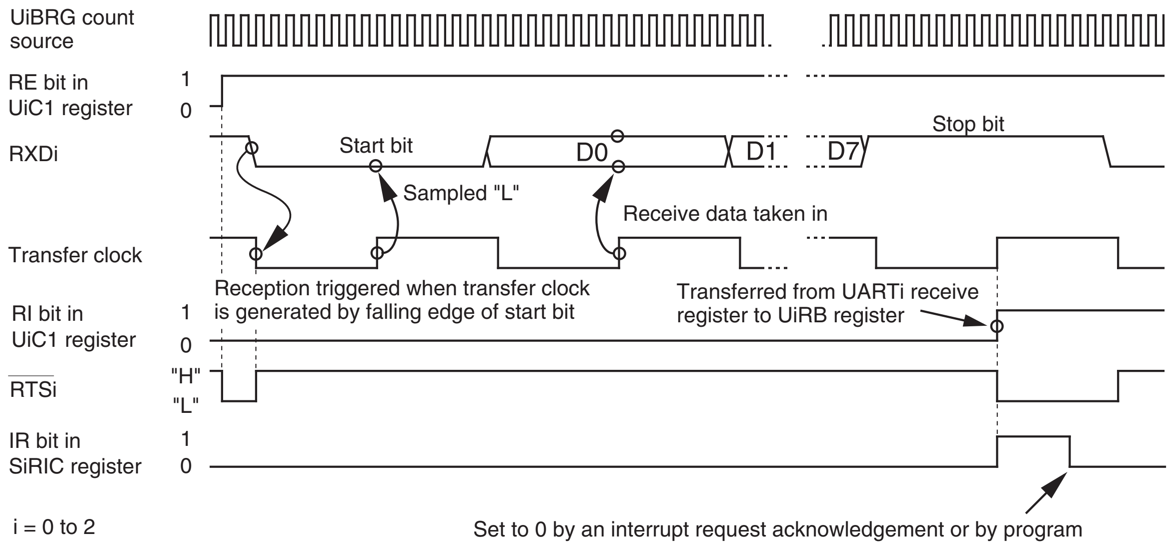
The above timing diagram applies to the case where the register bits are set as follows:

- PRYE bit in UiMR register = 0 (parity disabled)
- STPS bit in UiMR register = 1 (2 stop bits)
- CRD bit in UiC0 register = 1 (CTS/RTS disabled)
- UiIRS bit = 0 (an interrupt request is generated when transmit buffer becomes empty):
  U0IRS bit is bit 0 in UCON register
  U1IRS bit is bit 1 in UCON register
  U2IRS bit is bit 4 in U2C1 register

TC = 16(n+1) / fj or 16(n+1) / fEXT
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT: frequency of UiBRG count source (external clock)
n: value set to the UiBRG register

Figure 15.17 Transmit Operation

- Example of receive timing when transfer data is 8-bit long (parity disabled, one stop bit)


UiBRG count source
RE bit in UiC1 register
1
0
RXDi
Start bit
D0
D1
D7
Stop bit
Sampled "L"
Receive data taken in
Transfer clock
Reception triggered when transfer clock is generated by falling edge of start bit
Transferred from UARTi receive register to UiRB register
RI bit in UiC1 register
1
0
RTSi
"H"
"L"
IR bit in SiRIC register
1
0
i = 0 to 2
Set to 0 by an interrupt request acknowledgement or by program

The above timing diagram applies to the case where the register bits are set as follows:

- PRYE bit in UiMR register = 0 (parity disabled)
- STPS bit in UiMR register = 0 (1 stop bit)
- CRD bit in UiC0 register = 0 ($\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ enabled) and CRS bit = 1 ($\overline{\text{RTSi}}$ selected)

Figure 15.18 Receive Operation

15.1.2.1 Bit Rates

In UART mode, the frequency set by the UiBRG register (i = 0 to 2) divided by 16 become the bit rates. Table 15.9 lists an Example of Bit Rates and Settings.

Table 15.9 Example of Bit Rates and Settings

Bit Rate (bps)	Count Source of UiBRG	Peripheral Function Clock: 16 MHz		Peripheral Function Clock: 20 MHz		Peripheral Function Clock: 24 MHz	
		Set Value of UiBRG: n	Bit Rate (bps)	Set Value of UiBRG: n	Bit Rate (bps)	Set Value of UiBRG: n	Bit Rate (bps)
1200	f8	103 (67h)	1202	129 (81h)	1202	155 (9Bh)	1202
2400	f8	51 (33h)	2404	64 (40h)	2404	77 (4Dh)	2404
4800	f8	25 (19h)	4808	32 (20h)	4735	38 (26h)	4808
9600	f1	103 (67h)	9615	129 (81h)	9615	155 (9Bh)	9615
14400	f1	68 (44h)	14493	86 (56h)	14368	103 (67h)	14423
19200	f1	51 (33h)	19231	64 (40h)	19231	77 (4Dh)	19231
28800	f1	34 (22h)	28571	42 (2Ah)	29070	51 (33h)	28846
31250	f1	31 (1Fh)	31250	39 (27h)	31250	47 (2Fh)	31250
38400	f1	25 (19h)	38462	32 (20h)	37879	38 (26h)	38462
51200	f1	19 (13h)	50000	23 (17h)	52083	28 (1Ch)	51724

i = 0 to 2

15.1.2.2 Counter Measure for Communication Error Occurs

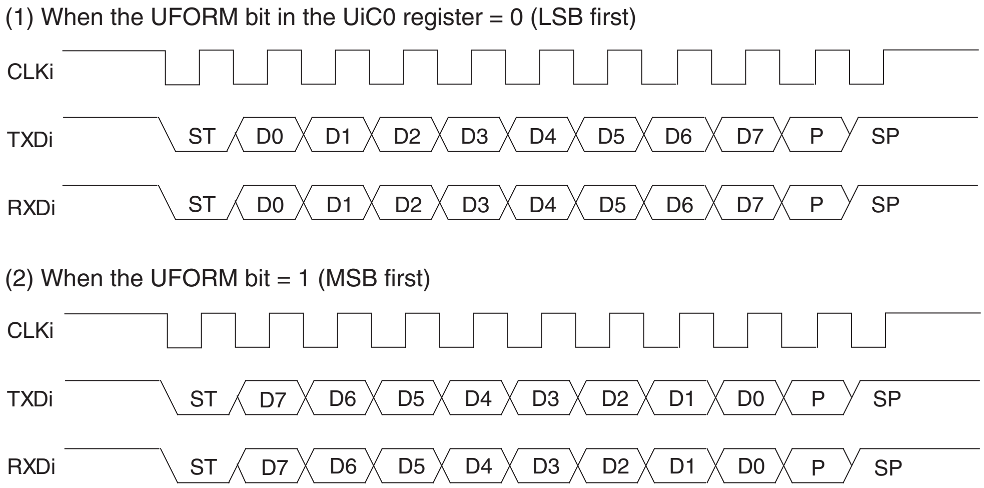
If a communication error occurs while transmitting or receiving in UART mode, follow the procedures below.

- Resetting the UiRB register (i = 0 to 2)
  (1) Set the RE bit in the UiC1 register to 0 (reception disabled)
  (2) Set the RE bit in the UiC1 register to 1 (reception enabled)

- Resetting the UiTB register (i = 0 to 2)
  (1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled)
  (2) Set bits SMD2 to SMD0 in the UiMR register to 001b, 101b, 110b
  (3) 1 (transmission enabled) is written to the TE bit in the UiC1 register, regardless of the TE bit

15.1.2.3 LSB First/MSB First Select Function

As shown in Figure 15.19, use the UFORM bit in the UiC0 register to select the transfer format. Figure 15.19 shows the Transfer Format. This function is valid when transfer data is 8-bit long.



(1) When the UFORM bit in the UiC0 register = 0 (LSB first)

(2) When the UFORM bit = 1 (MSB first)

CLKi
TXDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
RXDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
CLKi
TXDi
ST D7 D6 D5 D4 D3 D2 D1 D0 P SP
RXDi
ST D7 D6 D5 D4 D3 D2 D1 D0 P SP

i = 0 to 2

ST: Start bit
P: Parity bit
SP: Stop bit

NOTE:

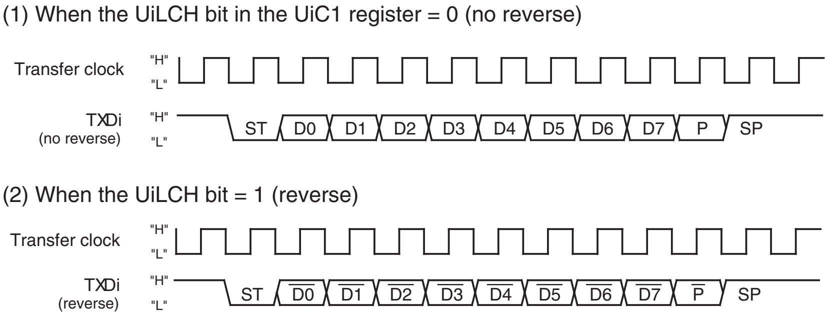
1. This applies to the case where the register bits are set as follows:
   - CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock)
   - UiLCH bit in UiC1 register = 0 (no reverse)
   - STPS bit in UiMR register = 0 (1 stop bit)
   - PRYE bit in UiMR register = 1 (parity enabled)

Figure 15.19 Transfer Format

15.1.2.4 Serial Data Logic Switching Function

The data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register.

Figure 15.20 shows the Serial Data Logic Switching.


Transfer clock
"H"
"L"
TXDi
(no reverse)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
Transfer clock
TXDi
(reverse)
ST $\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$ $\overline{P}$ SP

(1) When the UiLCH bit in the UiC1 register = 0 (no reverse)

(2) When the UiLCH bit = 1 (reverse)

i = 0 to 2
ST: Start bit
P: Parity bit
SP: Stop bit

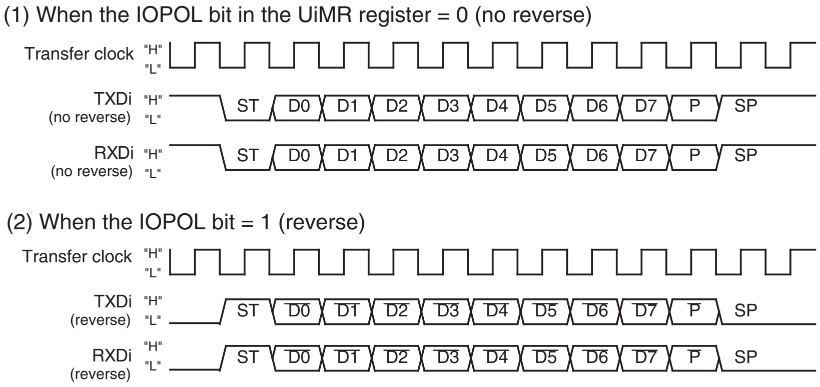
NOTE:
1. This applies to the case where the register bit are set as follows:
- CKPOL bit in UiC0 register = 0 (transmit data output at the falling edge of the transfer clock)
- UFORM bit in UiC0 register = 0 (LSB first)
- STPS bit in UiMR register = 0 (1 stop bit)
- PRYE bit in UiMR register = 1 (parity enabled)

Figure 15.20 Serial Data Logic Switching

15.1.2.5 TXD and RXD I/O Polarity Inverse Function

This function inverses the polarities of the TXDi pin output and RXDi pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inversed.

Figure 15.21 shows the TXD and RXD I/O Polarity Inverse.


Transfer clock
"H"
"L"
TXDi
(no reverse)
RXDi
(no reverse)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
Transfer clock
TXDi
(reverse)
RXDi
(reverse)
ST $\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$ $\overline{P}$ SP

(1) When the IOPOL bit in the UiMR register = 0 (no reverse)

(2) When the IOPOL bit = 1 (reverse)

i = 0 to 2
ST: Start bit
P: Parity bit
SP: Stop bit

NOTE:
1. This applies to the case where the register bits are set as follows:
- UFORM bit in UiC0 register = 0 (LSB first)
- STPS bit in UiMR register = 0 (1 stop bit)
- PRYE bit in UiMR register = 1 (parity enabled)

Figure 15.21 TXD and RXD I/O Polarity Inverse

15.1.2.6 CTS/RTS Function

When the CTS function is used transmit operation start when “L” is applied to the CTSi/RTSi (i = 0 to 2) pin. Transmit operation begins when the CTSi/RTSi pin is held “L”. If the “L” signal is switched to “H” during a transmit operation, the operation stops before the next data.

When the RTS function is used, the CTSi/RTSi pin outputs on “L” signal when the MCU is ready to receive. The output level becomes “H” on the first falling edge of the CLKi pin.

- CRD bit in UiC0 register = 1 (CTS/RTS function of UART0 disabled)
  CTSi/RTSi pin is programmable I/O function
- CRD bit = 0, CRS bit in UiC0 register= 0 (CTS function is selected)
  CTSi/RTSi pin is CTS function
- CRD bit = 0, CRS bit = 1 (RTS function is selected) CTSi/RTSi pin is RTS function

15.1.2.7 CTS/RTS Separate Function (UART0)

This function separates CTS0/RTS0, outputs RTS0 from the P6_0 pin, and accepts as input the CTS0 from the P6_4 pin. To use this function, set the register bits as shown below.

- CRD bit in U0C0 register = 0 (CTS/RTS of UART0 enabled)
- CRS bit in U0C0 register = 1 (output RTS of UART0)
- CRD bit in U1C0 register = 0 (CTS/RTS of UART1 enabled)
- CRS bit in U1C0 register = 0 (input CTS of UART1)
- RCSP bit in UCON register = 1 (input CTS0 from the P6_4 pin)
- CLKMD1 bit in UCON register = 0 (CLKS1 not used)

Note that when using the CTS/RTS separate function, CTS/RTS of UART1 separate function cannot be used.

Figure 15.22 shows CTS/RTS separate function usage.

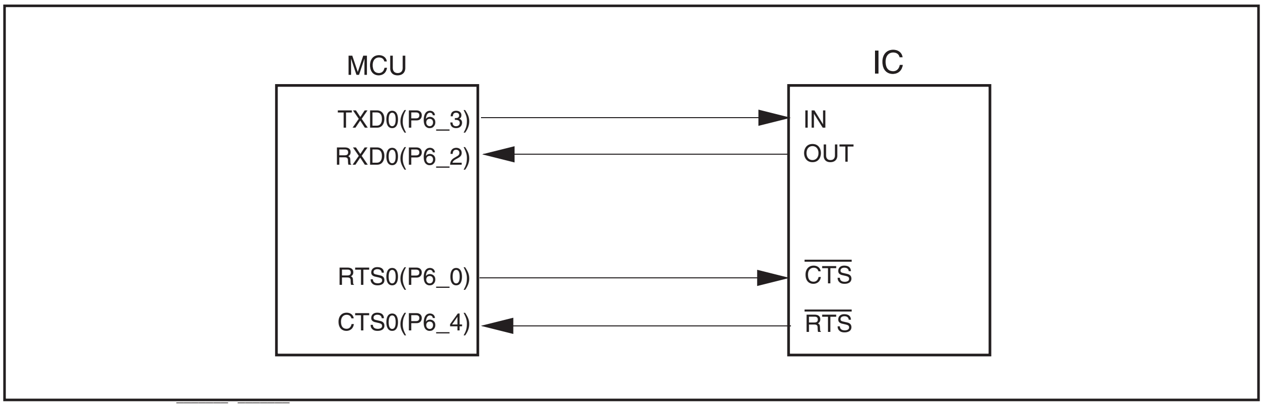


Figure 15.22 CTS/RTS Separate Function

15.1.3 Special Mode 1 (I²C Mode)

I²C mode is provided for use as a simplified I²C interface compatible mode. Table 15.10 lists the I²C Mode Specifications. Figure 15.23 shows the I²C Mode Block Diagram. Table 15.11 lists the Registers to be Used and Setting in I²C Mode. Table 15.12 lists the I²C Mode Functions. Figure 15.24 shows the Transfer to UiRB Register and Interrupt Timing.

As shown in Table 15.12, the MCU is placed in I²C mode by setting bits SMD2 to SMD0 to 010b and the IICM bit to 1. Because SDAi transmit output has a delay circuit attached, SDAi output does not change state until SCLi goes low and remains stably low.

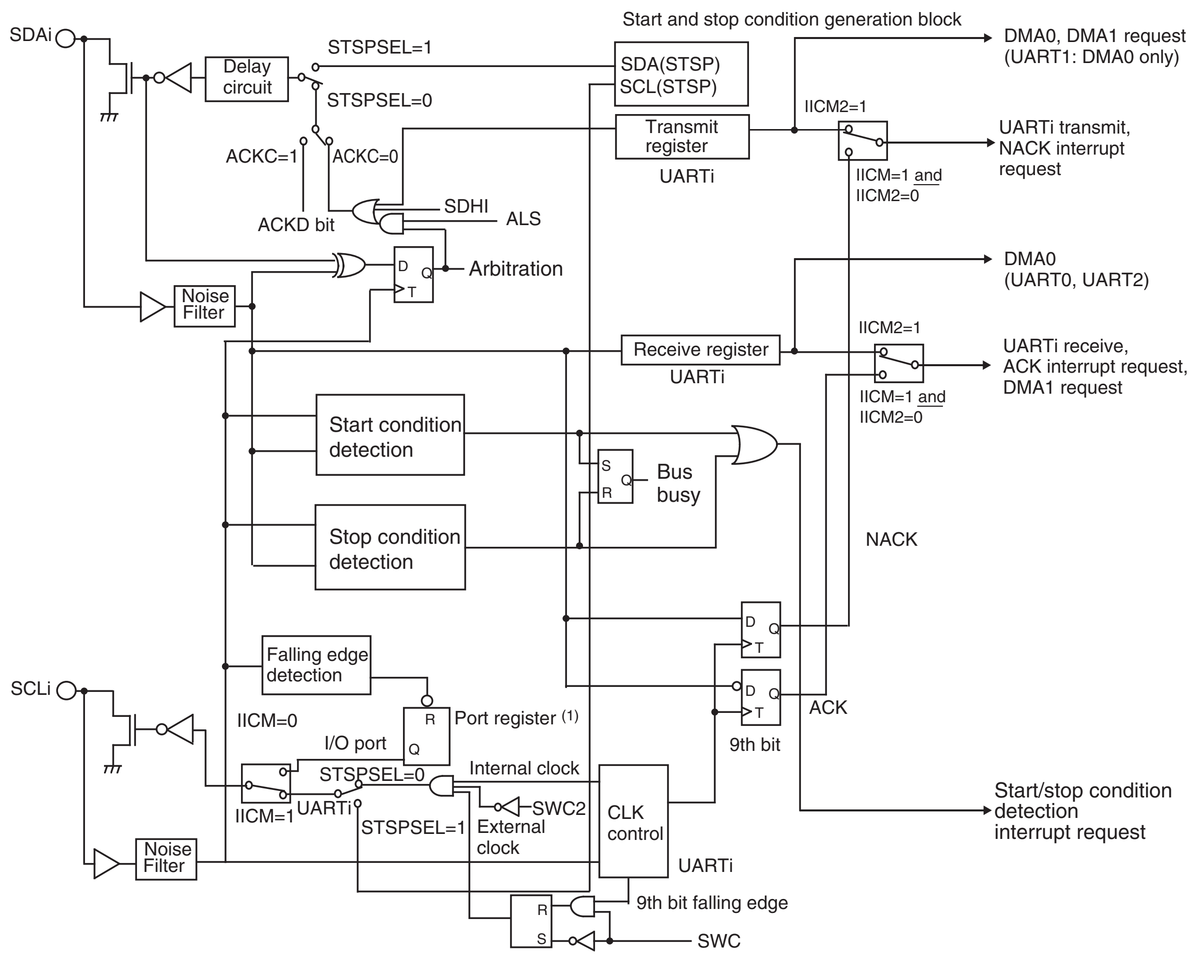
Table 15.10 I²C Mode Specifications

Item	Specification
Transfer data format	Transfer data length: 8 bits
Transfer clock	• During master The CKDIR bit in the UiMR register = 0 (internal clock) : fj/(2(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO. n: Setting value of the UiBRG register 00h to FFh • During slave The CKDIR bit = 1 (external clock) : Input from SCLi pin
Transmit start condition	Before transmission can start, meet the following requirements (1) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Receive start condition	Before reception can start, meet the following requirements (1) • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt request generation timing	When start or stop condition is detected, acknowledge undetected, and acknowledge detected
Error detection	Overrun error (2) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 8th bit of the next data
Select function	• Arbitration lost Timing at which the ABT bit in the UiRB register is updated can be selected • SDAi digital delay No digital delay or a delay of 2 to 8 UiBRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable

i = 0 to 2

NOTES:

1. When an external clock is selected, the conditions must be met while the external clock is in the high state.
2. If an overrun error occurs, the value of UiRB register will be undefined. The IR bit in the SiRIC register remains unchanged.



This diagram applies to the case where bits SMD2 to SMD0 in the UiMR register = 010b and the IICM bit in the UiSMR register = 1.

i = 0 to 2

IICM: Bit in UiSMR register
IICM2, SWC, ALS, SWC2, SDHI: Bits in UiSMR2 register
STSPSEL, ACKD, ACKC: Bits in UiSMR4 register

NOTE:
1. If the IICM bit =1, the pins can be read even when the PD6_2, PD6_6 or PD7_1 bit = 1 (output mode).

Figure 15.23 I^{2}C Mode Block Diagram

Table 15.11 Registers to Be Used and Settings in I^{2}C Mode

Register	Bit	Function	
		Master	Slave
UiTB (1)	0 to 7	Set transmit data	
UiRB (1)	0 to 7	Receive data can be read	
	8	ACK or NACK is set in this bit	
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	
UiBRG	0 to 7	Set a bit rate	Invalid
UiMR (1)	SMD2 to SMD0	Set to 010b	
	CKDIR	Set to 0	Set to 1
	IOPOL	Set to 0	
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register	Invalid
	CRS	Invalid because the CRD bit = 1	
	TXEPT	Transmit register empty flag	
	CRD (3)	Set to 1	
	NCH	Set to 1	
	CKPOL	Set to 0	
	UFORM	Set to 1	
UiC1	TE	Set this bit to 1 to enable transmission	
	TI	Transmit buffer empty flag	
	RE	Set this bit to 1 to enable reception	
	RI	Reception complete flag	
	U2IRS (2)	Invalid	
	U2RRM (2), UiLCH, UiERE	Set to 0	
UiSMR	IICM	Set to 1	
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	
	3 to 7	Set to 0	
UiSMR2	IICM2	See **Table 15.12 I^{2}C Mode Functions**	
	CSC	Set this bit to 1 to enable clock synchronization	Set to 0
	SWC	Set this bit to 1 to have SCLi output fixed to “L” at the falling edge of the 9th bit of clock	
	ALS	Set this bit to 1 to have SDAi output stopped when arbitration-lost is detected	Set to 0
	STAC	Set to 0	Set this bit to 1 to initialize UARTi at start condition detection
	SWC2	Set this bit to 1 to have SCLi output forcibly pulled low	
	SDHI	Set this bit to 1 to disable SDAi output	
	7	Set to 0	
UiSMR3	0, 2, 4, and NODC	Set to 0	
	CKPH	See **Table 15.12 I^{2}C Mode Functions**	
	DL2 to DL0	Set the amount of SDAi digital delay	
UiSMR4	STAREQ	Set this bit to 1 to generate start condition	Set to 0
	RSTAREQ	Set this bit to 1 to generate restart condition	Set to 0
	STPREQ	Set this bit to 1 to generate stop condition	Set to 0
	STSPSEL	Set this bit to 1 to output each condition	Set to 0
	ACKD	Select ACK or NACK	
	ACKC	Set this bit to 1 to output ACK data	
	SCLHI	Set this bit to 1 to have SCLi output stopped when stop condition is detected	Set to 0
	SWC9	Set to 0	Set this bit to 1 to set the SCLi to “L” hold at the falling edge of the 9th bit of clock
IFSR0	IFSR06, ISFR07	Set to 1	
UCON	U0IRS, U1IRS	Invalid	
	2 to 7	Set to 0	

i = 0 to 2

NOTES:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in I^{2}C mode.
2. Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
3. When using UART1 in I^{2}C mode and enabling the $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ separate function of UART0, set the CRD bit in the U1C0 register to 0 ($\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled) and the CRS bit to 0 ($\overline{\text{CTS}}$ input).

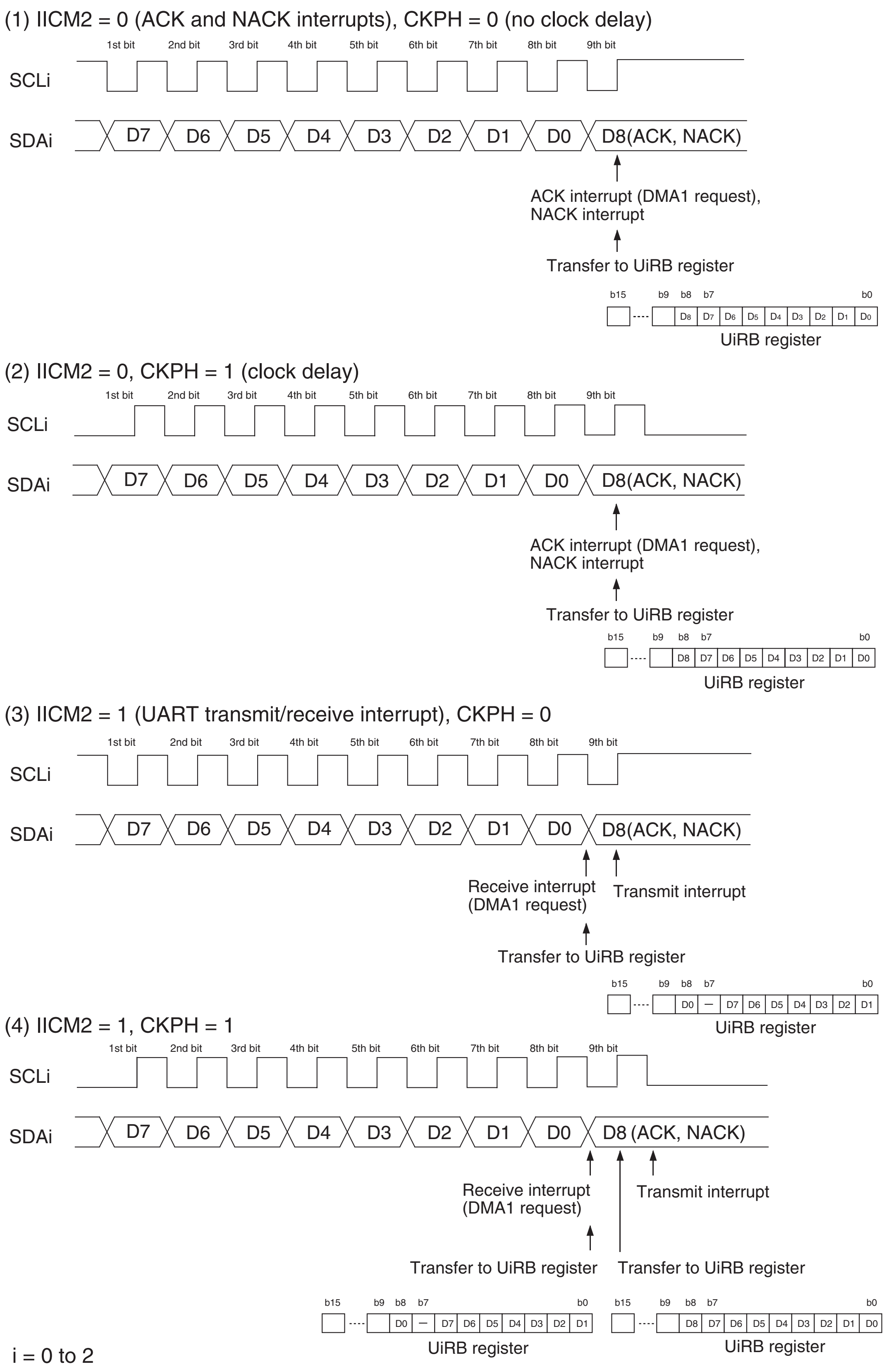
Table 15.12 I^{2}C Mode Functions

Function	Clock Synchronous Serial I/O Mode (SMD2 to SMD0 = 001b, IICM = 0)	I^{2}C Mode (SMD2 to SMD0 = 010b, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delay)
Source of interrupt number 6, 7, and 10 (1) (5) (7)	-	Start condition detection or stop condition detection (See Table **15.13 STSPSEL Bit Functions**)			
Source of interrupt number 15, 17, and 19 (1) (6)	UARTi transmission Transmission started or completed (selected by UiIRS)	No acknowledgment detection (NACK) Rising edge of SCLi 9th bit		UARTi transmission Rising edge of SCLi 9th bit	UARTi transmission Falling edge of SCLi next to the 9th bit
Source of interrupt number 16, 18, and 20 (1) (6)	UARTi reception When 8th bit received CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit		UARTi reception Falling edge of SCLi 9th bit	
Timing for transferring data from UART reception shift register to UiRB register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of SCLi 9th bit		Falling edge of SCLi 9th bit	Falling and rising edges of SCLi 9th bit
UARTi transmission output delay	Not delayed	Delayed			
Functions of pins P6_3, P6_7, and P7_0	TXDi output	SDAi input/output			
Functions of pins P6_2, P6_6, and P7_1	RXDi input	SCLi input/output			
Functions of pins P6_1, P6_5, and P7_2	CLKi input or output selected	- (Cannot be used in I^{2}C mode)			
Noise filter width	15 ns	200 ns			
Read RXDi and SCLi pins levels	Possible when the corresponding port direction bit = 0	Always possible no matter how the corresponding port direction bit is set			
Initial value of TXDi and SDAi outputs	CKPOL = 0 (H) CKPOL = 1 (L)	The value set in the port register before setting I^{2}C mode (2)			
Initial and end value of SCLi	-	H	L	H	L
DMA1 source (6)	UARTi reception	Acknowledgment detection (ACK)		UARTi reception Falling edge of SCLi 9th bit	
Store received data	1st to 8th bits of the received data are stored into bits 7 to 0 in the UiRB register			1st to 7th bits of the received data are stored into bits 6 to 0 in the UiRB register, 8th bit is stored into bit 8 in the UiRB register	1st to 8th bits are stored into bit 7 to bit 0 in UiRB register (3)
Read received data	The UiRB register status is read				Bit 6 to bit 0 in the UiRB register (4) are read as bit 7 to bit 1. Bit 8 in the UiRB register is read as bit 0.

i = 0 to 2

NOTES:

1. If the interrupt source is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). (Refer to **23.8 Interrupts**.)
   If one of the bits shown below is changed, the interrupt source, the interrupt timing, etc. change. Therefore, always be sure to set the IR bit to 0 (interrupt not requested) after changing those bits.
   - Bits SMD2 to SMD0 in UiMR register
   - IICM bit in UiSMR register
   - IICM2 bit in UiSMR2 register
   - CKPH bit in UiSMR3 register
2. Set the initial value of SDAi output while bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled).
3. Second data transfer to the UiRB register (rising edge of SCLi 9th bit)
4. First data transfer to the UiRB register (falling edge of SCLi 9th bit)
5. See **Figure 15.26 STSPSEL Bit Functions**.
6. See **Figure 15.24 Transfer to UiRB Register and Interrupt Timing**.
7. When using UART0, be sure to set the IFSR06 bit in the IFSR0 register to 1 (interrupt source: UART0 bus collision detection). When using UART1, be sure to set the IFSR07 bit in the IFSR0 register to 1 (interrupt source: UART1 bus collision detection).


(1) IICM2 = 0 (ACK and NACK interrupts), CKPH = 0 (no clock delay)
1st bit
2nd bit
3rd bit
4th bit
5th bit
6th bit
7th bit
8th bit
9th bit
SCLi
SDAi
D7
D6
D5
D4
D3
D2
D1
D0
D8(ACK, NACK)
ACK interrupt (DMA1 request), NACK interrupt
Transfer to UiRB register
b15
b9
b8
b7
b0
D8 D7 D6 D5 D4 D3 D2 D1 D0
UiRB register
(2) IICM2 = 0, CKPH = 1 (clock delay)
SCLi
SDAi
D8(ACK, NACK)
ACK interrupt (DMA1 request), NACK interrupt
Transfer to UiRB register
D8 D7 D6 D5 D4 D3 D2 D1 D0
UiRB register
(3) IICM2 = 1 (UART transmit/receive interrupt), CKPH = 0
SCLi
SDAi
D8(ACK, NACK)
Receive interrupt (DMA1 request)
Transmit interrupt
Transfer to UiRB register
D0 — D7 D6 D5 D4 D3 D2 D1
UiRB register
(4) IICM2 = 1, CKPH = 1
SCLi
SDAi
D8 (ACK, NACK)
Receive interrupt (DMA1 request)
Transmit interrupt
Transfer to UiRB register
Transfer to UiRB register
D0 — D7 D6 D5 D4 D3 D2 D1
UiRB register
D8 D7 D6 D5 D4 D3 D2 D1 D0
UiRB register

i = 0 to 2

This diagram applies to the case where the following condition is met.
・The CKDIR bit in the UiMR register = 0 (slave selected)

Figure 15.24 Transfer to UiRB Register and Interrupt Timing

15.1.3.1 Detection of Start and Stop Condition

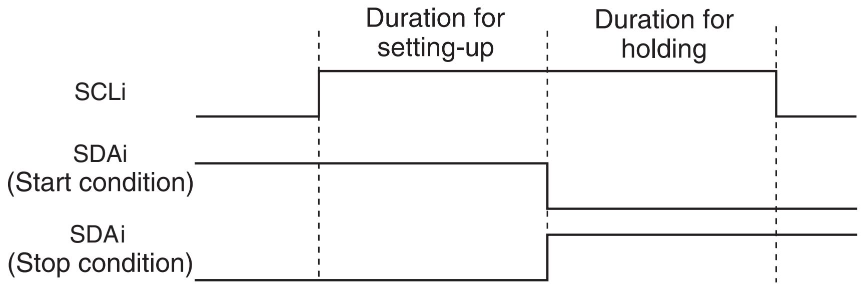
Whether a start or a stop condition has been detected is determined.

A start condition-detected interrupt request is generated when the SDAi pin changes state from high to low while the SCLi pin is in the high state. A stop condition-detected interrupt request is generated when the SDAi pin changes state from low to high while the SCLi pin is in the high state.

Figure 15.25 shows the Detection of Start and Stop Condition.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the BBS bit in the UiSMR register to determine which interrupt source is requesting the interrupt.

3 to 6 cycles < duration for setting-up (1)
3 to 6 cycles < duration for holding (1)



i = 0 to 2

NOTE:
1. When the PCLK1 bit in the PCLKR register = 1, this is the cycle number of f1SIO, and when the PCLK1 bit = 0, this is the cycle number of f2SIO.

Figure 15.25 Detection of Start and Stop Condition

15.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the UiSMR4 register (i = 0 to 2) to 1 (start).

A restart condition is generated by setting the RSTAREQ bit in the UiSMR4 register to 1 (start).

A stop condition is generated by setting the STPREQ bit in the UiSMR4 register to 1 (start).

The output procedure is described below.

(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to 1 (start).

(2) Set the STSPSEL bit in the UiSMR4 register to 1 (output).

Table 15.13 and Figure 15.26 show the STSPSEL Bit Functions.

Table 15.13 STSPSEL Bit Functions

Function	STSPSEL Bit = 0	STSPSEL Bit = 1
Output of pins SCLi and SDAi	Output of transfer clock and data Output of start/stop condition is accomplished by a program using ports (not automatically generated in hardware)	Output of a start/stop condition depending on bits STAREQ, RSTAREQ, and STPREQ
Start/stop condition interrupt request generation timing	Start/stop condition detection	Finish generating start/stop condition

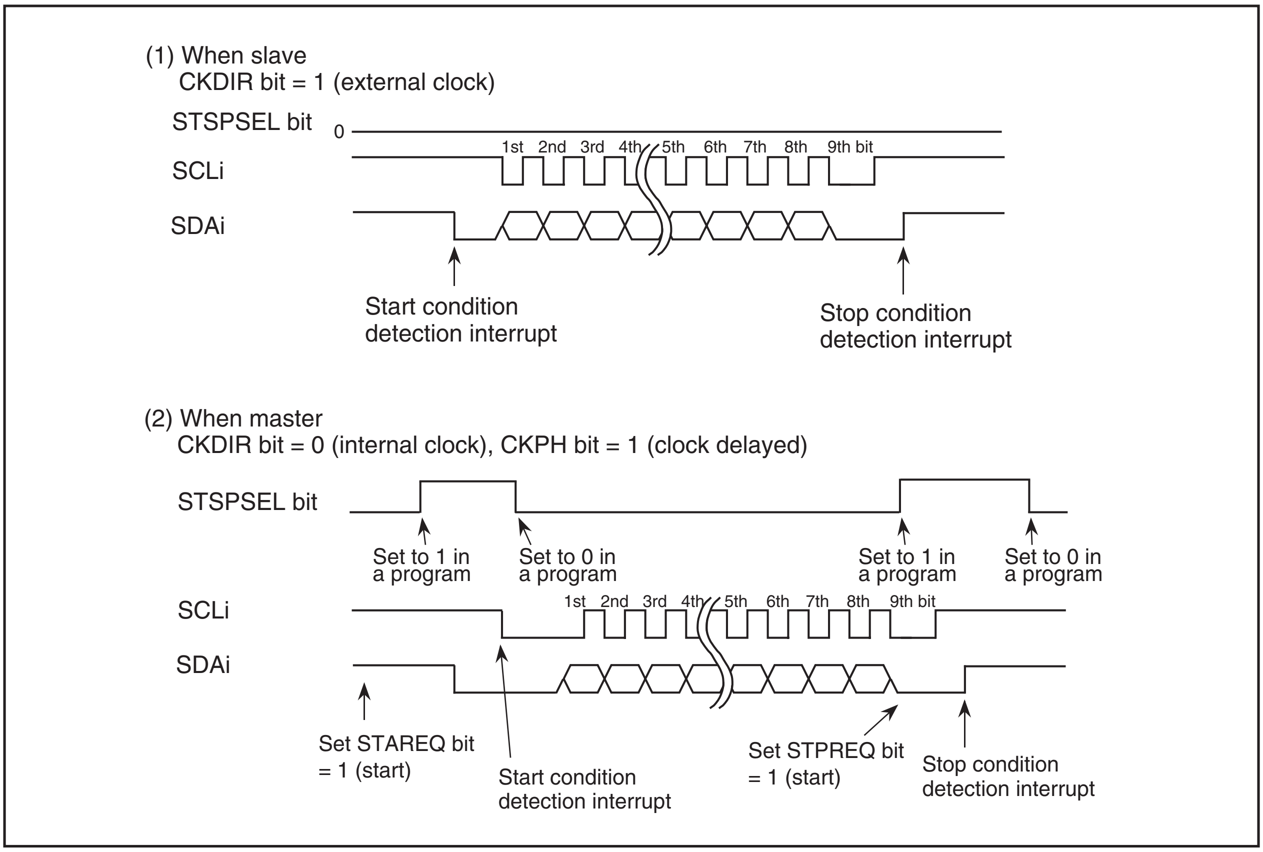

(1) When slave
CKDIR bit = 1 (external clock)
STSPSEL bit
0
SCLi
SDAi
1st 2nd 3rd 4th 5th 6th 7th 8th 9th bit
Start condition detection interrupt
Stop condition detection interrupt
(2) When master
CKDIR bit = 0 (internal clock), CKPH bit = 1 (clock delayed)
STSPSEL bit
Set to 1 in a program
Set to 0 in a program
Set to 1 in a program
Set to 0 in a program
SCLi
SDAi
1st 2nd 3rd 4th 5th 6th 7th 8th 9th bit
Set STAREQ bit = 1 (start)
Start condition detection interrupt
Set STPREQ bit = 1 (start)
Stop condition detection interrupt

Figure 15.26 STSPSEL Bit Functions

15.1.3.3 Arbitration

Unmatching of the transmit data and SDAi pin input data is checked synchronously with the rising edge of SCLi. Use the ABC bit in the UiSMR register to select the timing at which the ABT bit in the UiRB register is updated. If the ABC bit = 0 (updated per bit), the ABT bit is set to 1 at the same time unmatching is detected during check, and is set to 0 when not detected. In cases when the ABC bit is set to 1, if unmatching is detected even once during check, the ABT bit is set to 1 (unmatching detected) at the falling edge of the clock pulse of 9th bit. If the ABT bit needs to be updated per byte, set the ABT bit to 0 (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the ALS bit in the UiSMR2 register to 1 (SDA output stop enabled) causes arbitration-lost to occur, in which case the SDAi pin is placed in the high-impedance state at the same time the ABT bit is set to 1 (unmatching detected).

15.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in Figure 15.24 Transfer to UiRB Register and Interrupt Timing.

The CSC bit in the UiSMR2 register is used to synchronize the internally generated clock (internal SCLi) and an external clock supplied to the SCLi pin. In cases when the CSC bit is set to 1 (clock synchronization enabled), if a falling edge on the SCLi pin is detected while the internal SCLi is high, the internal SCLi goes low, at which time the value of the UiBRG register is reloaded with and starts counting in the low-level interval. If the internal SCLi changes state from low to high while the SCLi pin is low, counting stops, and when the SCLi pin goes high, counting restarts.

In this way, the UARTi transfer clock is comprised of the logical product of the internal SCLi and SCLi pin signal. The transfer clock works from a half period before the falling edge of the internal SCLi 1st bit to the rising edge of the 9th bit. To use this function, select an internal clock for the transfer clock.

The SWC bit in the UiSMR2 register allows to select whether the SCLi pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the SCLHI bit in the UiSMR4 register is set to 1 (enabled), SCLi output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the UiSMR2 register = 1 (0 output) makes it possible to forcibly output a low-level signal from the SCLi pin even while sending or receiving data. Setting the SWC2 bit to 0 (transfer clock) allows the transfer clock to be output from or supplied to the SCLi pin, instead of outputting a low-level signal.

If the SWC9 bit in the UiSMR4 register is set to 1 (SCL hold low enabled) when the CKPH bit in the UiSMR3 register = 1, the SCLi pin is fixed to low-level output at the falling edge of the clock pulse next to the 9th. Setting the SWC9 bit = 0 (SCL hold low disabled) frees the SCLi pin from low-level output.

15.1.3.5 SDA Output

The data written to bits 7 to 0 (D7 to D0) in the UiTB register is sequentially output beginning with D7. The 9th bit (D8) is ACK or NACK.

The initial value of SDAi transmit output can only be set when IICM = 1 (I^2C mode) and bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled).

Bits DL2 to DL0 in the UiSMR3 register allow to add no delays or a delay of 2 to 8 UiBRG count source clock cycles to SDAi output.

Setting the SDHI bit in the UiSMR2 register = 1 (SDA output disabled) forcibly places the SDAi pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UARTi transfer clock. This is because the ABT bit may inadvertently be set to 1 (detected).

15.1.3.6 SDA Input

When the IICM2 bit = 0, 1st to 8th bits (D7 to D0) of receive data are stored in bits 7 to 0 in the UiRB register. The 9th bit (D8) is ACK or NACK.

When the IICM2 bit = 1, the 1st to 7th bits (D7 to D1) of receive data are stored in bits 6 to 0 in the UiRB register and the 8th bit (D0) is stored in the bit 8 in the UiRB register. Even when the IICM2 bit = 1, providing the CKPH bit = 1, the same data as when the IICM2 bit = 0 can be read out by reading the UiRB register after the rising edge of the corresponding clock pulse of 9th bit.

15.1.3.7 ACK and NACK

If the STSPSEL bit in the UiSMR4 register is set to 0 (start and stop conditions not generated) and the ACKC bit in the UiSMR4 register is set to 1 (ACK data output), the value of the ACKD bit in the UiSMR4 register is output from the SDAi pin.

If the IICM2 bit = 0, a NACK interrupt request is generated if the SDAi pin remains high at the rising edge of the 9th bit of transmit clock pulse. An ACK interrupt request is generated if the SDAi pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACKi is selected for the DMA1 request source, a DMA transfer can be activated by detection of an acknowledge.

15.1.3.8 Initialization of Transmission/Reception

If a start condition is detected while the STAC bit = 1 (UARTi initialization enabled), the serial interface operates as described below.

- The transmit shift register is initialized, and the content of the UiTB register is transferred to the transmit shift register. In this way, the serial interface starts transmitting data synchronously with the next clock pulse applied. However, the UARTi output value does not change state and remains the same as when a start condition was detected until the first bit of data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial interface starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to 1 (SCL wait output enabled). Consequently, the SCLi pin is pulled low at the falling edge of the 9th clock pulse.

Note that when UARTi transmission/reception is started using this function, the TI bit does not change state. Note also that when using this function, the selected transfer clock should be an external clock.

15.1.4 Special Mode 2

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. Table 15.14 lists the Special Mode 2 Specifications. Figure 15.27 shows the Serial Bus Communication Control Example (UART2). Table 15.15 lists the Registers to be Used an Settings in Special Mode 2.

Table 15.14 Special Mode 2 Specifications

Item	Specification
Transfer data format	Transfer data length: 8 bits
Transfer clock	• Master mode The CKDIR bit in the UiMR register = 0 (internal clock) : fj/(2(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO. n: Setting value of the UiBRG register 00h to FFh • Slave mode The CKDIR bit = 1 (external clock selected) : Input from CLKi pin
Transmit/receive control	Controlled by input/output ports
Transmit start condition	Before transmission can start, meet the following requirements [1] • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Receive start condition	Before reception can start, meet the following requirements [1] • The RE bit in the UiC1 register = 1 (reception enabled) • The TE bit in the UiC1 register = 1 (transmission enabled) • The TI bit in the UiC1 register = 0 (data present in the UiTB register)
Interrupt request generation timing	For transmission, one of the following conditions can be selected • The UiIRS bit [2] = 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) • The UiIRS bit =1 (transmission completed): when the serial interface finished transmitting data from the UARTi transmit register For reception • When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	Overrun error [3] This error occurs if the serial interface started receiving the next data before reading the UiRB register and received the 7th bit of the next data
Select function	Clock phase setting Selectable from four combinations of transfer clock polarities and phases

i = 0 to 2

NOTES:

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. Bits U0IRS and U1IRS are bits 0 and 1 in the UCON register ; the U2IRS bit is bit 4 in the U2C1 register.
3. If an overrun error occurs, the value of UiRB register will be undefined. The IR bit in SiRIC register remains unchanged.

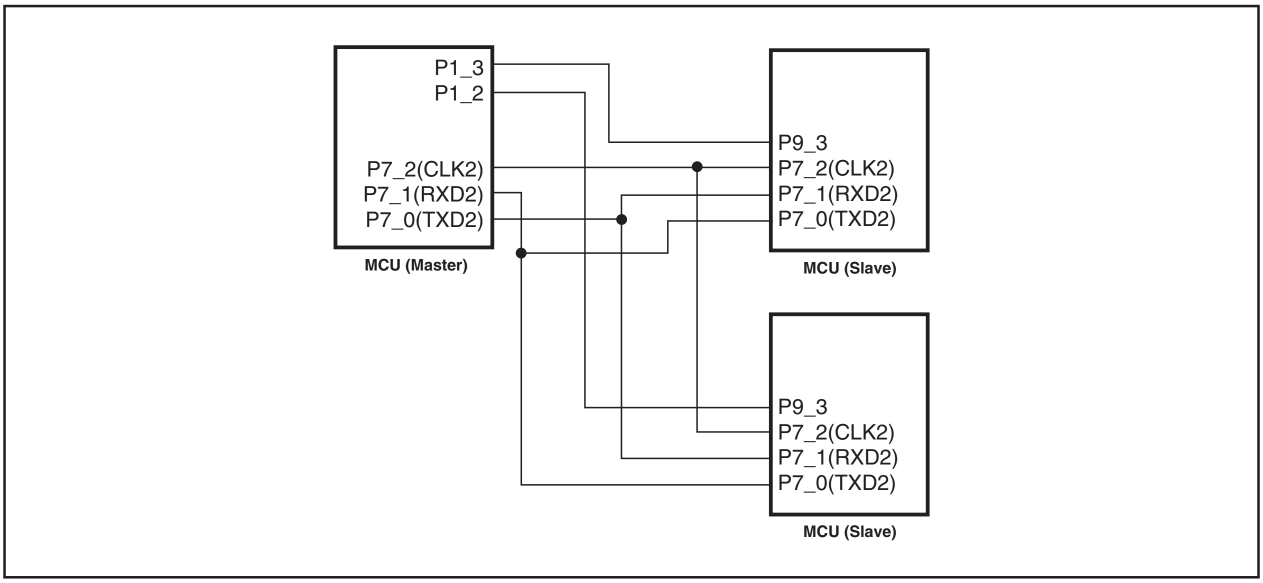


Figure 15.27 Serial Bus Communication Control Example (UART2)

Table 15.15 Registers to Be Used and Settings in Special Mode 2

Register	Bit	Function
UiTB [1]	0 to 7	Set transmit data
UiRB [1]	0 to 7	Receive data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a bit rate
UiMR [1]	SMD2 to SMD0	Set to 001b
	CKDIR	Set this bit to 0 for master mode or 1 for slave mode
	IOPOL	Set to 0
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Invalid because the CRD bit = 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TXDi pin output format
	CKPOL	Clock phases can be set in combination with the CKPH bit in the UiSMR3 register
	UFORM	Set to 0
UiC1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS [2]	Select the UART2 transmit interrupt source
	U2RRM [2], UiLCH, UiERE	Set to 0
UiSMR	0 to 7	Set to 0
UiSMR2	0 to 7	Set to 0
UiSMR3	CKPH	Clock phases can be set in combination with the CKPOL bit in the UiC0 register
	NODC	Set to 0
	0, 2, 4 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
UCON	U0IRS, U1IRS	Select the UART0 and UART1 transmit interrupt source
	U0RRM, U1RRM	Set to 0
	CLKMD0	Invalid because the CLKMD1 bit = 0
	CLKMD1, RCSP, 7	Set to 0

i = 0 to 2

NOTES:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in Special Mode 2.
2. Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.

15.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the UiSMR3 register and the CKPOL bit in the UiC0 register.

Make sure the transfer clock polarity and phase are the same for the master and salves to be communicated.

Figure 15.28 shows the Transmission and Reception Timing in Master Mode (internal clock).

Figure 15.29 shows the Transmission and Reception Timing (CKPH = 0) in Slave Mode (external clock).

Figure 15.30 shows the Transmission and Reception Timing (CKPH = 1) in Slave Mode (external clock).

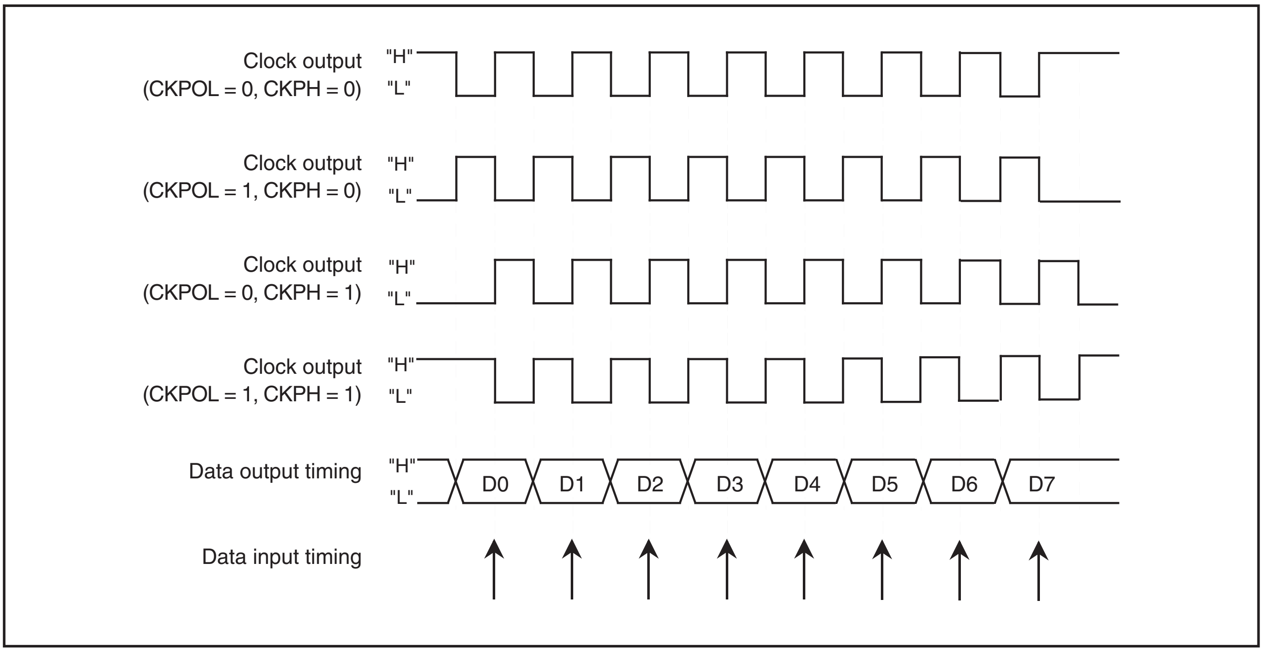


Figure 15.28 Transmission and Reception Timing in Master Mode (Internal Clock)

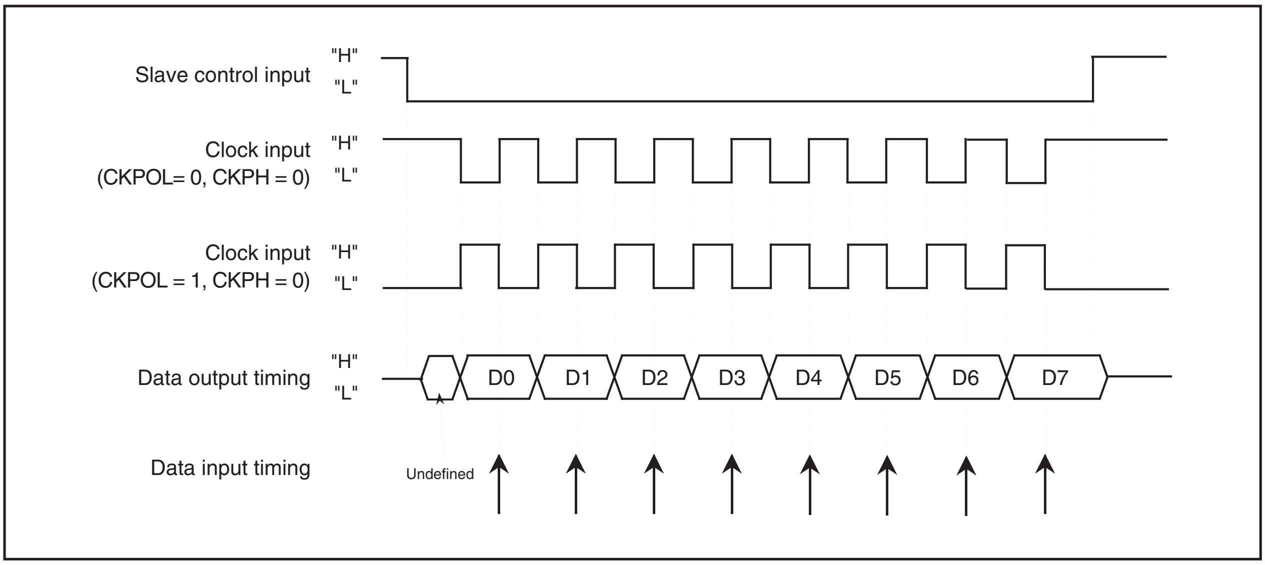


Figure 15.29 Transmission and Reception Timing (CKPH = 0) in Slave Mode (External Clock)

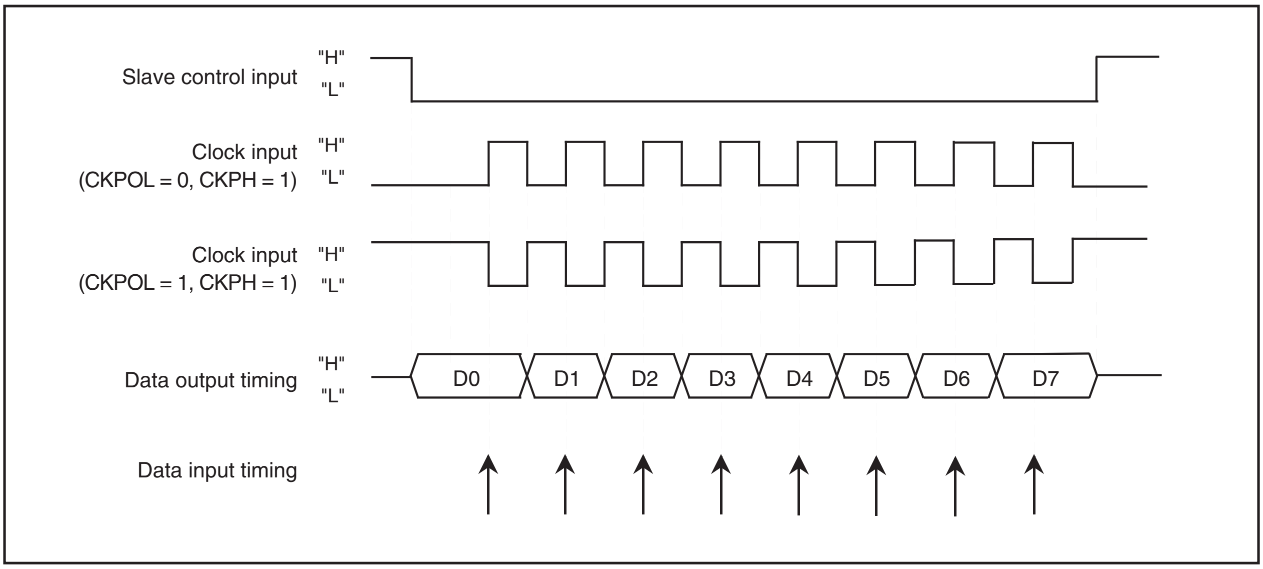


Figure 15.30 Transmission and Reception Timing (CKPH = 1) in Slave Mode (External Clock)

15.1.5 Special Mode 3 (IE Mode)

In this mode, one bit of IEBus is approximated with one byte of UART mode waveform.

Table 15.16 lists the Registers to be Used and Settings in IE mode. Figure 15.31 shows the Bus Collision Detect Function-Related Bits.

If the TXDi pin (i = 0 to 2) output level and RXDi pin input level do not match, a UARTi bus collision detect interrupt request is generated.

Use bits IFSR06 and IFSR07 in the IFSR0 register to enable the UART0/UART1 bus collision detect function.

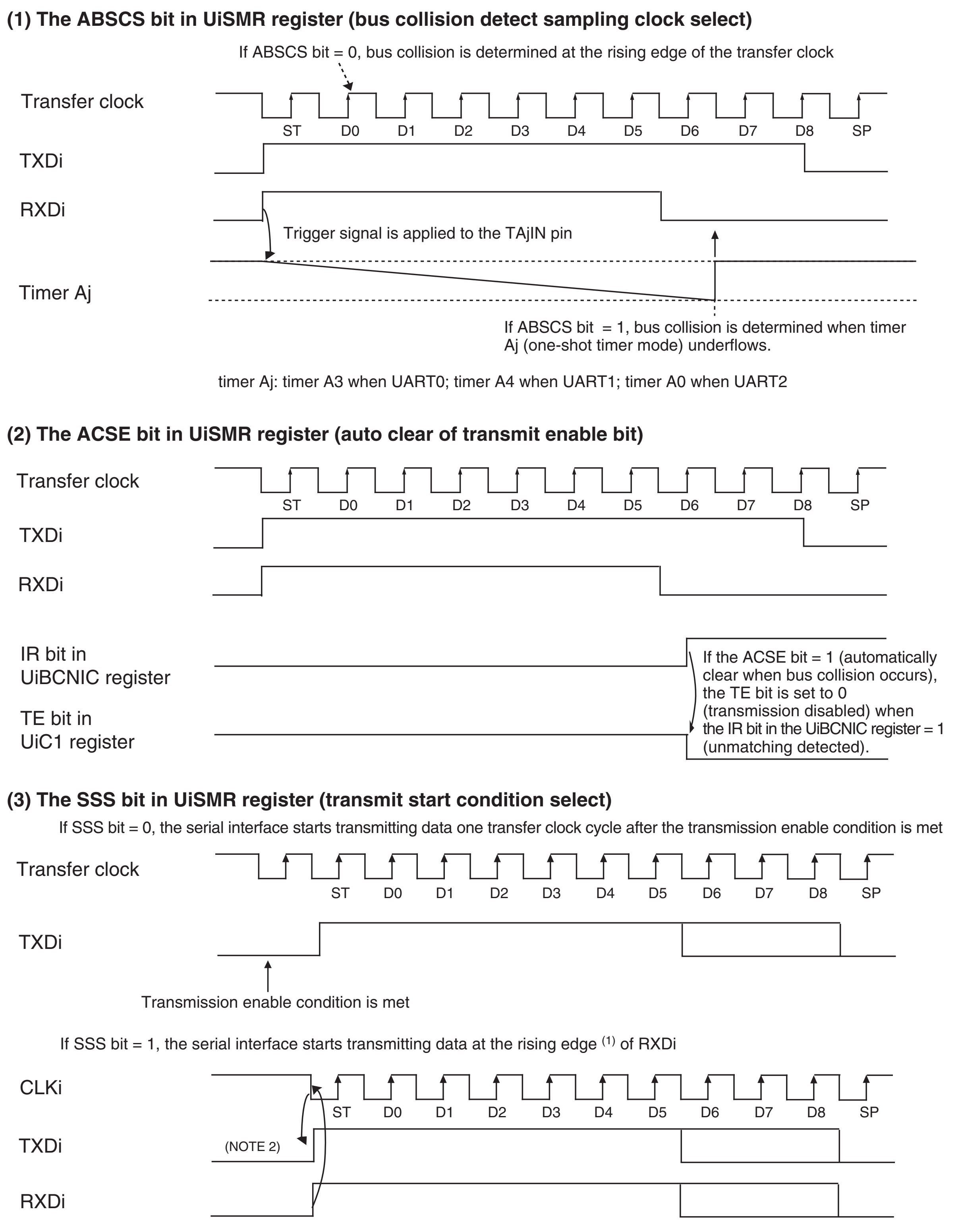
Table 15.16 Registers to Be Used and Settings in IE Mode

Register	Bit	Function
UiTB	0 to 8	Set transmit data
UiRB (1)	0 to 8	Receive data can be read
	OER,FER,PER,SUM	Error flag
UiBRG	0 to 7	Set a bit rate
UiMR	SMD2 to SMD0	Set to 110b
	CKDIR	Select the internal clock or external clock
	STPS	Set to 0
	PRY	Invalid because the PRYE bit = 0
	PRYE	Set to 0
	IOPOL	Select the TXD/RXD input/output polarity
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Invalid because the CRD bit = 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TXDi pin output mode
	CKPOL	Set to 0
	UFORM	Set to 0
UiC1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS (2)	Select the UART2 transmit interrupt source
	U2RRM (2), UiLCH, UiERE	Set to 0
UiSMR	0 to 3, 7	Set to 0
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to 1 to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
UiSMR2	0 to 7	Set to 0
UiSMR3	0 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
IFSR0	IFSR06, IFSR07	Set to 1
UCON	U0IRS, U1IRS	Select the UART0/UART1 transmit interrupt source
	U0RRM, U1RRM	Set to 0
	CLKMD0	Invalid because the CLKMD1 bit = 0
	CLKMD1, RCSP, 7	Set to 0

i= 0 to 2

NOTES:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in IE mode.
2. Set bits 4 and 5 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.


(1) The ABSCS bit in UiSMR register (bus collision detect sampling clock select)

(2) The ACSE bit in UiSMR register (auto clear of transmit enable bit)

(3) The SSS bit in UiSMR register (transmit start condition select)

If ABSCS bit = 0, bus collision is determined at the rising edge of the transfer clock
Transfer clock
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP
TXDi
RXDi
Trigger signal is applied to the TAjIN pin
Timer Aj
If ABSCS bit = 1, bus collision is determined when timer Aj (one-shot timer mode) underflows.
timer Aj: timer A3 when UART0; timer A4 when UART1; timer A0 when UART2
Transfer clock
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP
TXDi
RXDi
IR bit in UiBCNIC register
TE bit in UiC1 register
If the ACSE bit = 1 (automatically clear when bus collision occurs), the TE bit is set to 0 (transmission disabled) when the IR bit in the UiBCNIC register = 1 (unmatching detected).
If SSS bit = 0, the serial interface starts transmitting data one transfer clock cycle after the transmission enable condition is met
Transfer clock
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP
TXDi
Transmission enable condition is met
If SSS bit = 1, the serial interface starts transmitting data at the rising edge (1) of RXDi
CLKi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP
TXDi
(NOTE 2)
RXDi

NOTES:

1. The falling edge of RXDi when IOPOL bit = 0; the rising edge of RXDi when IOPOL bit = 1.
2. The transmit condition must be met before the falling edge (1) of RXDi.

i = 0 to 2
This diagram applies to the case where IOPOL bit =1 (reversed).

Figure 15.31 Bus Collision Detect Function-Related Bits

15.1.6 Special Mode 4 (SIM Mode) (UART2)

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows to output a low from the TXD2 pin when a parity error is detected. Table 15.17 lists the SIM Mode Specifications. Table 15.18 lists the Registers to be Used and Settings in SIM Mode. Figure 15.32 shows the Transmit and Receive Riming in SIM Mode.

Table 15.17 SIM Mode Specifications

Item	Specification
Transfer data format	• Direct format • Inverse format
Transfer clock	• The CKDIR bit in the U2MR register = 0 (internal clock) : fi/(16(n+1)) fi = f1SIO, f2SIO, f8SIO, f32SIO. n: Setting value of the U2BRG register 00h to FFh • The CKDIR bit = 1 (external clock) : fEXT/(16(n+1)) fEXT: Input from CLK2 pin. n: Setting value of the U2BRG register 00h to FFh
Transmit start condition	Before transmission can start, meet the following requirements • The TE bit in the U2C1 register = 1 (transmission enabled) • The TI bit in the U2C1 register = 0 (data present in the U2TB register)
Receive start condition	Before reception can start, meet the following requirements • The RE bit in the U2C1 register = 1 (reception enabled) • Start bit detection
Interrupt request generation timing [2]	• For transmission When the serial interface finished sending data from the U2TB transfer register (U2IRS bit = 1) • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error detection	• Overrun error [1] This error occurs if the serial interface started receiving the next data before reading the U2RB register and received the bit one before the last stop bit of the next data • Framing error [3] This error occurs when the number of stop bits set is not detected • Parity error [3] During reception, if a parity error is detected, parity error signal is output from the TXD2 pin. During transmission, a parity error is detected by the level of input to the RXD2 pin when a transmission interrupt occurs • Error sum flag This flag is set to 1 when any of the overrun, framing, and parity errors is encountered

NOTES:

1. If an overrun error occurs, the value of the U2RB register will be undefined. The IR bit in the S2RIC register remains unchanged.
2. A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to 1 (transmission completed) and U2ERE bit in the U2C1 register to 1 (error signal output) after reset. Therefore, when using SIM mode, set the IR bit to 0 (interrupt not requested) after setting these bits.
3. The timing at which the framing error flag and the parity error flag are set is detected when data is transferred from the UARTi receive register to the UiRB register.

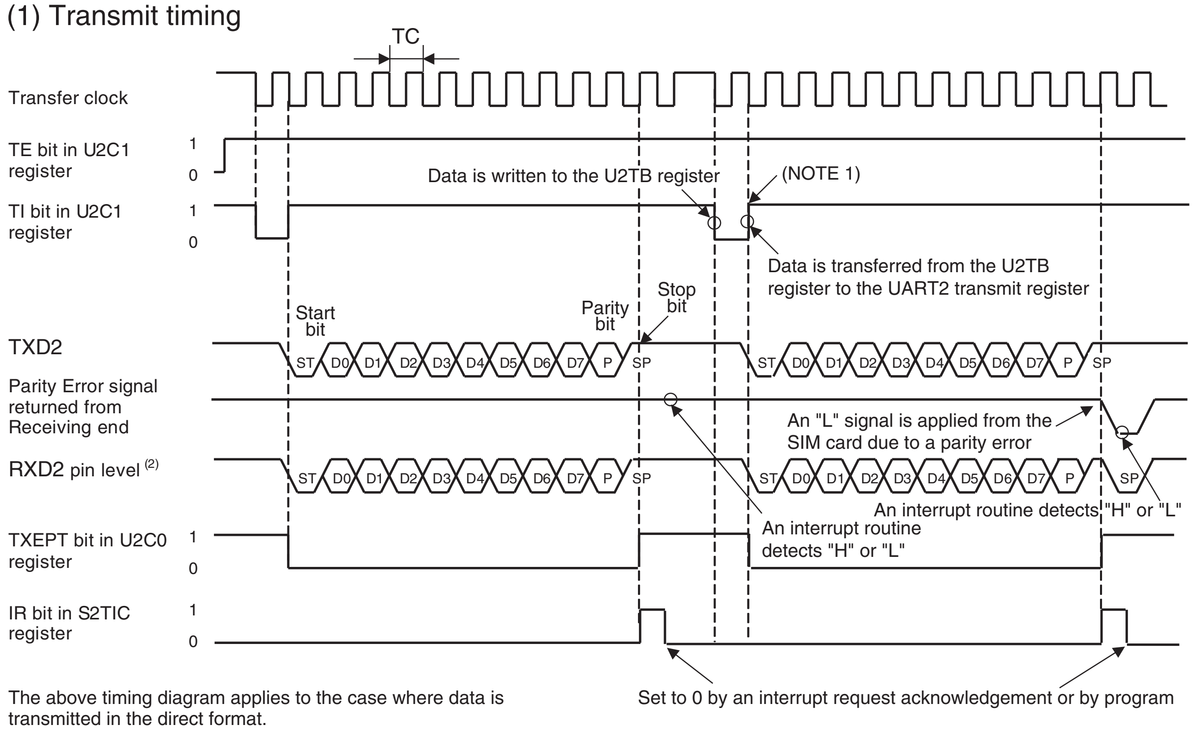
Table 15.18 Registers to Be Used and Settings in SIM Mode

Register	Bit	Function
U2TB [1]	0 to 7	Set transmit data
U2RB [1]	0 to 7	Receive data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set a bit rate
U2MR	SMD2 to SMD0	Set to 101b
	CKDIR	Select the internal clock or external clock
	STPS	Set to 0
	PRY	Set this bit to 1 for direct format or 0 for inverse format
	PRYE	Set to 1
	IOPOL	Set to 0
U2C0	CLK1 to CLK0	Select the count source for the U2BRG register
	CRS	Invalid because the CRD bit = 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Set to 0
	CKPOL	Set to 0
	UFORM	Set this bit to 0 for direct format or 1 for inverse format
U2C1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS	Set to 1
	U2RRM	Set to 0
	U2LCH	Set this bit to 0 for direct format or 1 for inverse format
	U2ERE	Set to 1
U2SMR [1]	0 to 3	Set to 0
U2SMR2	0 to 7	Set to 0
U2SMR3	0 to 7	Set to 0
U2SMR4	0 to 7	Set to 0

NOTE:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in SIM mode.

(1) Transmit timing

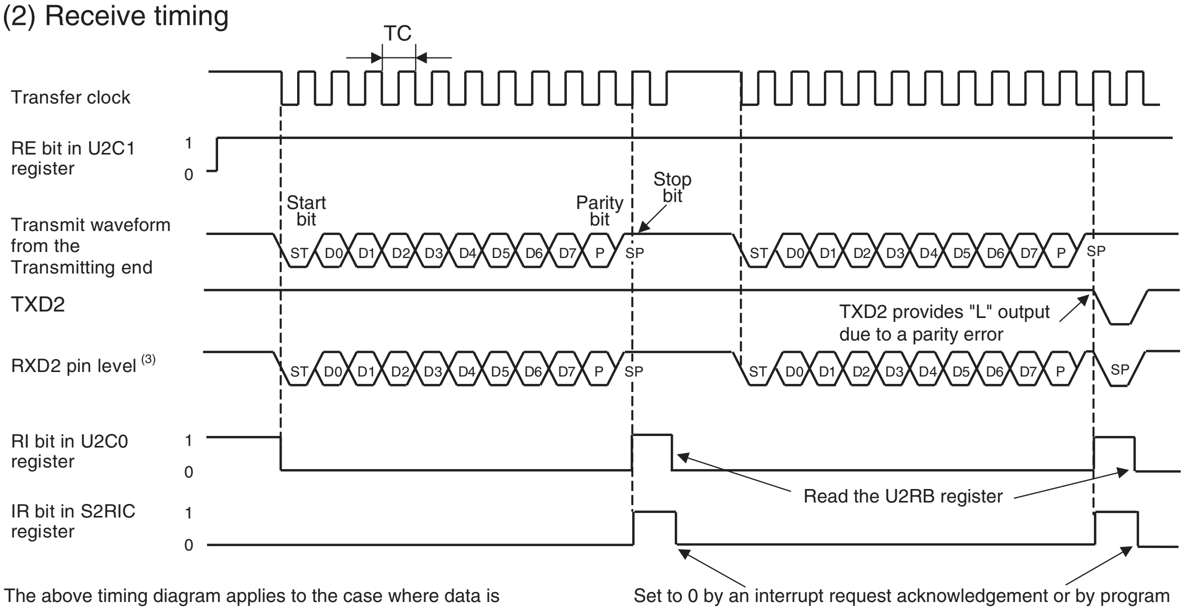

TC
Transfer clock
TE bit in U2C1 register
1
0
TI bit in U2C1 register
1
0
Data is written to the U2TB register
(NOTE 1)
Data is transferred from the U2TB register to the UART2 transmit register
Start bit
Parity bit
Stop bit
TXD2
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
Parity Error signal returned from Receiving end
An "L" signal is applied from the SIM card due to a parity error
RXD2 pin level (2)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
An interrupt routine detects "H" or "L"
An interrupt routine detects "H" or "L"
TXEPT bit in U2C0 register
1
0
IR bit in S2TIC register
1
0
Set to 0 by an interrupt request acknowledgement or by program

The above timing diagram applies to the case where data is transmitted in the direct format.

- STPS bit in U2MR register = 0 (1 stop bit)
- PRY bit in U2MR register = 1 (even)
- UFORM bit in U2C0 register = 0 (LSB first)
- U2LCH bit in U2C1 register = 0 (no reverse)
- U2IRS bit in U2C1 register = 1 (transmission completed)

TC = 16(n+1) / fi or 16(n+1) / fEXT
fi: frequency of U2BRG count source
(f1SIO, f2SIO, f8SIO, f32SIO)
fEXT: frequency of U2BRG count source (external clock)
n: value set to the U2BRG register

(2) Receive timing


TC
Transfer clock
RE bit in U2C1 register
1
0
Start bit
Parity bit
Stop bit
Transmit waveform from the Transmitting end
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
TXD2
TXD2 provides "L" output due to a parity error
RXD2 pin level (3)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
RI bit in U2C0 register
1
0
Read the U2RB register
IR bit in S2RIC register
1
0
Set to 0 by an interrupt request acknowledgement or by program

The above timing diagram applies to the case where data is received in the direct format.

- STPS bit in U2MR register = 0 (1 stop bit)
- PRY bit in U2MR register = 1 (even)
- UFORM bit in U2C0 register = 0 (LSB first)
- U2LCH bit In U2C1 register = 0 (no reverse)
- U2IRS bit in U2C1 register = 1 (transmission completed)

TC = 16(n+1) / fi or 16(n+1) / fEXT
fi: frequency of U2BRG count source
(f1SIO, f2SIO, f8SIO, f32SIO)
fEXT: frequency of U2BRG count source (external clock)
n: value set to the U2BRG register

NOTES:
1. Data transmission starts when BRG overflows after a value is set to the U2TB register on the rising edge of the TI bit.
2. Because the TXD2 and RXD2 pins are connected, a composite waveform, consisting of transmit waveform from the TXD2 pin and parity error signal from the receiving end, is generated.
3. Because the TXD2 and RxD2 pins are connected, a composite waveform, consisting of transmit waveform from the transmitting end and parity error signal from the TXD2 pin, is generated.

Figure 15.32 Transmit and Receive Timing in SIM Mode

Figure 15.33 shows the SIM Interface Connection. Connect TXD2 and RXD2 and apply pull-up.

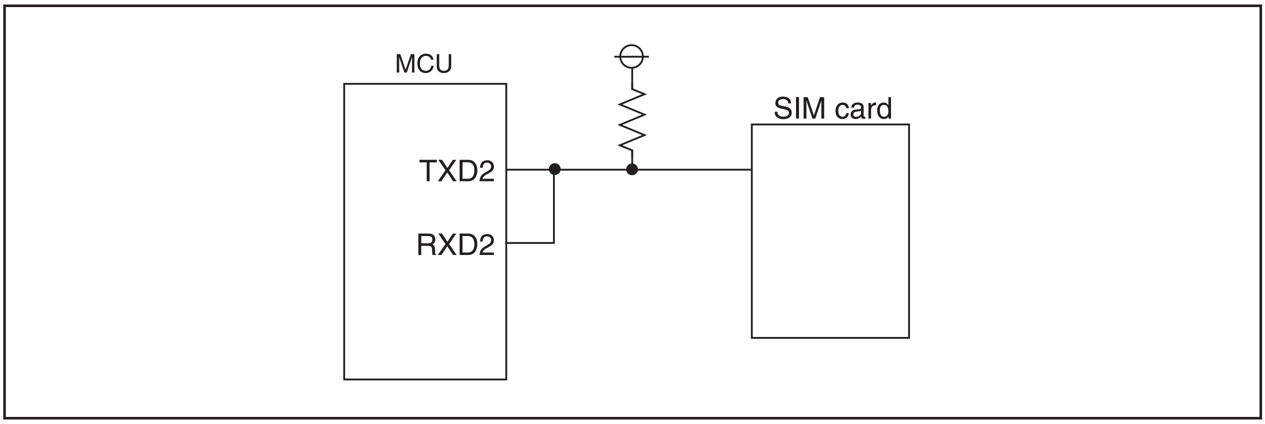


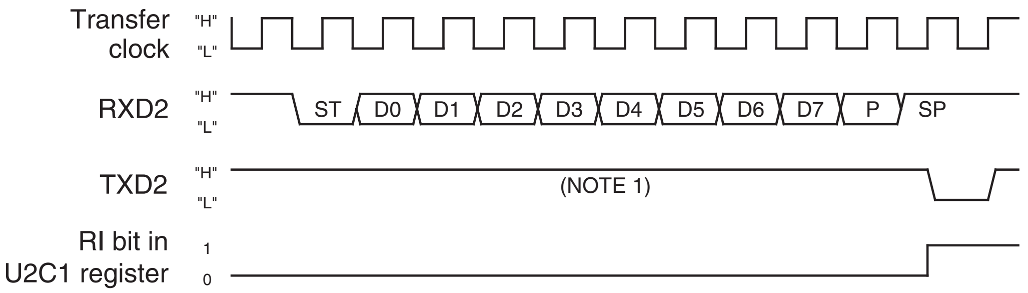
Figure 15.33 SIM Interface Connection

15.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in the U2C1 register to 1 (output enabled). The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TXD2 output low with the timing shown in Figure 15.32. If the U2RB register is read while outputting a parity error signal, the PER bit in the U2RB register is set to 0 (no parity error) and at the same time the TXD2 output is returned high.

When transmitting, a transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the UXD2 pin in a transmission-finished interrupt routine.

Figure 15.34 shows the output timing of the parity error signal



This timing diagram applies to the case where the direct format is implemented.

ST: Start bit
P: Even Parity
SP: Stop bit

NOTE:
1: The output of MCU is in the high-impedance state (pulled up externally).

Figure 15.34 Parity Error Signal Output Timing

15.1.6.2 Format

When direct format, set the PRYE bit in the U2MR register to 1, the PRY bit to 1, the UFORM bit in the U2C0 register to 0 and the U2LCH bit in the U2C1 register to 0. When data are transmitted, data set in the U2TB register are transmitted with the even-numbered parity, starting from D0. When data are received, received data are stored in the U2RB register, starting from D0. The even-numbered parity determines whether a parity error occurs.

When inverse format, set the PRYE bit to 1, the PRY bit to 0, the UFORM bit to 1 and the U2LCH bit to 1. When data are transmitted, values set in the U2TB register are logically inversed and are transmitted with the odd-numbered parity, starting from D7. When data are received, received data are logically inversed to be stored in the U2RB register, starting from D7. The odd-numbered parity determines whether a parity error occurs.

Figure 15.35 shows the SIM Interface Format.

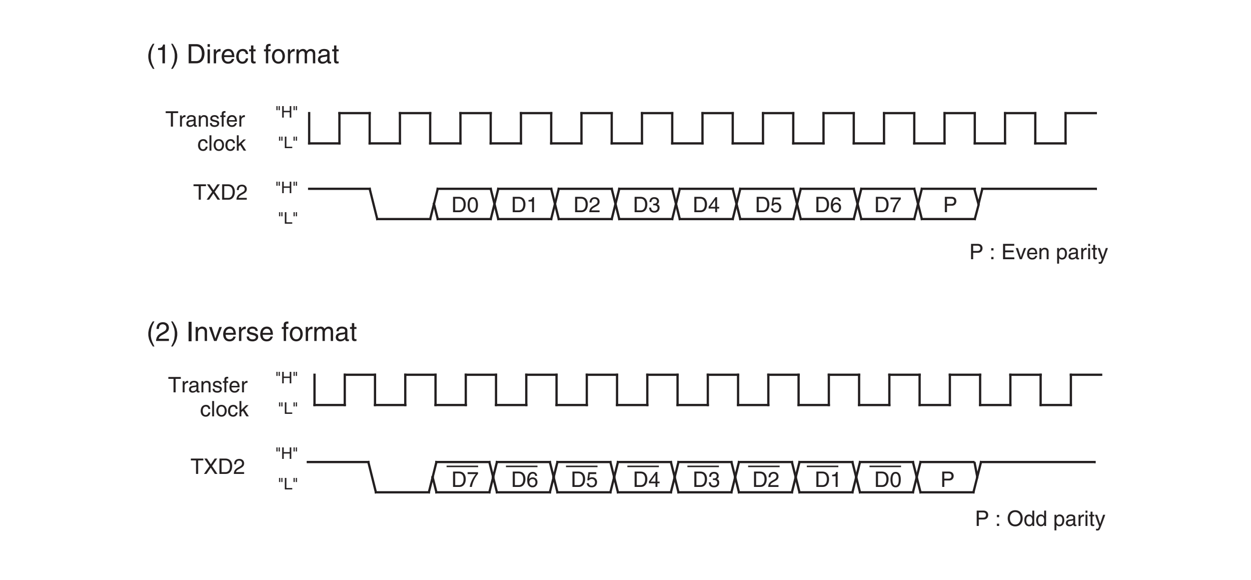


Figure 15.35 SIM Interface Format

15.2 SI/Oi (i = 3 to 6) [1]

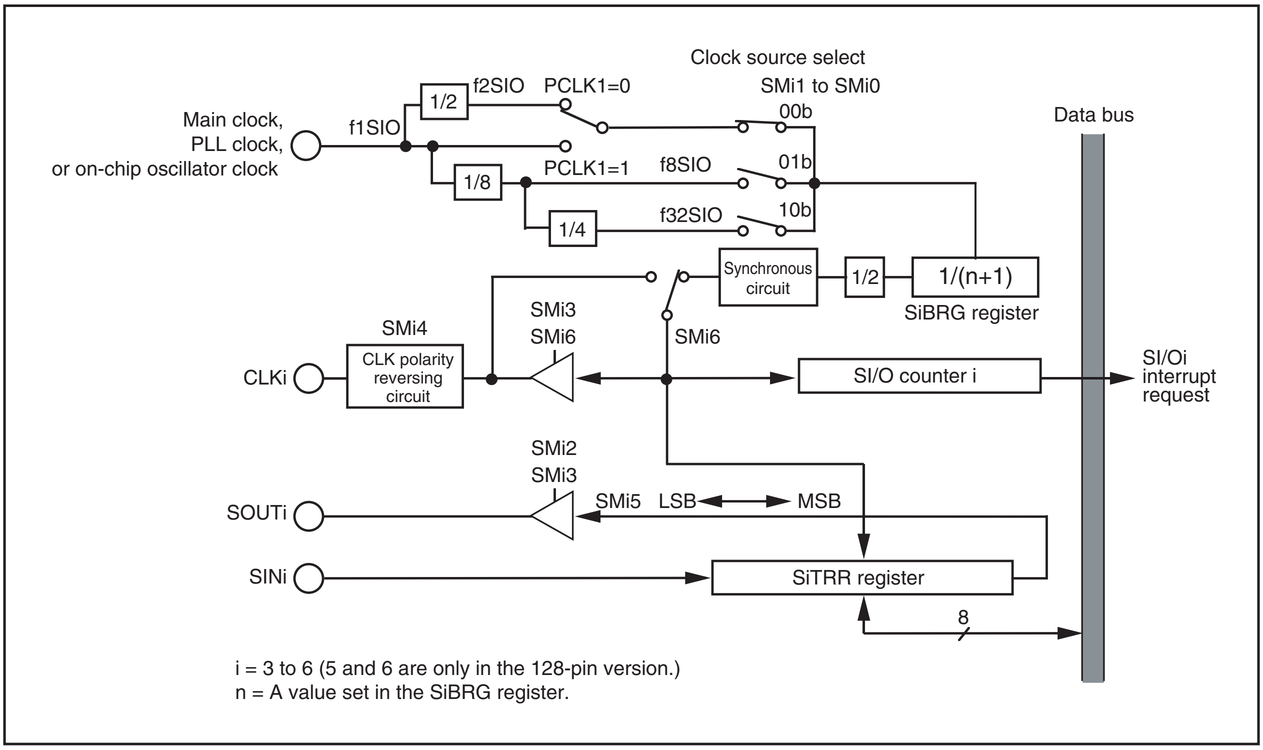
SI/Oi is exclusive clock-synchronous serial I/Os.

Figure 15.36 shows the SI/Oi Block Diagram, and Figures 15.37 and 15.38 show the SI/Oi-related registers.

Table 15.19 lists the SI/Oi Specifications.

NOTE:

1. 100-pin version supports SI/O3 and SI/O4.
   128-pin version supports SI/O3, SI/O4, SI/O5 and SI/O6.

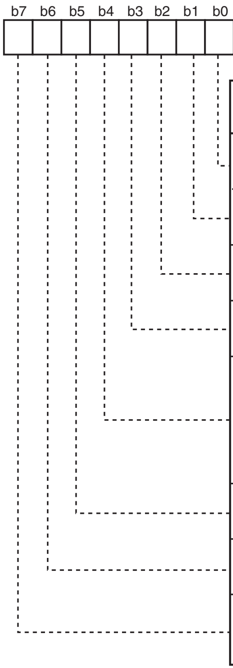


i = 3 to 6 (5 and 6 are only in the 128-pin version.)
n = A value set in the SiBRG register.

Figure 15.36 SI/Oi Block Diagram

SI/Oi Control Register (i = 3 to 6) (1)

Symbol	Address	After Reset
S3C	01E2h	01000000b
S4C	01E6h	01000000b
S5C (6)	01EAh	01000000b
S6C (6)	01D8h	01000000b



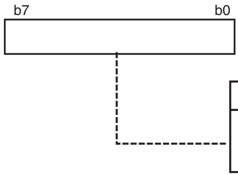
Bit Symbol	Bit Name	Description	RW
SMi0	Internal synchronous clock select bit (7)	b1 b0 0 0 : f1SIO or f2SIO is selected (8) 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Do not set a value	RW
SMi1			RW
SMi2	SOUTi output disable bit (4)	0 : SOUTi output 1 : SOUTi output disabled (high-impedance)	RW
SMi3	SI/Oi port select bit (5)	0 : Input/output port 1 : SOUTi output, CLKi function	RW
SMi4	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	RW
SMi5	Transfer direction select bit	0 : LSB first 1 : MSB first	RW
SMi6	Synchronous clock select bit	0 : External clock (2) 1 : Internal clock (3)	RW
SMi7	SOUTi initial value set bit	Effective when the SMi3 bit = 0 0 : "L" output 1 : "H" output	RW

NOTES:

1. Make sure this register is written to by the next instruction after setting the PRC2 bit in the PRCR register to 1 (write enabled).
2. Set the SMi3 bit to 1 (SOUTi output, CLKi function) and the corresponding port direction bit to 0 (input mode).
3. Set the SMi3 bit to 1 (SOUTi output, CLKi function) .
4. When the SM32, SM52 or SM62 bit = 1 (SOUT3, SOUT5, SOUT6 output disabled), the corresponding pin is placed in the high-impedance state regardless of which functions of those pins are being used.
   SI/O4 is effective only when the SM43 bit = 1 (SOUT4 output, CLK4 function).
5. When using SI/O4, set the SM43 bit to 1 (SOUT4 output, CLK4 function) and the corresponding port direction bit for SOUT4 pin to 0 (input mode).
6. Registers S5C and S6C are only in the 128-pin version. When using registers S5C and S6C, set these registers after setting the PU37 bit in the PUR3 register to 1 (Pins P11 to P14 are usable).
7. When changing bits SMi1 to SMi0, set the SiBRG register.
8. Selected by the PCLK1 bit in the PCLKR register.

SI/Oi Bit Rate Register (i = 3 to 6) (1) (2) (4)

Symbol	Address	After Reset
S3BRG	01E3h	Undefined
S4BRG	01E7h	Undefined
S5BRG (3)	01EBh	Undefined
S6BRG (3)	01D9h	Undefined



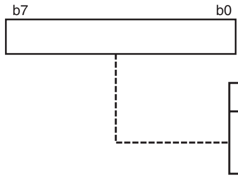
Description	Setting Range	RW
Assuming that set value = n, SiBRG divides the count source by n + 1	00h to FFh	WO

NOTES:

1. Write to this register while serial interface is neither transmitting nor receiving.
2. Use the MOV instruction to write to this register.
3. Registers S5BRG and S6BRG are only in the 128-pin version.
4. Write to this register after setting bits SMi1 to SMi0 in the SiC register.

SI/Oi Transmit/Receive Register (i = 3 to 6) (1) (2)

Symbol	Address	After Reset
S3TRR	01E0h	Undefined
S4TRR	01E4h	Undefined
S5TRR (3)	01E8h	Undefined
S6TRR (3)	01D6h	Undefined



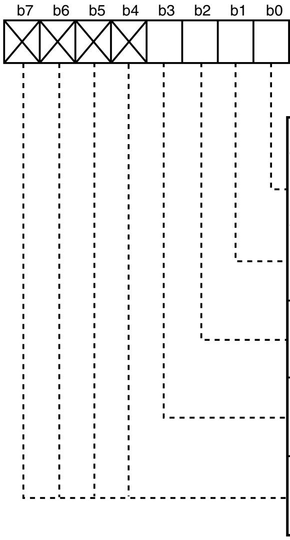
Description	RW
Transmission/reception starts by writing transmit data to this register. After transmission/reception finishes, reception data can be read by reading this register.	RW

NOTES:

1. Write to this register while serial I/O is neither transmitting nor receiving.
2. To receive data, set the corresponding port direction bit for SINi to 0 (input mode).
3. Registers S5TRR and S6TRR are only in the 128-pin version.

Figure 15.37 Registers SiC, SiBRG, and SiTRR

SI/O3, 4, 5, 6 Transmit/Receive Register (1) (2)



Symbol	Address	After Reset
S3456TRR	01DAh	XXXX0000b

Bit Symbol	Bit Name	Function	RW
S3TRF	SI/O3 transmit/receive complete flag	0 : During transmission/reception 1 : Transmission/reception completed	RW
S4TRF	SI/O4 transmit/receive complete flag	0 : During transmission/reception 1 : Transmission/reception completed	RW
S5TRF	SI/O5 transmit/receive complete flag	0 : During transmission/reception 1 : Transmission/reception completed	RW
S6TRF	SI/O6 transmit/receive complete flag	0 : During transmission/reception 1 : Transmission/reception completed	RW
- (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTES:
1. Bits S3TRF to S6TRF can only be reset by writing to 0. (Bits S5TRF and S6TRF are only in the 128-pin version.)
2. When setting bits S3TRF to S6TRF to 0, use the MOV instruction to write to these bits after setting the bit that wants to set 0 to 0 and setting other bits to 1.

Figure 15.38 S3456TRR Register

Table 15.19 SI/Oi Specifications

Item	Specification
Transfer data format	Transfer data length: 8 bits
Transfer clock	• SMi6 bit in SiC register = 1 (internal clock) : fj/(2(n+1)) fj = f1SIO, f8SIO, f32SIO. n = Setting value of SiBRG register 00h to FFh • SMi6 bit = 0 (external clock) : Input from CLKi pin (1)
Transmit/receive start condition	Before transmission/reception can start, meet the following requirements Write transmit data to the SiTRR register (2) (3)
Interrupt request generation timing	• When SMi4 bit in SiC register = 0 The rising edge of the last transfer clock pulse (4) • When SMi4 bit = 1 The falling edge of the last transfer clock pulse (4)
CLKi pin function	I/O port, transfer clock input, transfer clock output
SOUTi pin function	I/O port, transmit data output, high-impedance
SINi pin function	I/O port, receive data input
Select function	• LSB first or MSB first selection Whether to start transmitting or receiving data begins with bit 0 or begins with bit 7 can be selected • Function for setting an SOUTi initial value set function When the SMi6 bit in the SiC register = 0 (external clock), the SOUTi pin output level while not transmitting can be selected. • CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected.

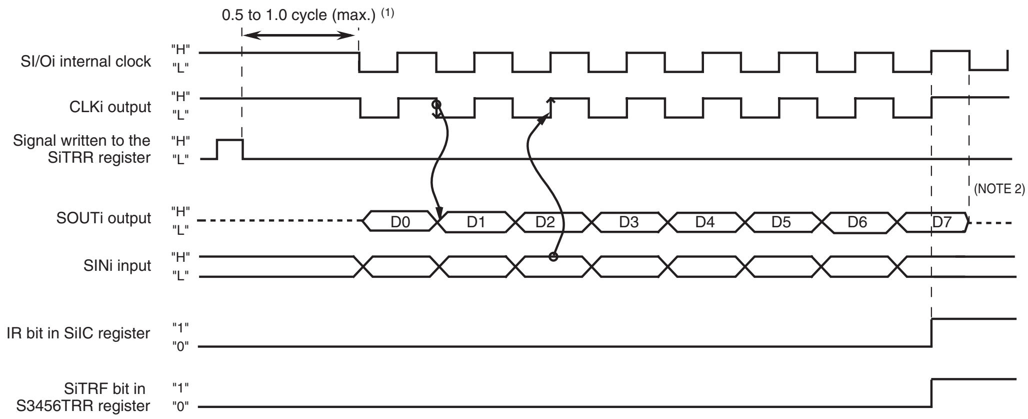
i = 3 to 6 (5 and 6 are only in the 128-pin version.)

NOTES:

1. To set the SMi6 bit in the SiC register to 0 (external clock), follow the procedure described below.
   - If the SMi4 bit in the SiC register = 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SMi7 bit in the SiC register.
   - If the SMi4 bit = 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMi7 bit.
   - Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock after supplying eight pulses. If the SMi6 bit = 1 (internal clock), the transfer clock automatically stops.
2. Unlike UART0 to UART2, SI/Oi is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.
3. When the SMi6 bit = 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.
4. When the SMi6 bit = 1 (internal clock), the transfer clock stops in the high state if the SMi4 bit = 0, or stops in the low state if the SMi4 bit = 1.

15.2.1 SI/Oi Operation Timing

Figure 15.39 shows the SI/Oi Operation Timing.



i = 3 to 6 (5 and 6 are only in the 128-pin version.)
* This diagram applies to the case where the bits in the SiC register are set as follows:

- SMi2 = 0 (SOUTi output)
- SMi3 = 1 (SOUTi output, CLKi function)
- SMi4 = 0 (transmit data output at the falling edge and receive data input at the rising edge of the transfer clock)
- SMi5 = 0 (LSB first)
- SMi6 = 1 (internal clock)

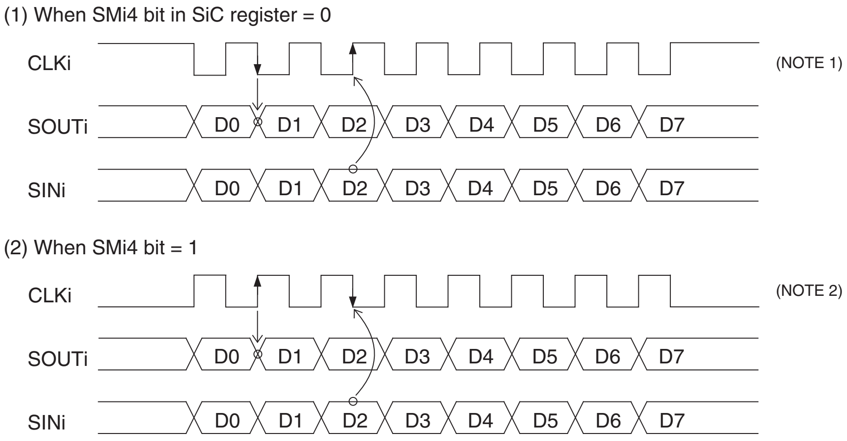
NOTES:
1. If the SMi6 bit = 1 (internal clock), the serial interface starts transmitting or receiving data a maximum of 0.5 to 1.0 transfer clock cycles after writing to the SiTRR register.
2. When the SMi6 bit = 1 (internal clock), the SOUTi pin is placed in the high-impedance state after the transfer finishes.

Figure 15.39 SI/Oi Operation Timing

15.2.2 CLK Polarity Selection

The SMi4 bit in the SiC register allows selection of the polarity of the transfer clock.

Figure 15.40 shows the Polarity of Transfer Clock.



i = 3 to 6 (5 and 6 are only in the 128-pin version.)
*This diagram applies to the case where the bits in the SiC register are set as follows:

- SMi5 = 0 (LSB first)
- SMi6 = 1 (internal clock)

NOTES:
1. When the SMi6 bit = 1 (internal clock), a high level is output from the CLKi pin if not transferring data.
2. When the SMi6 bit = 1 (internal clock), a low level is output from the CLKi pin if not transferring data.

Figure 15.40 Polarity of Transfer Clock

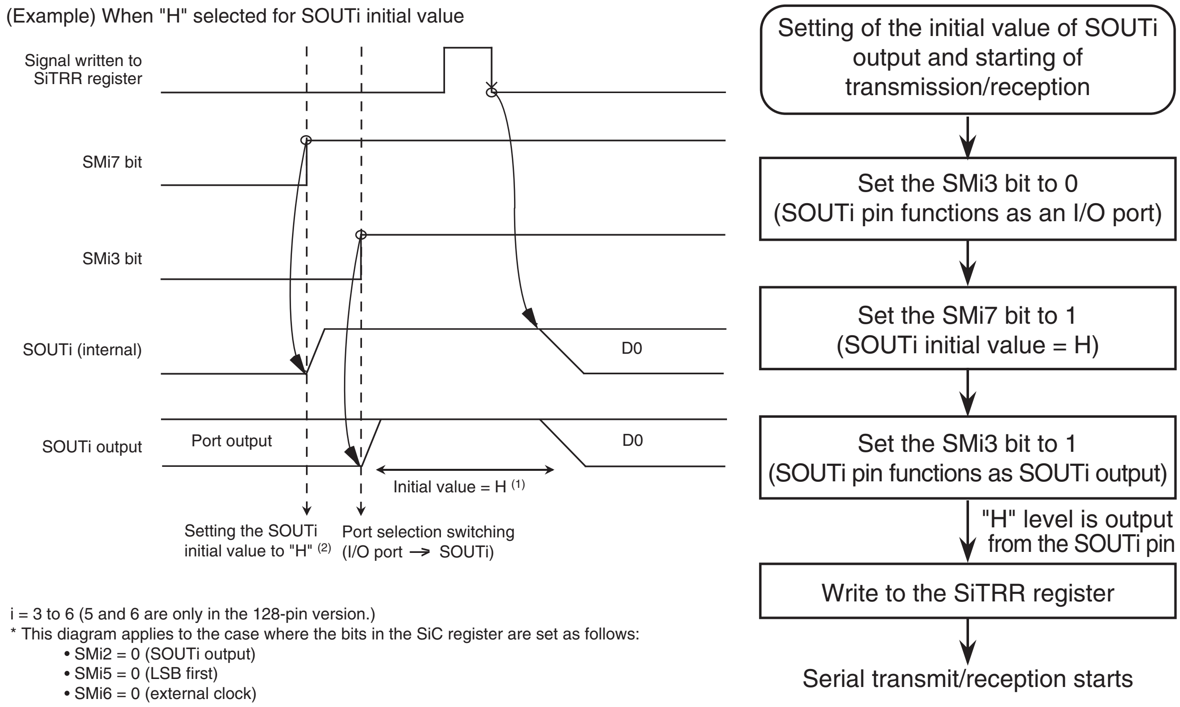
15.2.3 Functions for Setting SOUTi Initial Value

If the SMi6 bit in the SiC register = 0 (external clock), the SOUTi pin output can be fixed high or low when not transferring [1]. However, the last bit value of the former data is retained between data and data when transmitting the continuous data.

Figure 15.41 shows the timing chart for setting an SOUTi initial value and how to set it.

NOTE:

1. When CAN0 function is selected, P7_4, P7_5 and P8_0 can be used as input/output pins for SI/O4. When CAN0 function is not selected, P9_5, P9_6 and P9_7 can be used as input/output pis for SI/O4.


(Example) When "H" selected for SOUTi initial value
Signal written to SiTRR register
SMi7 bit
SMi3 bit
SOUTi (internal)
D0
SOUTi output
Port output
D0
Initial value = H (1)
Setting the SOUTi initial value to "H" (2)
Port selection switching (I/O port → SOUTi)
Setting of the initial value of SOUTi output and starting of transmission/reception
Set the SMi3 bit to 0 (SOUTi pin functions as an I/O port)
Set the SMi7 bit to 1 (SOUTi initial value = H)
Set the SMi3 bit to 1 (SOUTi pin functions as SOUTi output)
"H" level is output from the SOUTi pin
Write to the SiTRR register
Serial transmit/reception starts

i = 3 to 6 (5 and 6 are only in the 128-pin version.)

* This diagram applies to the case where the bits in the SiC register are set as follows:

- SMi2 = 0 (SOUTi output)
- SMi5 = 0 (LSB first)
- SMi6 = 0 (external clock)

NOTES:

1. If the SMi6 bit = 1 (internal clock) or if the SMi2 bit = 1 (SOUTi output disabled), this output goes to the high-impedance state.
2. SOUTi can only be initialized when input on the CLKi pin is in the high state if the SMi4 bit in the SiC register = 0 (transmit data output at the falling edge of the transfer clock) or in the low state if the SMi4 bit = 1 (transmit data output at the rising edge of the transfer clock).

Figure 15.41 SOUTi's Initial Value Setting

16. A/D Converter

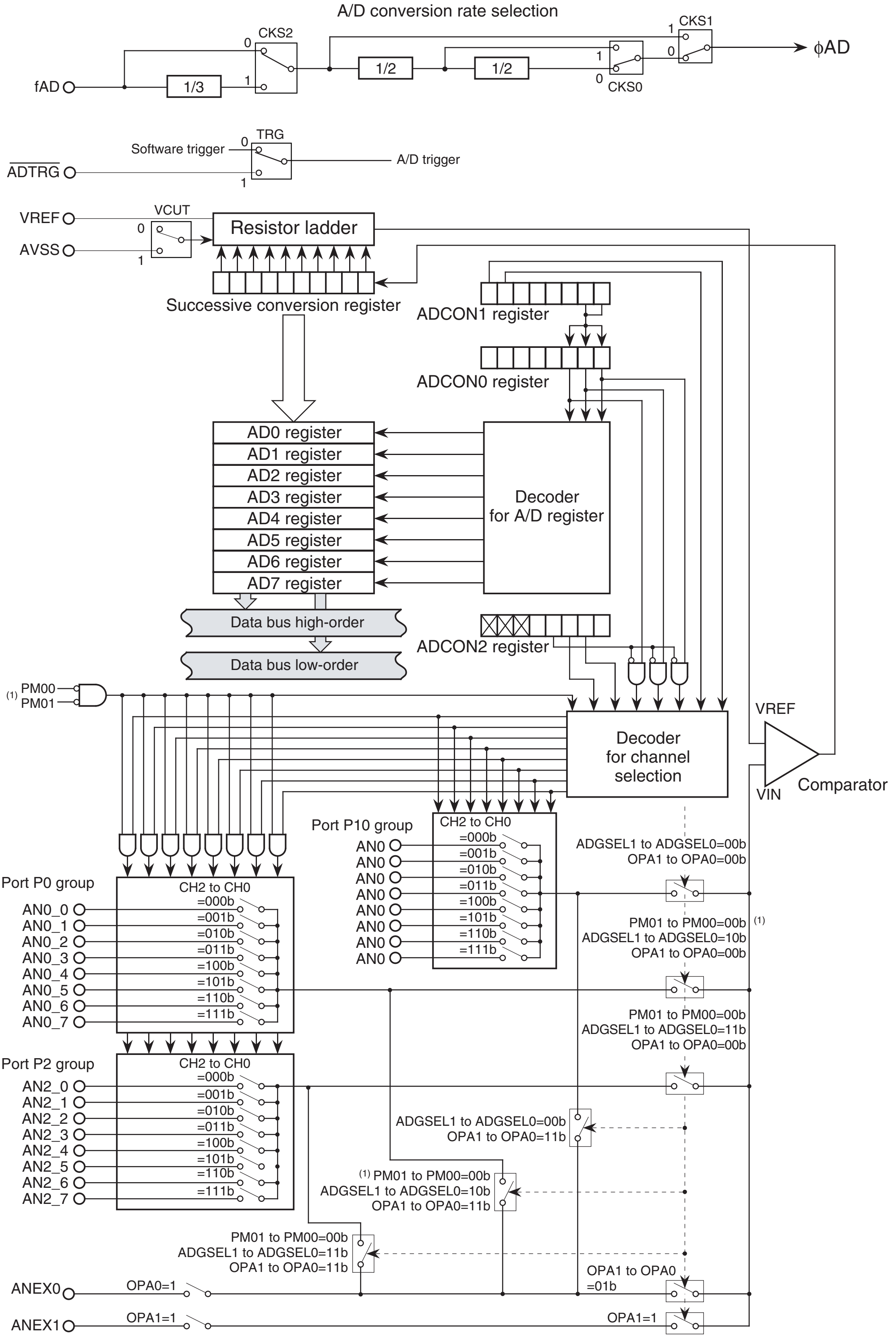
The MCU contains one A/D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P10_0 to P10_7, P9_5, P9_6, P0_0 to P0_7, and P2_0 to P2_7. Similarly, $\overline{\text{ADTRG}}$ input shares the pin with P9_7. Therefore, when using these inputs, make sure the corresponding port direction bits are set to 0 (input mode).
When not using the A/D converter, set the VCUT bit to 0 (VREF unconnected), so that no current will flow from the VREF pin into the resistor ladder, helping to reduce the power consumption of the chip.
The A/D conversion result is stored in the bits in the ADi register for pins ANi, AN0_i, and AN2_i (i = 0 to 7).
Table 16.1 shows the A/D Converter Performance. Figure 16.1 shows the A/D Converter Block Diagram, and Figures 16.2 and 16.3 show the A/D converter-related registers.

Table 16.1 A/D Converter Performance

Item	Performance
Method of A/D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage [(1)]	0 V to AVCC (VCC)
Operating clock φAD [(2)]	fAD, divide-by-2 of fAD, divide-by-3 of fAD, divide-by-4 of fAD, divide-by-6 of fAD, divide-by-12 of fAD
Resolution	8 bits or 10 bits (selectable)
Integral nonlinearity error	When AVCC = VREF = 5 V • With 8-bit resolution: ±2 LSB • With 10-bit resolution AN0 to AN7 input, AN0_0 to AN0_7 input and AN2_0 to AN2_7 input: ±3 LSB ANEX0 and ANEX1 input (including mode in which external operation amp is selected): ±7 LSB When AVCC = VREF = 3.3 V • With 8-bit resolution: ±2 LSB • With 10-bit resolution AN0 to AN7 input, AN0_0 to AN0_7 input and AN2_0 to AN2_7 input: ±5 LSB ANEX0 and ANEX1 input (including mode in which external operation amp is selected): ±7 LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN0 to AN7) + 2 pins (ANEX0 and ANEX1) + 8 pins (AN0_0 to AN0_7) + 8 pins (AN2_0 to AN2_7)
A/D conversion start condition	• Software trigger The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • External trigger (retriggerable) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts)
Conversion speed per pin	• Without sample and hold 8-bit resolution: 49 φAD cycles, 10-bit resolution: 59 φAD cycles • With sample and hold 8-bit resolution: 28 φAD cycles, 10-bit resolution: 33 φAD cycles

NOTES:

1. Does not depend on use of sample and hold.
2. φAD frequency must be 10 MHz or less.
   When sample and hold is disabled, φAD frequency must be 250 kHz or more.
   When sample and hold is enabled, φAD frequency must be 1 MHz or more.

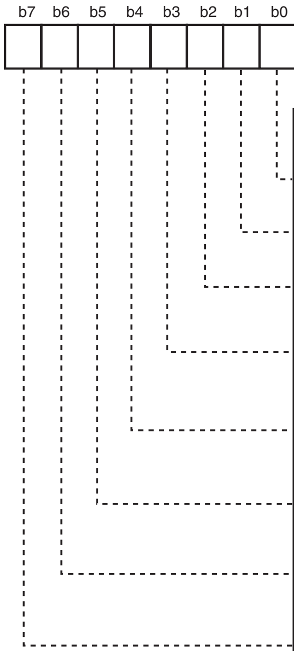


NOTE:

1. Port P0 group (AN0_0 to AN0_7) can be used as analog input pins even when bits PM01 to PM00 are set to 01b (memory expansion mode) and bits PM05 to PM04 are set to 11b (multiplex bus allocated to the entire CS space).

Figure 16.1 A/D Converter Block Diagram

A/D Control Register 0 (1)

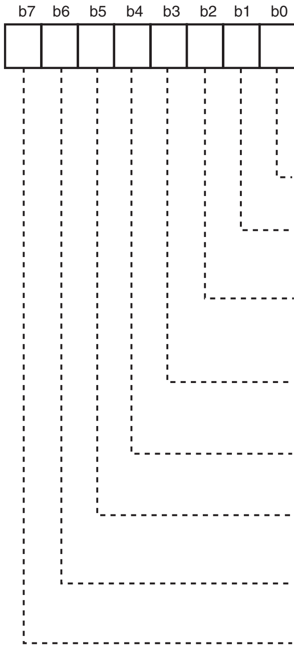


Symbol	Address	After Reset
ADCON0	03D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits	Function varies depending on operating mode	RW
CH1			RW
CH2			RW
MD0	A/D operating mode select bits 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{\text{ADTRG}}$ trigger	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	Refer to **NOTE 2 for ADCON2 Register**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 (1)



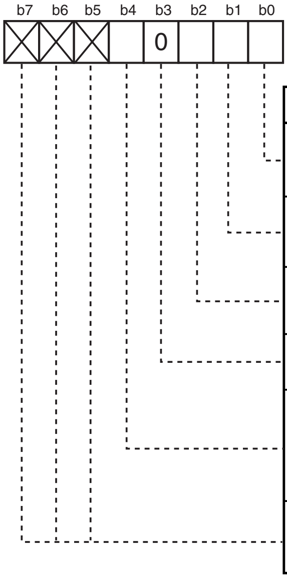
Symbol	Address	After Reset
ADCON1	03D7h	00h

Bit symbol	Bit name	Function	RW
SCAN0	A/D sweep pin select bits	Function varies depending on operating mode	RW
SCAN1			RW
MD2	A/D operating mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	Refer to **NOTE 2 for ADCON2 Register**	RW
VCUT	VREF connect bit (2)	0 : VREF not connected 1 : VREF connected	RW
OPA0	External op-amp connection mode bits	Function varies depending on operating mode	RW
OPA1			RW

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (VREF unconnected) to 1 (VREF connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.2 Registers ADCON0 and ADCON1

A/D Control Register 2 [1]



Symbol	Address	After Reset
ADCON2	03D4h	00h

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
ADGSEL0	A/D input group select bits	b2 b1 0 0 : Port P10 group is selected 0 1 : Do not set a value 1 0 : Port P0 group is selected 1 1 : Port P2 group is selected	RW
ADGSEL1			RW
– (b3)	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2 [2]	0 : Selects fAD, divide-by-2 of fAD, or divide-by-4 of fAD. 1 : Selects divide-by-3 of fAD, divide-by-6 of fAD, or divide-by-12 of fAD.	RW
– (b7-b5)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–

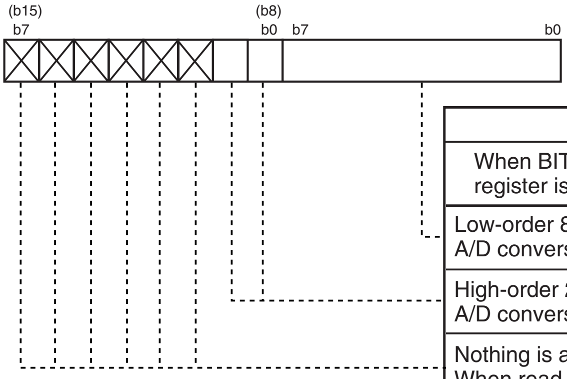
NOTES:

1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.
2. The ϕAD frequency must be 10 MHz or less. The selected ϕAD frequency is determined by a combination of the CKS0 bit in the ADCON0 register, the CKS1 bit in the ADCON1 register, and the CKS2 bit in the ADCON2 register.

CKS2	CKS1	CKS0	ϕAD
0	0	0	Divide-by-4 of fAD
0	0	1	Divide-by-2 of fAD
0	1	0	fAD
0	1	1	
1	0	0	Divide-by-12 of fAD
1	0	1	Divide-by-6 of fAD
1	1	0	Divide-by-3 of fAD
1	1	1	

A/D Register i (i = 0 to 7)

Symbol	Address	After Reset
AD0	03C1h to 03C0h	Undefined
AD1	03C3h to 03C2h	Undefined
AD2	03C5h to 03C4h	Undefined
AD3	03C7h to 03C6h	Undefined
AD4	03C9h to 03C8h	Undefined
AD5	03CBh to 03CAh	Undefined
AD6	03CDh to 03CCh	Undefined
AD7	03CFh to 03CEh	Undefined



Function		RW
When BITS bit in ADCON1 register is 1 (10-bit mode)	When BITS bit is 0 (8-bit mode)	
Low-order 8 bits of A/D conversion result	A/D conversion result	RO
High-order 2 bits of A/D conversion result	When read, the content is undefined.	RO
Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–

Figure 16.3 Registers ADCON2, and AD0 to AD7

16.1 Mode Description

16.1.1 One-shot Mode

In one-shot mode, analog voltage applied to a selected pin is converted to a digital code once.

Table 16.2 lists the One-shot Mode Specifications. Figure 16.4 shows Registers ADCON0 and ADCON1 in One-shot Mode.

Table 16.2 One-shot Mode Specifications

Item	Specification
Function	Bits CH2 to CH0 in the ADCON0 register, bits ADGSEL1 to ADGSEL0 in the ADCON2 register, and bits OPA1 to OPA0 in the ADCON1 register select a pin Analog voltage applied to the pin is converted to a digital code once.
A/D conversion start condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts)
A/D conversion stop condition	• Completion of A/D conversion (If a software trigger is selected, the ADST bit is set to 0 (A/D conversion halted).) • Set the ADST bit to 0
Interrupt request generation timing	Completion of A/D conversion
Analog input pin	Select one pin from AN0 to AN7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0 to ANEX1
Reading of result of A/D converter	Read one of registers AD0 to AD7 that corresponds to the selected pin

A/D Control Register 0 [1]

b7	b6	b5	b4	b3	b2	b1	b0
			0	0			

Symbol	Address	After Reset
ADCON0	03D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected [2] [3]	RW
CH1			RW
CH2			RW
MD0	A/D operating mode select bits 0	b4 b3 0 0 : One-shot mode [3]	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{\mathrm{ADTRG}}$ trigger	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	Refer to **NOTE 2 for ADCON2 Register**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in same way as AN0 to AN7. Use bits ADGSEL1 to ADGSEL0 in the ADCON2 register to select the desired pin.
3. After rewriting bits MD1 to MD0, set bits CH2 to CH0 over again using another instruction.

A/D Control Register 1 [1]

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol	Address	After Reset
ADCON1	03D7h	00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bits	Invalid in one-shot mode	RW
SCAN1			RW
MD2	A/D operating mode select bit 1	Set to 0 when one-shot mode is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	Refer to **NOTE 2 for ADCON2 Register**	RW
VCUT	VREF connect bit [2]	1 : VREF connected	RW
OPA0	External op-amp connection mode bits	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (VREF unconnected) to 1 (VREF connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.4 Registers ADCON0 and ADCON1 in One-shot Mode

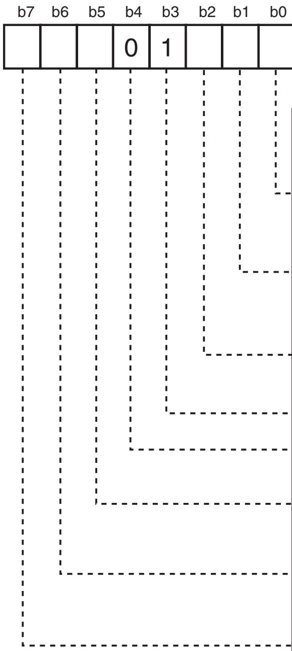
16.1.2 Repeat Mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. Table 16.3 lists the Repeat Mode Specifications. Figure 16.5 shows Registers ADCON0 and ADCON1 in Repeat Mode.

Table 16.3 Repeat Mode Specifications

Item	Specification
Function	Bits CH2 to CH0 in the ADCON0 register, bits ADGSEL1 to ADGSEL0 in the ADCON2 register, and bits OPA1 to OPA0 in the ADCON1 register select a pin. Analog voltage applied to this pin is repeatedly converted to a digital code.
A/D conversion start condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts)
A/D conversion stop condition	Set the ADST bit to 0 (A/D conversion halted)
Interrupt request generation timing	None generated
Analog input pin	Select one pin from AN0 to AN7, AN0_0 to AN0_7, AN2_0 to AN2_7, ANEX0 to ANEX1
Reading of result of A/D converter	Read one of registers AD0 to AD7 that corresponds to the selected pin

A/D Control Register 0 (1)

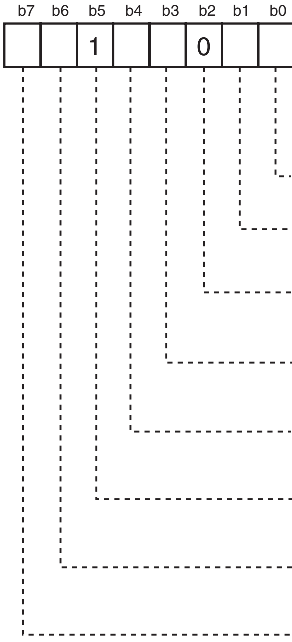


Symbol	Address	After Reset
ADCON0	03D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected (2) (3)	RW
CH1			RW
CH2			RW
MD0	A/D operating mode select bits 0	b4 b3 0 1 : Repeat mode (3)	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	Refer to **NOTE 2 for ADCON2 Register**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in same way as AN0 to AN7. Use bits ADGSEL1 to ADGSEL0 in the ADCON2 register to select the desired pin.
3. After rewriting bits MD1 to MD0, set bits CH2 to CH0 over again using another instruction.

A/D Control Register 1 (1)



Symbol	Address	After Reset
ADCON1	03D7h	00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bits	Invalid in repeat mode	RW
SCAN1			RW
MD2	A/D operating mode select bit 1	Set to 0 when repeat mode is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	Refer to **NOTE 2 for ADCON2 Register**	RW
VCUT	VREF connect bit (2)	1 : VREF connected	RW
OPA0	External op-amp connection mode bits	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A/D converted 1 0 : ANEX1 input is A/D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (VREF unconnected) to 1 (VREF connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.5 Registers ADCON0 and ADCON1 in Repeat Mode

16.1.3 Single Sweep Mode

In single sweep mode, analog voltage that is applied to selected pins is converted one-by-one to a digital code. Table 16.4 lists the Single Sweep Mode Specifications. Figure 16.6 shows Registers ADCON0 and ADCON1 in Single Sweep Mode.

Table 16.4 Single Sweep Mode Specifications

Item	Specification
Function	Bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 to ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to this pins is converted one-by-one to a digital code.
A/D conversion start condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts)
A/D conversion stop condition	• Completion of A/D conversion (If a software trigger is selected, the ADST bit is set to 0 (A/D conversion halted).) • Set the ADST bit to 0
Interrupt request generation timing	Completion of A/D conversion
Analog input pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) (1)
Reading of result of A/D converter	Read one of registers AD0 to AD7 that corresponds to the selected pin

NOTE:

1. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in the same way as AN0 to AN7.

A/D Control Register 0 [(1)]

b7	b6	b5	b4	b3	b2	b1	b0
			1	0			

Symbol	Address	After Reset
ADCON0	03D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits	Invalid in single sweep mode	RW
CH1			RW
CH2			RW
MD0	A/D operating mode select bits 0	b4 b3 1 0 : Single sweep mode	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{\text{ADTRG}}$ trigger	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	Refer to **NOTE 2 for ADCON2 Register**	RW

NOTE:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 [(1)]

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol	Address	After Reset
ADCON1	03D7h	00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bits	When single sweep mode is selected b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins) [(2)]	RW
SCAN1			RW
MD2	A/D operating mode select bit 1	Set to 0 when single sweep mode is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	Refer to **NOTE 2 for ADCON2 Register**	RW
VCUT	VREF connect bit [(3)]	1 : VREF connected	RW
OPA0	External op-amp connection mode bits	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Do not set a value 1 0 : Do not set a value 1 1 : External op-amp connection mode	RW
OPA1			RW

NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in same way as AN0 to AN7. Use bits ADGSEL1 to ADGSEL0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (VREF unconnected) to 1 (VREF connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.6 Registers ADCON0 and ADCON1 in Single Sweep Mode

16.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage applied to selected pins is repeatedly converted to a digital code. Table 16.5 lists the Repeat Sweep Mode 0 Specifications. Figure 16.7 shows Registers ADCON0 and ADCON1 in Repeat Sweep Mode 0.

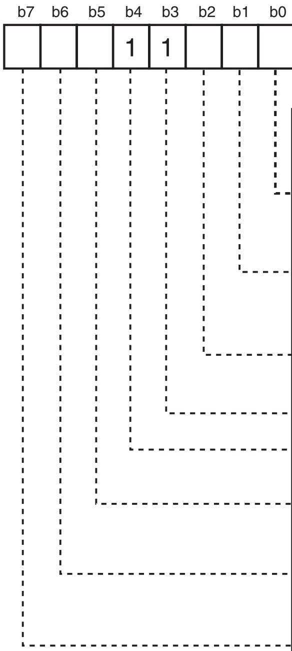
Table 16.5 Repeat Sweep Mode 0 Specifications

Item	Specification
Function	Bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 to ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the pins is repeatedly converted to a digital code.
A/D conversion start condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts)
A/D conversion stop condition	Set the ADST bit to 0 (A/D conversion halted)
Interrupt request generation timing	None generated
Analog input pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), AN0 to AN7 (8 pins) (1)
Reading of result of A/D converter	Read one of registers AD0 to AD7 that corresponds to the selected pin

NOTE:

1. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in the same way as AN0 to AN7.

A/D Control Register 0 [1]



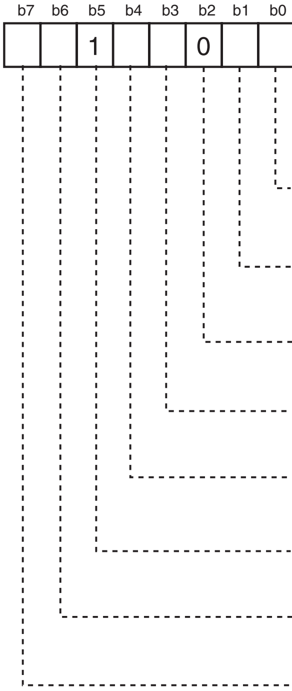
Symbol	Address	After Reset
ADCON0	03D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits	Invalid in repeat sweep mode 0	RW
CH1			RW
CH2			RW
MD0	A/D operating mode select bits 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{\text{ADTRG}}$ trigger	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	Refer to **NOTE 2 for ADCON2 Register**	RW

NOTE:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 [1]



Symbol	Address	After reset
ADCON1	03D7h	00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bits	When repeat sweep mode 0 is selected b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins) [2]	RW
SCAN1			RW
MD2	A/D operating mode select bit 1	Set to 0 when repeat sweep mode 0 is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	Refer to **NOTE 2 for ADCON2 Register**	RW
VCUT	VREF connect bit [3]	1 : VREF connected	RW
OPA0	External op-amp connection mode bits	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Do not set a value 1 0 : Do not set a value 1 1 : External op-amp connection mode	RW
OPA1			RW

NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in same way as AN0 to AN7. Use bits ADGSEL1 to ADGSEL0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (VREF unconnected) to 1 (VREF connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.7 Registers ADCON0 and ADCON1 in Repeat Sweep Mode 0

16.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage selectively applied to all pins is repeatedly converted to a digital code. Table 16.6 lists the Repeat Sweep Mode 1 Specifications. Figure 16.8 shows Registers ADCON0 and ADCON1 in Repeat Sweep Mode 1.

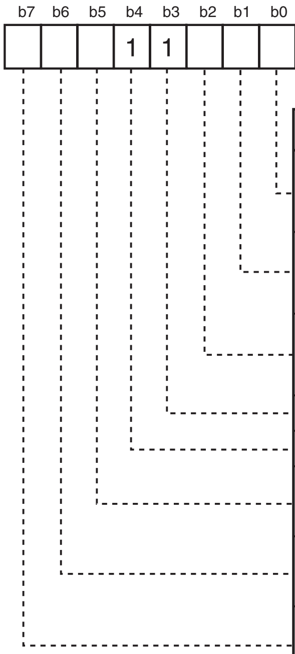
Table 16.6 Repeat Sweep Mode 1 Specifications

Item	Specification
Function	The input voltages on all pins selected by bits ADGSEL1 to ADGSEL0 in the ADCON2 register are A/D converted repeatedly, with priority given to pins selected by bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 to ADGSEL0. Example : If AN0 selected, input voltages are A/D converted in order of AN0 → AN1 → AN0 → AN2 → AN0 → AN3, and so on.
A/D conversion start condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) The ADST bit in the ADCON0 register is set to 1 (A/D conversion starts) • When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger) Input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion starts)
A/D conversion stop condition	Set the ADST bit to 0 (A/D conversion halted)
Interrupt request generation timing	None generated
Analog input pins to be given priority when A/D converted	Select from AN0 (1 pin), AN0 to AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins) [1]
Reading of result of A/D converter	Read one of registers AD0 to AD7 that corresponds to the selected pin

NOTE:

1. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in the same way as AN0 to AN7.

A/D Control Register 0 (1)



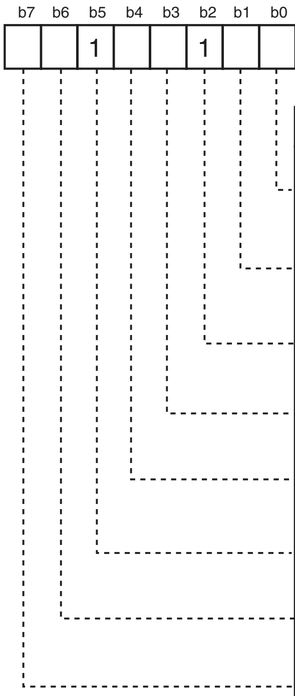
Symbol	Address	After Reset
ADCON0	03D6h	00000XXXb

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits	Invalid in repeat sweep mode 1	RW
CH1			RW
CH2			RW
MD0	A/D operating mode select bits 0	b4 b3 1 1 : Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0	Refer to **NOTE 2 for ADCON2 Register**	RW

NOTE:

1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 (1)



Symbol	Address	After Reset
ADCON1	03D7h	00h

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bits	When repeat sweep mode 1 is selected b1 b0 0 0 : AN0 (1 pin) 0 1 : AN0, AN1 (2 pins) 1 0 : AN0 to AN2 (3 pins) 1 1 : AN0 to AN3 (4 pins) (2)	RW
SCAN1			RW
MD2	A/D operating mode select bit 1	Set to 1 when repeat sweep mode 1 is selected	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	Refer to **NOTE 2 for ADCON2 Register**	RW
VCUT	VREF connect bit (3)	1 : VREF connected	RW
OPA0	External op-amp connection mode bits	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : Do not set a value 1 0 : Do not set a value 1 1 : External op-amp connection mode	RW
OPA1			RW

NOTES:

1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN0_0 to AN0_7, and AN2_0 to AN2_7 can be used in same way as AN0 to AN7. Use bits ADGSEL1 to ADGSEL0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (VREF unconnected) to 1 (VREF connected), wait for 1 µs or more before starting A/D conversion.

Figure 16.8 Registers ADCON0 and ADCON1 in Repeat Sweep Mode 1

16.2 Function

16.2.1 Resolution Select Function

The desired resolution can be selected using the BITS bit in the ADCON1 register. If the BITS bit is set to 1 (10-bit conversion accuracy), the A/D conversion result is stored in the bits 0 to 9 in the ADi register (i = 0 to 7). If the BITS bit is set to 0 (8-bit conversion accuracy), the A/D conversion result is stored in the bits 0 to 7 in the ADi register.

16.2.2 Sample and Hold

If the SMP bit in the ADCON2 register is set to 1 (with sample and hold), the conversion speed per pin is increased to 28 ϕAD cycles for 8-bit resolution or 33 ϕAD cycles for 10-bit resolution. Sample and hold is effective in all operating modes. Select whether or not to use the sample and hold function before starting A/D conversion.

16.2.3 Extended Analog Input Pins

In one-shot and repeat modes, pins ANEX0 and ANEX1 can be used as analog input pins. Use bits OPA1 to OPA0 in the ADCON1 register to select whether or not use ANEX0 and ANEX1.

The A/D conversion results of ANEX0 and ANEX1 inputs are stored in registers AD0 and AD1, respectively.

16.2.4 External Operation Amplifier (Op-Amp) Connection Mode

Multiple analog inputs can be amplified using a single external op-amp via pins ANEX0 and ANEX1.

Set bits OPA1 to OPA0 in the ADCON1 register to 11b (external op-amp connection mode). The inputs from ANi (i = 0 to 7) [1] are output from the ANEX0 pin. Amplify this output with an external op-amp before sending it back to the ANEX1 pin. The A/D conversion result is stored in the corresponding ADi register.

The A/D conversion speed depends on the response characteristics of the external op-amp.

Figure 16.9 shows an External Op-Amp Connection.

NOTE:

1. AN0_i and AN2_i can be used the same as ANi.

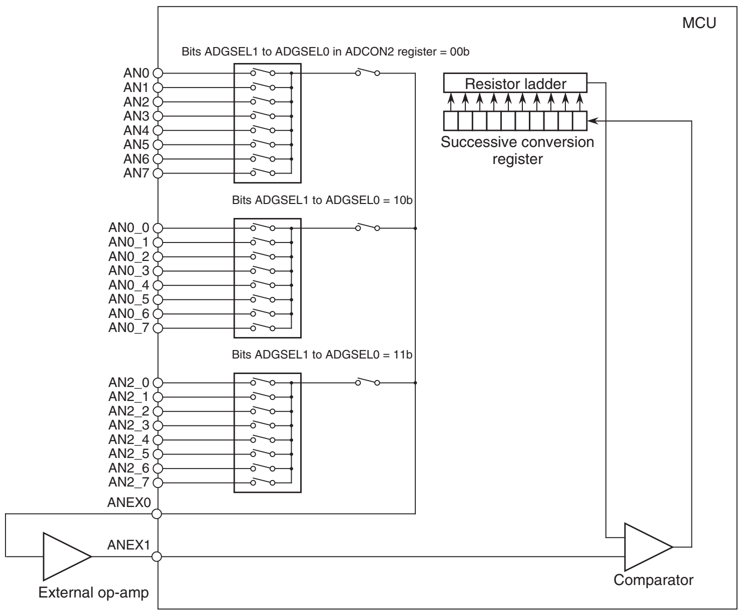


Figure 16.9 External Op-Amp Connection

16.2.5 Current Consumption Reducing Function

When not using the A/D converter, its resistor ladder and reference voltage input pin (VREF) can be separated using the VCUT bit in the ADCON1 register. When separated, no current will flow from the VREF pin into the resistor ladder, helping to reduce the power consumption of the chip.
To use the A/D converter, set the VCUT bit to 1 (VREF connected) and then set the ADST bit in the ADCON0 register to 1 (A/D conversion start). The VCUT and ADST bits cannot be set to 1 at the same time.
Nor can the VCUT bit be set to 0 (VREF unconnected) during A/D conversion.
Note that this does not affect VREF for the D/A converter (irrelevant).

16.2.6 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in Figure 16.10 has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, internal resistance of MCU be R, precision (error) of the A/D converter be X, and the resolution of A/D converter be Y (Y is 1024 in 10-bit mode, and 256 in 8-bit mode).

VC is generally $\mathrm{VC} = \mathrm{VIN}\,\{1 - e^{-\frac{1}{C\,(R0 + R)}t}\}$

And when t = T, $\mathrm{VC}=\mathrm{VIN} - \frac{X}{Y}\mathrm{VIN} = \mathrm{VIN}(1 - \frac{X}{Y})$

$$e^{-\frac{1}{C\,(R0+R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C\,(R0+R)}T = \ln\frac{X}{Y}$$

Hence, $R0 = -\dfrac{T}{C \bullet \ln\frac{X}{Y}} - R$

Figure 16.10 shows the Analog Input Pin and External Sensor Equivalent Circuit.
When the difference between VIN and VC becomes 0.1 LSB, we find impedance R0 when voltage between pins VC changes from 0 to VIN-(0.1/1024) VIN in time T. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is held to 0.1 LSB at time of A/D conversion in 10-bit mode. Actual error however is the value of absolute precision added to 0.1 LSB.
When f(ϕAD) = 10 MHz, T = 0.3 μs in the A/D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3 μs, R = 7.8 kΩ, C = 1.5 pF, X = 0.1, and Y = 1024. Hence,

$$R0 = -\frac{0.3 \times 10^{-6}}{1.5 \times 10^{-12} \bullet \ln\frac{0.1}{1024}} - 7.8 \times 10^{3} = 13.9 \times 10^{3}$$

Thus, the allowable output impedance of the sensor equivalent circuit, making the precision (error) 0.1 LSB or less, is approximately 13.9 kΩ. maximum.

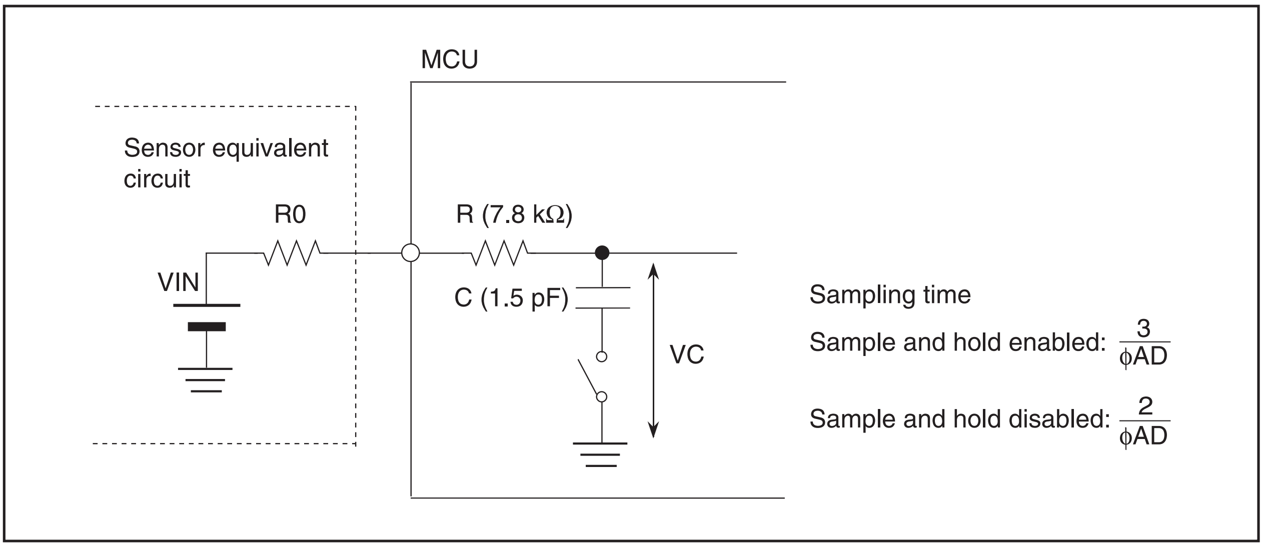

MCU
Sensor equivalent circuit
R0
R (7.8 kΩ)
VIN
C (1.5 pF)
VC
Sampling time
Sample and hold enabled: $\frac{3}{\phi AD}$
Sample and hold disabled: $\frac{2}{\phi AD}$

Figure 16.10 Analog Input Pin and External Sensor Equivalent Circuit

17. D/A Converter

This is an 8-bit, R-2R type D/A converter. These are two independent D/A converters.

D/A conversion is performed by writing to the DAi register (i = 0, 1). To output the result of conversion, set the DAiE bit in the DACON register to 1 (output enabled). Before D/A conversion can be used, the corresponding port direction bit is set to 0 (input mode). Setting the DAiE bit to 1 removes a pull-up from the corresponding port.

Output analog voltage (V) is determined by a set value (n : decimal) in the DAi register.

V = VREF × n/ 256 (n = 0 to 255)

VREF : reference voltage

Table 17.1 lists the D/A converter Performance. Figure 17.1 shows the D/A Converter Block Diagram. Figure 17.2 shows the D/A converter-related registers. Figure 17.3 shows the D/A Converter Equivalent Circuit.

Table 17.1 D/A Converter Performance

Item	Performance
D/A conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels (DA0 and DA1)

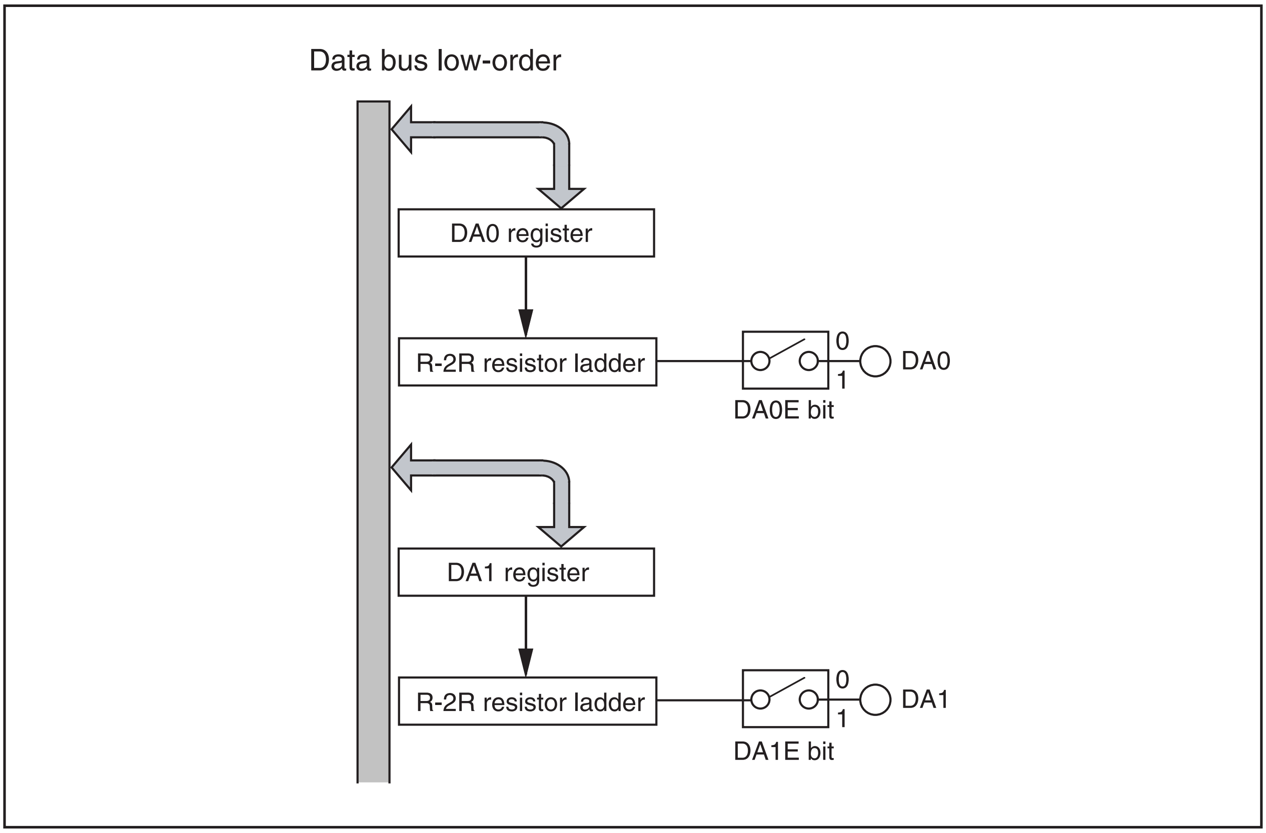
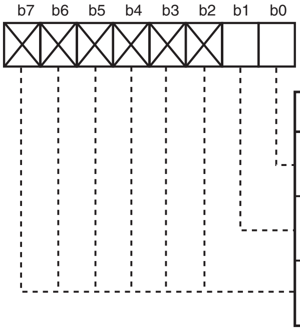


Figure 17.1 D/A Converter Block Diagram

D/A Control Register [1]



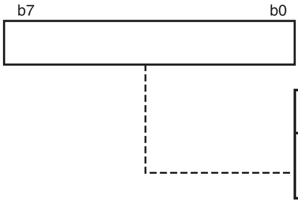
Symbol	Address	After Reset
DACON	03DCh	00h

Bit Symbol	Bit Name	Function	RW
DA0E	D/A0 output enable bit	0 : Output disabled 1 : Output enabled	RW
DA1E	D/A1 output enable bit	0 : Output disabled 1 : Output enabled	RW
- (b7-b2)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		–

NOTE:

1. When not using the D/A converter, set the DAiE bit (i = 0, 1) to 0 (output disabled) to reduce the unnecessary current consumption in the chip and set the DAi register to 00h to prevent current from flowing into the R-2R resistor.

D/A Register i (i = 0, 1) [1]



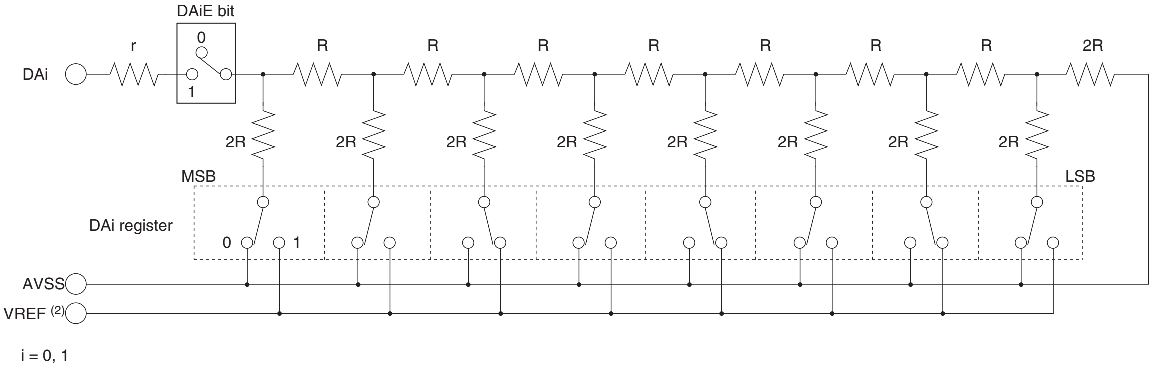
Symbol	Address	After Reset
DA0	03D8h	00h
DA1	03DAh	00h

Function	Setting Range	RW
Output value of D/A conversion	00h to FFh	RW

NOTE:

1. When not using the D/A converter, set the DAiE bit (i = 0, 1) to 0 (output disabled) to reduce the unnecessary current consumption in the chip and set the DAi register to 00h to prevent current from flowing into the R-2R resistor.

Figure 17.2 Registers DACON, DA0, and DA1



NOTES:

1. The above diagram shows an instance in which the DAi register is assigned 2Ah.
2. VREF is not related to VCUT bit setting in the ADCON1 register.

Figure 17.3 D/A Converter Equivalent Circuit

18. CRC Calculation

The Cyclic Redundancy Check (CRC) operation detects an error in data blocks. The MCU uses a generator polynomial of CRC-CCITT ($X^{16} + X^{12} + X^{5} + 1$) to generate CRC code.

The CRC code consists of 16 bits which are generated for each data block in given length, separated in 8-bit unit. After the initial value is set in the CRCD register, the CRC code is set in that register each time one byte of data is written to the CRCIN register. CRC code generation for one-byte data is finished in two cycles.

Figure 18.1 shows the CRC Circuit Block Diagram. Figure 18.2 shows the CRC-related registers. Figure 18.3 shows the calculation example using the CRC operation.

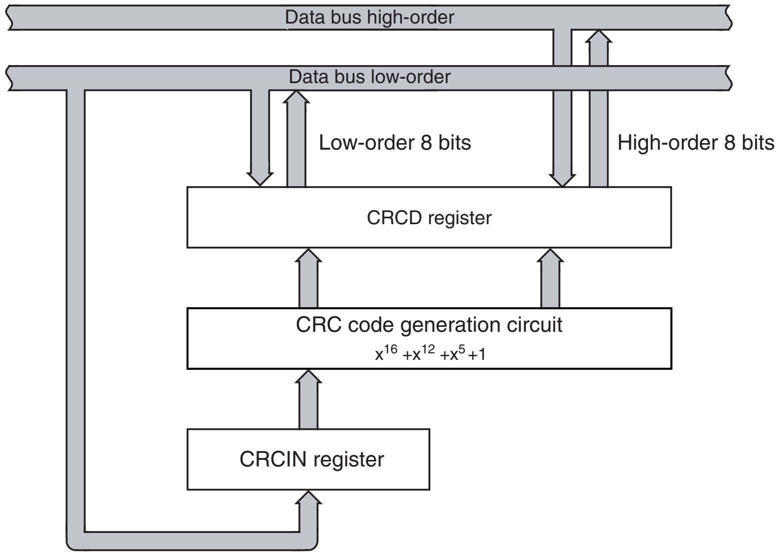
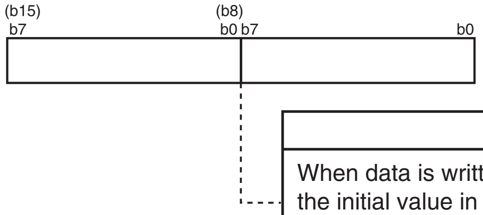


Figure 18.1 CRC Circuit Block Diagram

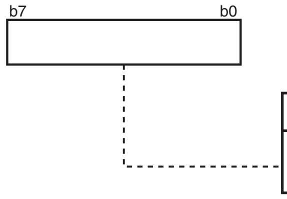
CRC Data Register



Symbol	Address	After Reset
CRCD	03BDh to 03BCh	Undefined

Function	Setting Range	RW
When data is written to the CRCIN register after setting the initial value in the CRCD register, the CRC code can be read out from the CRCD register.	0000h to FFFFh	RW

CRC Input Register



Symbol	Address	After Reset
CRCIN	03BEh	Undefined

Function	Setting Range	RW
Data input	00h to FFh	RW

Figure 18.2 Registers CRCD and CRCIN

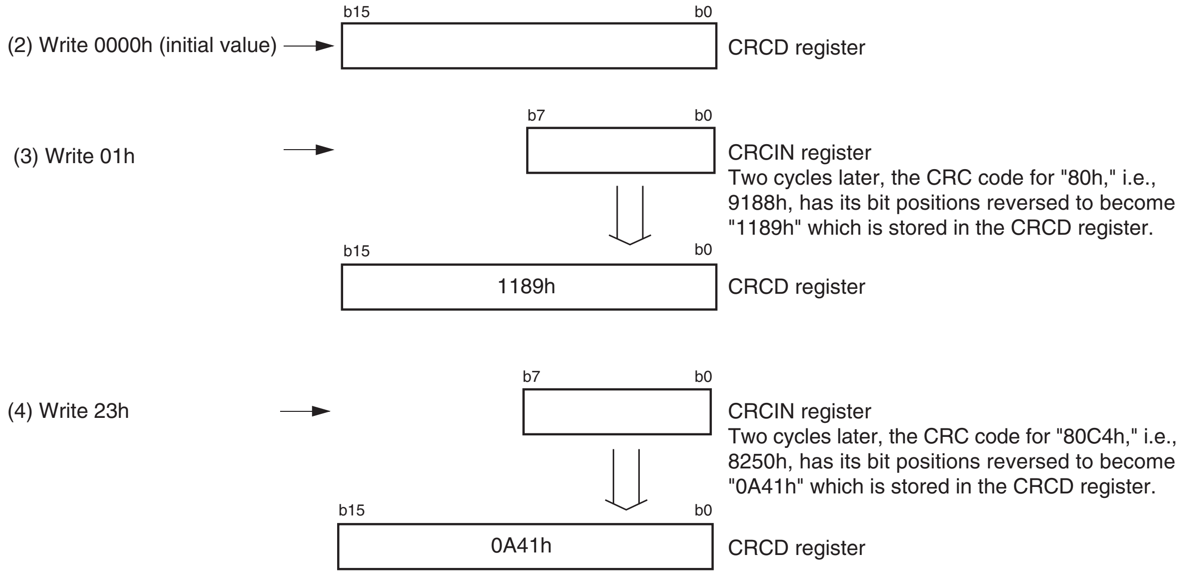
Setup procedure and CRC operation when generating CRC code "80C4h"

- CRC operation performed by the M16C

CRC code: Remainder of a division in which the value written to the CRCIN register with its bit positions reversed is divided by the generator polynomial
Generator polynomial: $X^6 + X^{12} + X^5 + 1$(1 0001 0000 0010 0001b)

- Setting procedure

(1) Reverse the bit positions of the value "80C4h" by program in 1-byte unit.
"80h" → "01h", "C4h" → "23h"


b15
b0
(2) Write 0000h (initial value)
CRCD register
b7
b0
(3) Write 01h
CRCIN register
Two cycles later, the CRC code for "80h," i.e., 9188h, has its bit positions reversed to become "1189h" which is stored in the CRCD register.
b15
b0
1189h
CRCD register
b7
b0
(4) Write 23h
CRCIN register
Two cycles later, the CRC code for "80C4h," i.e., 8250h, has its bit positions reversed to become "0A41h" which is stored in the CRCD register.
b15
b0
0A41h
CRCD register

- Details of CRC operation

As shown in (3) above, bit position of "01h" (00000001b) written to the CRCIN register is inversed and becomes "10000000b". Add "1000 0000 0000 0000 0000 0000b", as "10000000b" plus 16 digits, to "0000 0000 0000 0000 0000 0000b", as "0000 0000 0000 0000b" plus 8 digits as the default value of the CRCD register to perform the modulo-2 division.

```
                                            1000  1000
   1  0001  0000  0010  0001 | 1000  0000  0000  0000  0000  0000  ← Data
  ↗                            1000  1000  0001  0000  1
Generator polynomial                 1000  0001  0000  1000  0
                                     1000  1000  0001  0000  1
                                           1001  0001  1000 1000
                                         ↗
                                    CRC code
```

Modulo-2 operation is operation that complies with the law given below.

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
-1 = 1

"0001 0001 1000 1001b (1189h)", the remainder "1001 0001 1000 1000b (9188h)" with inversed bit position, can be read from the CRCD register.

When going on to (4) above, "23h (00100011b)" written in the CRCIN register is inversed and becomes "11000100b". Add "1100 0100 0000 0000 0000 0000b", as "11000100b" plus 16 digits, to "1001 0001 1000 1000 0000 0000b", as "1001 0001 1000 1000b" plus 8 digits as a remainder of (3) left in the CRCD register to perform the modulo-2 division. "0000 1010 0100 0001b (0A41h)", the remainder with inversed bit position, can be read from CRCD register.

Figure 18.3 CRC Calculation

19. CAN Module

The CAN (Controller Area Network) module for the M16C/6N Group (M16C/6NL, M16C/6NN) of MCUs is a communication controller implementing the CAN 2.0B protocol. The M16C/6N Group (M16C/6NL, M16C/6NN) contains one CAN module which can transmit and receive messages in both standard (11-bit) ID and extended (29-bit) ID formats.
Figure 19.1 shows the CAN Module Block Diagram.
External CAN bus driver and receiver are required.

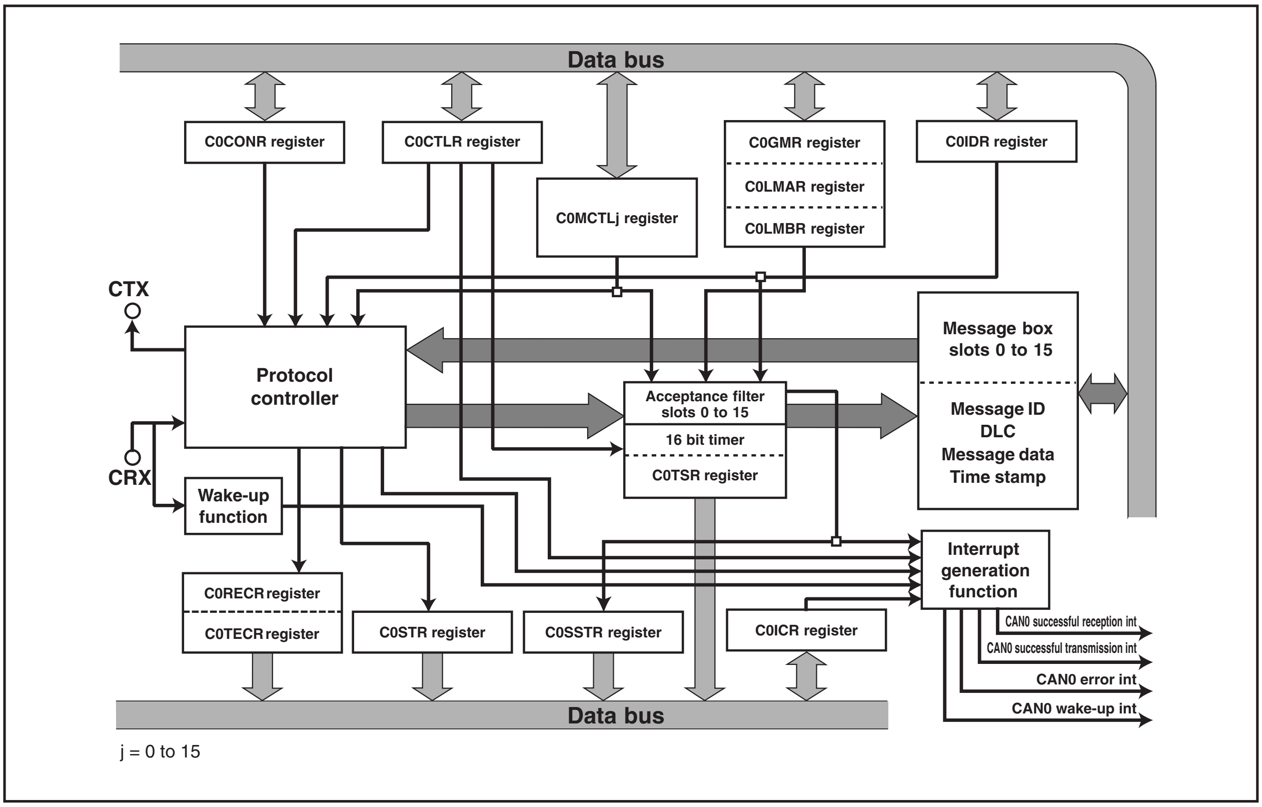


Figure 19.1 CAN Module Block Diagram

CTX/CRX: CAN I/O pins.

Protocol controller: This controller handles the bus arbitration and the CAN protocol services, i.e. bit timing, stuffing, error status etc.

Message box: This memory block consists of 16 slots that can be configured either as transmitter or receiver. Each slot contains an individual ID, data length code, a data field (8 bytes), and a time stamp.

Acceptance filter: This block performs filtering operation for received messages. For the filtering operation, the C0GMR register, the C0LMAR register, or the C0LMBR register is used.

16 bit timer: Used for the time stamp function. When the received message is stored in the message memory, the timer value is stored as a time stamp.

Wake-up function: CAN0 wake-up interrupt request is generated by a message from the CAN bus.

Interrupt generation function: The interrupt requests are generated by the CAN module. CAN0 successful reception interrupt, CAN0 successful transmission interrupt, CAN0 error interrupt, and CAN0 wake-up interrupt.

19.1 CAN Module-Related Registers

The CAN0 module has the following registers.

19.1.1 CAN0 Message Box

A CAN module is equipped with 16 slots (16 bytes or 8 words each). Slots 14 and 15 can be used as Basic CAN.

- Priority of the slots: The smaller the number of the slot, the higher the priority, in both transmission and reception.
- A program can define whether a slot is defined as transmitter or receiver.

19.1.2 Acceptance Mask Registers

A CAN module is equipped with 3 masks for the acceptance filter.

- CAN0 global mask register (C0GMR register: 6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slots 0 to 13
- CAN0 local mask A register (C0LMAR register: 6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slot 14
- CAN0 local mask B register (C0LMBR register: 6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slot 15

19.1.3 CAN SFR Registers

- CAN0 message control register j (j = 0 to 15) (C0MCTLj register: 8 bits × 16)
  Control of transmission and reception of a corresponding slot
- CANi control register (i = 0, 1) (CiCTLR register: 16 bits)
  Control of the CAN protocol
- CAN0 status register (C0STR register: 16 bits)
  Indication of the protocol status
- CAN0 slot status register (C0SSTR register: 16 bits)
  Indication of the status of contents of each slot
- CAN0 interrupt control register (C0ICR register: 16 bits)
  Selection of “interrupt enabled or disabled” for each slot
- CAN0 extended ID register (C0IDR register: 16 bits)
  Selection of ID format (standard or extended) for each slot
- CAN0 configuration register (C0CONR register: 16 bits)
  Configuration of the bus timing
- CAN0 receive error count register (C0RECR register: 8 bits)
  Indication of the error status of the CAN module in reception: the counter value is incremented or decremented according to the error occurrence.
- CAN0 transmit error count register (C0TECR register: 8 bits)
  Indication of the error status of the CAN module in transmission: the counter value is incremented or decremented according to the error occurrence.
- CAN0 time stamp register (C0TSR register: 16 bits)
  Indication of the value of the time stamp counter
- CAN0 acceptance filter support register (C0AFS register: 16 bits)
  Decoding the received ID for use by the acceptance filter support unit

Explanation of each register is given below.

19.2 CAN0 Message Box

Table 19.1 shows the CAN0 Message Box Memory Mapping.

It is possible to access to the message box in byte or word.

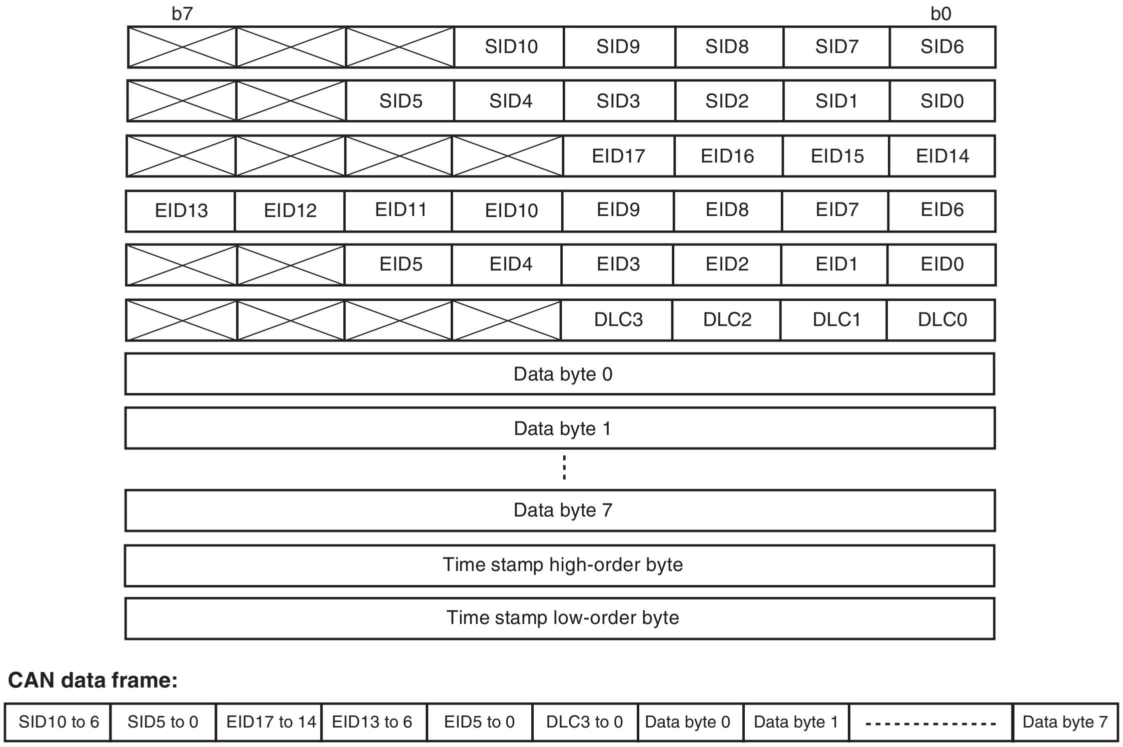
Mapping of the message contents differs from byte access to word access. Byte access or word access can be selected by the MsgOrder bit of the C0CTLR register.

Table 19.1 CAN0 Message Box Memory Mapping

Address	Message Content (Memory Mapping)	
	Byte Access (8 bits)	Word Access (16 bits)
0060h + n × 16 + 0	SID10 to SID6	SID5 to SID0
0060h + n × 16 + 1	SID5 to SID0	SID10 to SID6
0060h + n × 16 + 2	EID17 to EID14	EID13 to EID6
0060h + n × 16 + 3	EID13 to EID6	EID17 to EID14
0060h + n × 16 + 4	EID5 to EID0	Data length code (DLC)
0060h + n × 16 + 5	Data length code (DLC)	EID5 to EID0
0060h + n × 16 + 6	Data byte 0	Data byte 1
0060h + n × 16 + 7 ⋮ 0060h + n × 16 + 13	Data byte 1 ⋮ Data byte 7	Data byte 0 ⋮ Data byte 6
0060h + n × 16 + 14	Time stamp high-order byte	Time stamp low-order byte
0060h + n × 16 + 15	Time stamp low-order byte	Time stamp high-order byte

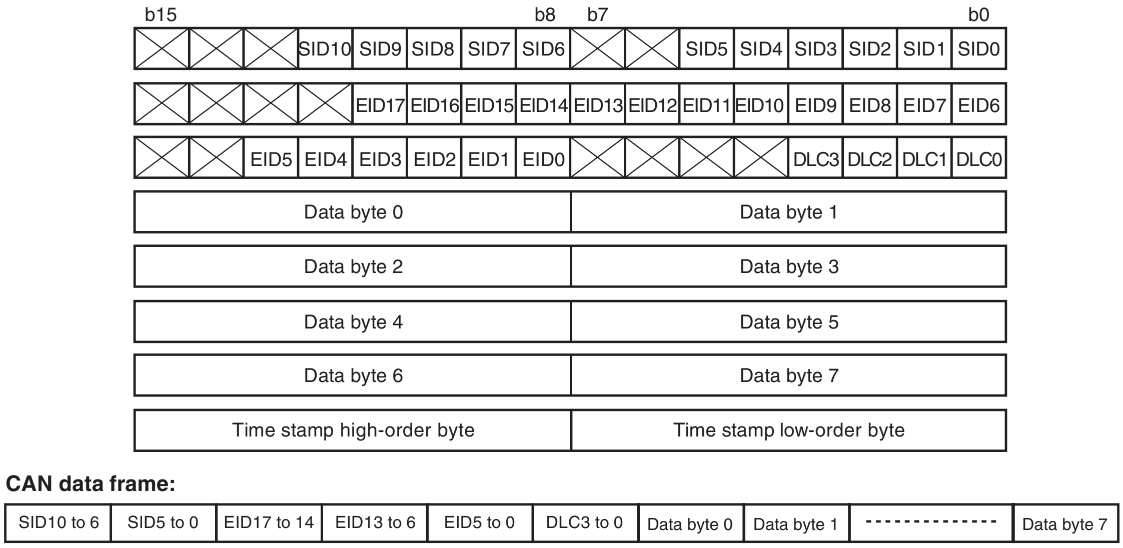
n = 0 to 15: the number of the slot

Figures 19.2 and 19.3 show the Bit Mapping in Byte Access and Word Access. The content of each slot remains unchanged unless transmission or reception of a new message is performed.



NOTE:
1. When ☒ is read, the value is the one written upon the transmission slot configuration.
The value is 0 when read on the reception slot configuration.

Figure 19.2 Bit Mapping in Byte Access

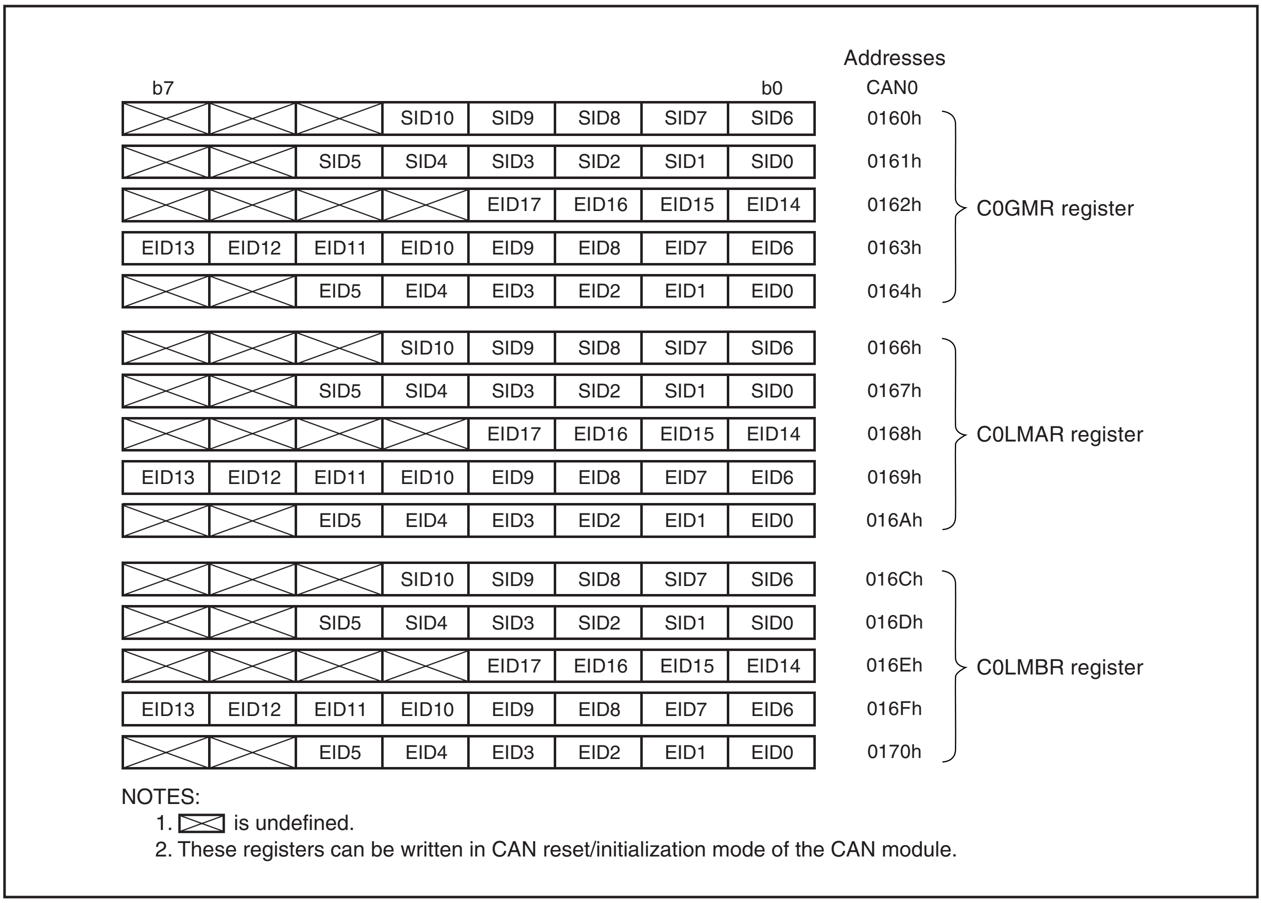


NOTE:
1. When ☒ is read, the value is the one written upon the transmission slot configuration.
The value is 0 when read on the reception slot configuration.

Figure 19.3 Bit Mapping in Word Access

19.3 Acceptance Mask Registers

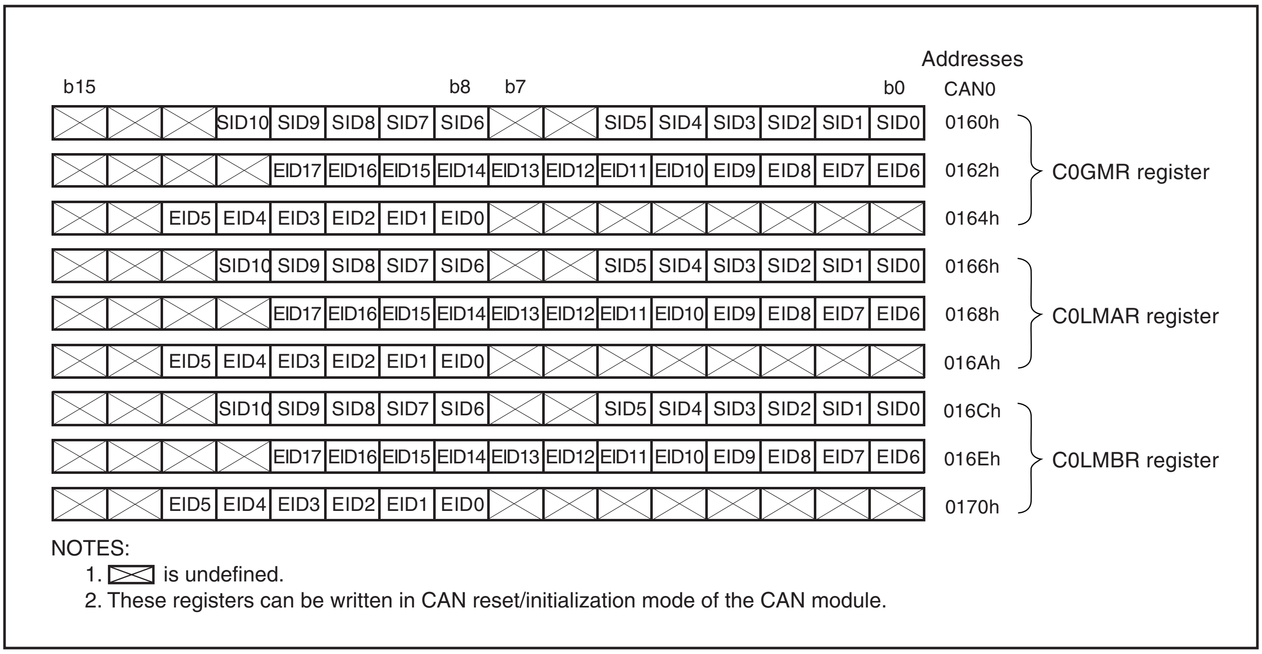
Figures 19.4 and 19.5 show the Mask Registers (registers C0GMR, C0LMAR, and C0LMBR) Bit Mapping in Byte Access and Word Access.



b7							b0	Addresses CAN0	
☒	☒	☒	SID10	SID9	SID8	SID7	SID6	0160h	C0GMR register
☒	☒	SID5	SID4	SID3	SID2	SID1	SID0	0161h	
☒	☒	☒	☒	EID17	EID16	EID15	EID14	0162h	
EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0163h	
☒	☒	EID5	EID4	EID3	EID2	EID1	EID0	0164h	
☒	☒	☒	SID10	SID9	SID8	SID7	SID6	0166h	C0LMAR register
☒	☒	SID5	SID4	SID3	SID2	SID1	SID0	0167h	
☒	☒	☒	☒	EID17	EID16	EID15	EID14	0168h	
EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0169h	
☒	☒	EID5	EID4	EID3	EID2	EID1	EID0	016Ah	
☒	☒	☒	SID10	SID9	SID8	SID7	SID6	016Ch	C0LMBR register
☒	☒	SID5	SID4	SID3	SID2	SID1	SID0	016Dh	
☒	☒	☒	☒	EID17	EID16	EID15	EID14	016Eh	
EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	016Fh	
☒	☒	EID5	EID4	EID3	EID2	EID1	EID0	0170h	

NOTES:
1. ☒ is undefined.
2. These registers can be written in CAN reset/initialization mode of the CAN module.

Figure 19.4 Mask Registers Bit Mapping in Byte Access



b15							b8	b7							b0	Addresses CAN0	
☒	☒	☒	SID10	SID9	SID8	SID7	SID6	☒	☒	SID5	SID4	SID3	SID2	SID1	SID0	0160h	C0GMR register
☒	☒	☒	☒	EID17	EID16	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0162h	
☒	☒	EID5	EID4	EID3	EID2	EID1	EID0	☒	☒	☒	☒	☒	☒	☒	☒	0164h	
☒	☒	☒	SID10	SID9	SID8	SID7	SID6	☒	☒	SID5	SID4	SID3	SID2	SID1	SID0	0166h	C0LMAR register
☒	☒	☒	☒	EID17	EID16	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	0168h	
☒	☒	EID5	EID4	EID3	EID2	EID1	EID0	☒	☒	☒	☒	☒	☒	☒	☒	016Ah	
☒	☒	☒	SID10	SID9	SID8	SID7	SID6	☒	☒	SID5	SID4	SID3	SID2	SID1	SID0	016Ch	C0LMBR register
☒	☒	☒	☒	EID17	EID16	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6	016Eh	
☒	☒	EID5	EID4	EID3	EID2	EID1	EID0	☒	☒	☒	☒	☒	☒	☒	☒	0170h	

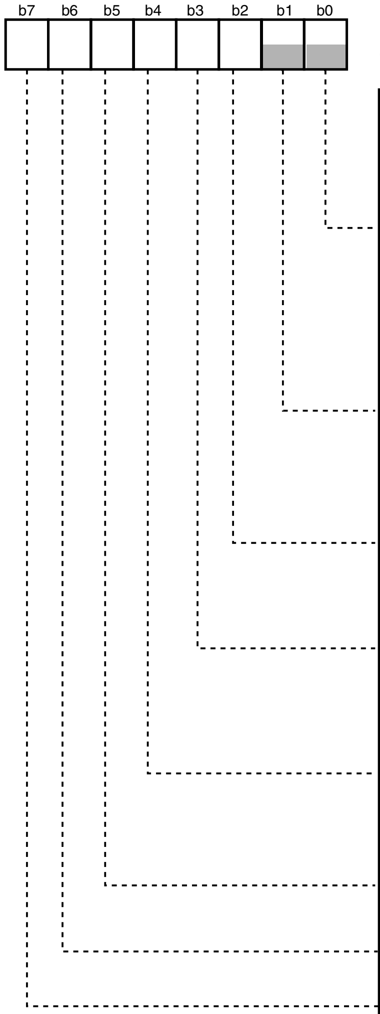
NOTES:
1. ☒ is undefined.
2. These registers can be written in CAN reset/initialization mode of the CAN module.

Figure 19.5 Mask Registers Bit Mapping in Word Access

19.4 CAN SFR Registers

Figures 19.6 to 19.12 show the CAN SFR registers.

CAN0 Message Control Register j (j = 0 to 15) (4)



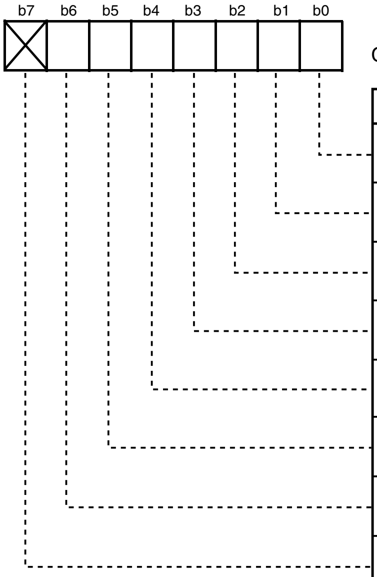
Symbol	Address	After Reset
C0MCTL0 to C0MCTL15	0200h to 020Fh	00h

Bit Symbol	Bit Name	Function	RW
NewData	Successful reception flag	When set to reception slot 0: The content of the slot is read or still under processing by the CPU. 1 The CAN module has stored new data in the slot.	RO (1)
SentData	Successful transmission flag	When set to transmission slot 0: Transmission is not started or completed yet. 1: Transmission is successfully completed.	RO (1)
InvalData	"Under reception" flag	When set to reception slot 0: The message is valid. 1: The message is invalid. (The message is being updated.)	RO
TrmActive	"Under transmission" flag	When set to transmission slot 0: Waiting for bus idle or completion of arbitration. 1: Transmitting	RO
MsgLost	Overwrite flag	When set to reception slot 0: No message has been overwritten in this slot. 1: This slot already contained a message, but it has been overwritten by a new one.	RO (1)
RemActive	Remote frame transmission/ reception status flag (2)	0: Data frame transmission/reception status 1: Remote frame transmission/reception status	RW
RspLock	Auto response lock mode select bit	When set to reception remote frame slot 0: After a remote frame is received, it will be answered automatically. 1: After a remote frame is received, no transmission will be started as long as this bit is set to 1. (Not responding)	RW
Remote	Remote frame corresponding slot select bit	0: Slot not corresponding to remote frame 1: Slot corresponding to remote frame	RW
RecReq	Reception slot request bit (3)	0: Not reception slot 1: Reception slot	RW
TrmReq	Transmission slot request bit (3)	0: Not transmission slot 1: Transmission slot	RW

NOTES:
1. As for write, only writing 0 is possible. The value of each bit is written when the CAN module enters the respective state.
2. In Basic CAN mode, slots 14 and 15 serve as data format identification flag.
The RemActive bit is set to 0 if the data frame is received and it is set to 1 if the remote frame is received.
3. One slot cannot be defined as reception slot and transmission slot at the same time.
4. This register cannot be set in CAN reset/initialization mode of the CAN module.

Figure 19.6 C0MCTLj Register

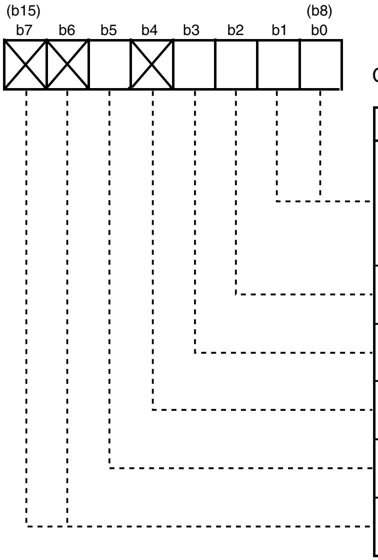
CAN0 Control Register



Symbol C0CTLR | Address 0210h | After Reset X0000001b

Bit Symbol	Bit Name	Function	RW
Reset	CAN module reset bit (1)	0: Operation mode 1: Reset/initialization mode	RW
LoopBack	Loop back mode select bit (2)	0: Loop back mode disabled 1: Loop back mode enabled	RW
MsgOrder	Message order select bit (2)	0: Word access 1: Byte access	RW
BasicCAN	Basic CAN mode select bit (2)	0: Basic CAN mode disabled 1: Basic CAN mode enabled	RW
BusErrEn	Bus error interrupt enable bit (2)	0: Bus error interrupt disabled 1: Bus error interrupt enabled	RW
Sleep	Sleep mode select bit (2) (3)	0: Sleep mode disabled 1: Sleep mode enabled; clock supply stopped	RW
PortEn	CAN port enable bit (2) (3)	0: I/O port function 1: CTX/CRX function (4)	RW
– (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTES:
1. When the Reset bit is set to 1 (CAN reset/initialization mode), check that the State_Reset bit in the C0STR register is set to 1 (reset mode).
2. Change this bit only in CAN reset/initialization mode.
3. When using CAN0 wake-up interrupt, set these bits to 1.
4. When the PortEn bit is set to 1 (CTX/CRX function), set the corresponding port direction bit for the CRX0 pin to 0 (input mode).



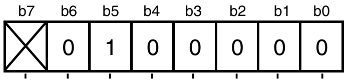
Symbol C0CTLR | Address 0211h | After Reset XX0X0000b

Bit Symbol	Bit Name	Function	RW
TSPreScale	Time stamp prescaler (3)	b1 b0 0 0: Period of 1 bit time 0 1: Period of 1/2 bit time 1 0: Period of 1/4 bit time 1 1: Period of 1/8 bit time	RW
TSReset	Time stamp counter reset bit (1)	0: Nothing is occurred. 1: Force reset of the time stamp counter	RW
RetBusOff	Return from bus off command bit (2)	0: Nothing is occurred. 1: Force return from bus off	RW
– (b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
RXOnly	Listen-only mode select bit (3)	0: Listen-only mode disabled 1: Listen-only mode enabled (4)	RW
– (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTES:
1. When the TSReset bit = 1, the C0TSR register is set to 0000h. After this, the bit is automatically set to 0.
2. When the RetBusOff bit = 1, registers C0RECR and C0TECR are set to 00h. After this, this bit is automatically set to 0.
3. Change this bit only in CAN reset/initialization mode.
4. When listen-only mode is selected, do not request the transmission.

Figure 19.7 C0CTLR Register

CAN1 Control Register [1]

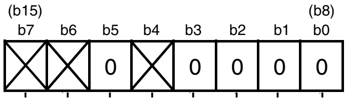


Symbol	Address	After Reset
C1CTLR	0230h	X0000001b

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved bits	Set to 0	RW
— (b5)	Reserved bit	Set to 1	RW
— (b6)	Reserved bit	Set to 0	RW
— (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—

NOTE:

1. Make sure 0020h is set to this register (addresses 0230h, 0231h). Moreover, make sure the CCLKR register is set after setting 0020h to this register.

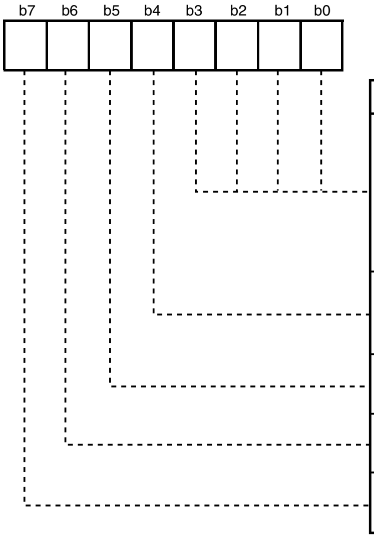


Symbol	Address	After Reset
C1CTLR	0231h	XX0X0000b

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved bits	Set to 0	RW
— (b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—
— (b5)	Reserved bit	Set to 0	RW
— (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		—

Figure 19.8 C1CTLR Register

CAN0 Status Register

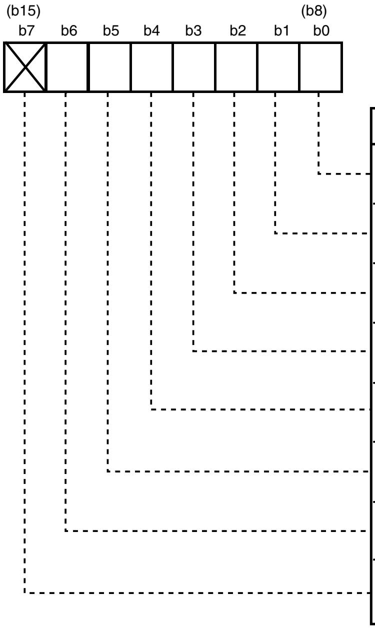


Symbol	Address	After Reset
C0STR	0212h	00h

Bit Symbol	Bit Name	Function	RW
MBOX	Active slot bits [1]	b3 b2 b1 b0 0 0 0 0 : Slot 0 0 0 0 1 : Slot 1 0 0 1 0 : Slot 2 : 1 1 1 0 : Slot 14 1 1 1 1 : Slot 15	RO
TrmSucc	Successful transmission flag [1]	0: No [successful] transmission 1: The CAN module has transmitted a message successfully.	RO
RecSucc	Successful reception flag [1]	0: No [successful] reception 1: CAN module received a message successfully.	RO
TrmState	Transmission flag (transmitter)	0: CAN module is idle or receiver. 1: CAN module is transmitter.	RO
RecState	Reception flag (receiver)	0: CAN module is idle or transmitter. 1: CAN module is receiver.	RO

NOTE:

1. These bits can be changed only when a slot which an interrupt is enabled by the C0ICR register is transmitted or received successfully.

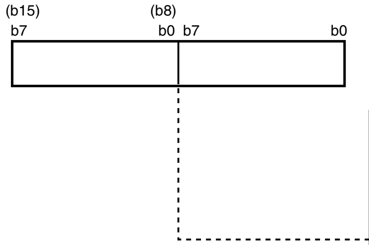


Symbol	Address	After Reset
C0STR	0213h	X0000001b

Bit Symbol	Bit Name	Function	RW
State_Reset	Reset state flag	0: Operation mode 1: Reset mode	RO
State_ LoopBack	Loop back state flag	0: Not Loop back mode 1: Loop back mode	RO
State_ MsgOrder	Message order state flag	0:Word access 1: Byte access	RO
State_ BasicCAN	Basic CAN mode state flag	0: Not Basic CAN mode 1: Basic CAN mode	RO
State_ BusError	Bus error state flag	0: No error has occurred. 1: A CAN bus error has occurred.	RO
State_ ErrPass	Error passive state flag	0: CAN module is not in error passive state. 1: CAN module is in error passive state.	RO
State_ BusOff	Error bus off state flag	0: CAN module is not in error bus off state. 1: CAN module is in error bus off state.	RO
– (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

Figure 19.9 C0STR Register

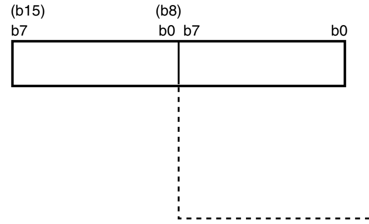
CAN0 Slot Status Register



Symbol	Address	After Reset
C0SSTR	0215h, 0214h	0000h

Function	Setting Values	RW
Slot status bits Each bit corresponds to the slot with the same number.	0: Reception slot The message has been read. Transmission slot Transmission is not completed. 1: Reception slot The message has not been read. Transmission slot Transmission is completed.	RO

CAN0 Interrupt Control Register [1]



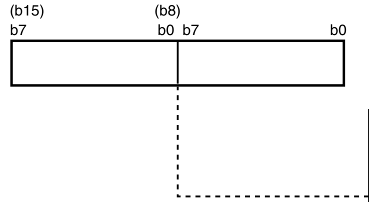
Symbol	Address	After Reset
C0ICR	0217h, 0216h	0000h

Function	Setting Values	RW
Interrupt enable bits: Each bit corresponds with a slot with the same number. Enabled/disabled of successful transmission interrupt or successful reception interrupt can be selected.	0: Interrupt disabled 1: Interrupt enabled	RW

NOTE:

1. This register cannot be set in CAN reset/initialization mode of the CAN module.

CAN0 Extended ID Register [1]



Symbol	Address	After Reset
C0IDR	0219h, 0218h	0000h

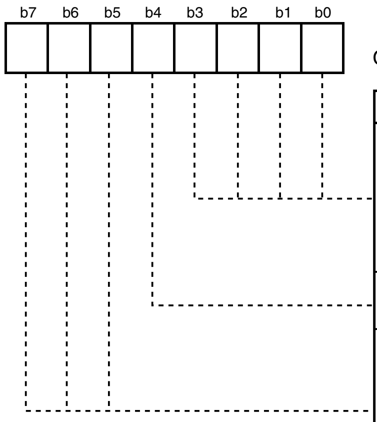
Function	Setting Values	RW
Extended ID bits: Each bit corresponds with a slot with the same number. Selection of the ID format that each slot handles.	0: Standard ID 1: Extended ID	RW

NOTE:

1. This register cannot be set in CAN reset/initialization mode of the CAN module.

Figure 19.10 Registers C0SSTR, C0ICR, and C0IDR

CAN0 Configuration Register

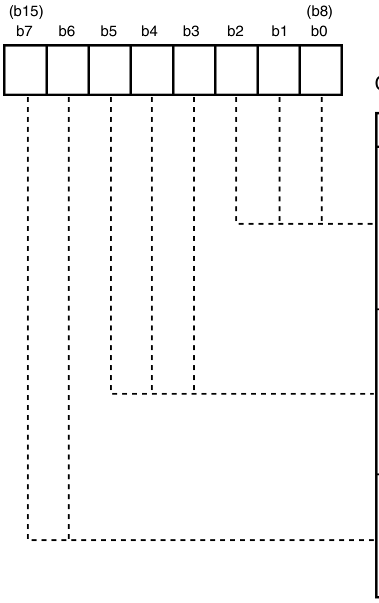


Symbol	Address	After Reset
C0CONR	021Ah	Undefined

Bit Symbol	Bit Name	Function	RW
BRP	Prescaler division ratio select bits	b3 b2 b1 b0 0 0 0 0 : Divide-by-1 of fCAN 0 0 0 1 : Divide-by-2 of fCAN 0 0 1 0 : Divide-by-3 of fCAN ⋮ 1 1 1 0 : Divide-by-15 of fCAN 1 1 1 1 : Divide-by-16 of fCAN (1)	RW
SAM	Sampling control bit	0 : One time sampling 1 : Three times sampling	RW
PTS	Propagation time segment control bits	b7 b6 b5 0 0 0 : 1Tq 0 0 1 : 2Tq 0 1 0 : 2Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	RW

NOTE:

1. fCAN serves for the CAN clock. The period is decided by configuration of the CCLKi bit (i = 0 to 2) in the CCLKR register.

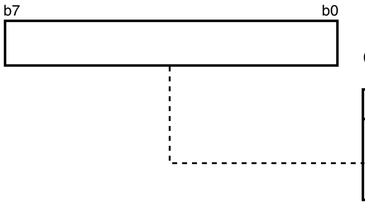


Symbol	Address	After Reset
C0CONR	021Bh	Undefined

Bit Symbol	Bit Name	Function	RW
PBS1	Phase buffer segment 1 control bits	b2 b1b0 0 0 0 : Do not set a value 0 0 1 : 2Tq 0 1 0 : 3Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	RW
PBS2	Phase buffer segment 2 control bits	b5 b4 b3 0 0 0 : Do not set a value 0 0 1 : 2Tq 0 1 0 : 3Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	RW
SJW	Resynchronization jump width control bits	b7 b6 0 0 : 1Tq 0 1 : 2Tq 1 0 : 3Tq 1 1 : 4Tq	RW

Figure 19.11 C0CONR Register

CAN0 Receive Error Count Register

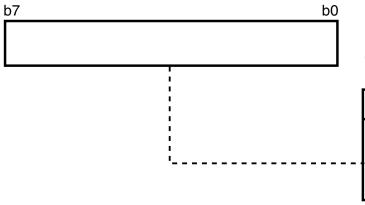


Symbol	Address	After Reset
C0RECR	021Ch	00h

Function	Counter Value	RW
Reception error counting function The value is incremented or decremented according to the CAN module's error status.	00h to FFh (1)	RO

NOTE:
1. The value is undefined in bus off state.

CAN0 Transmit Error Count Register

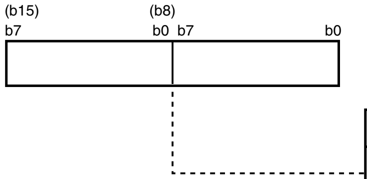


Symbol	Address	After Reset
C0TECR	021Dh	00h

Function	Counter Value	RW
Transmission error counting function The value is incremented or decremented according to the CAN module's error status.	00h to FFh (1)	RO

NOTE:
1. The value is undefined in bus off state.

CAN0 Time Stamp Register (1)

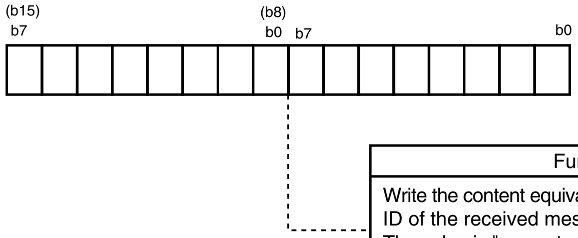


Symbol	Address	After Reset
C0TSR	021Fh, 021Eh	0000h

Function	Counter Value	RW
Time stamp function	0000h to FFFFh	RO

NOTE:
1. Use a 16-bit data for read.

CAN0 Acceptance Filter Support Register



Symbol	Address	After Reset
C0AFS	0243h, 0242h	Indeterminate

Function	Setting Values	RW
Write the content equivalent to the standard frame ID of the received message. The value is "converted standard frame ID" when read.	Standard frame ID	RW

Figure 19.12 Registers C0RECR, C0TECR, C0TSR, and C0AFS

19.5 Operational Modes

The CAN module has the following four operational modes.

- CAN Reset/Initialization Mode
- CAN Operation Mode
- CAN Sleep Mode
- CAN Interface Sleep Mode

Figure 19.13 shows the Transition between Operational Modes.

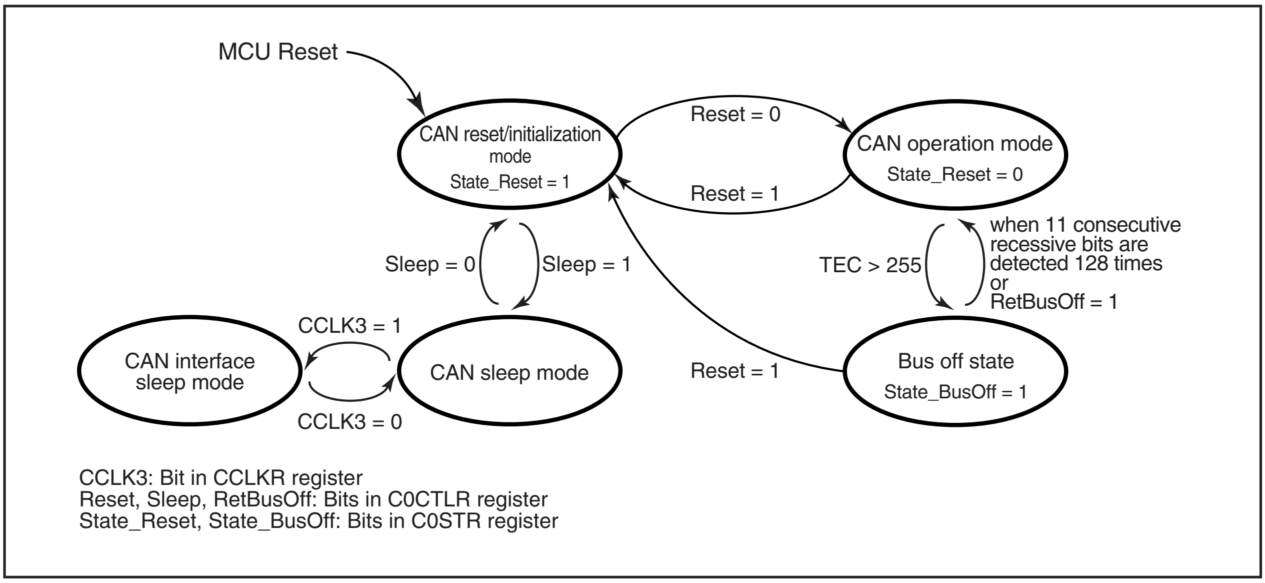

MCU Reset
CAN reset/initialization mode
State_Reset = 1
Reset = 0
Reset = 1
CAN operation mode
State_Reset = 0
Sleep = 0
Sleep = 1
TEC > 255
when 11 consecutive recessive bits are detected 128 times or RetBusOff = 1
CCLK3 = 1
CAN interface sleep mode
CAN sleep mode
CCLK3 = 0
Reset = 1
Bus off state
State_BusOff = 1
CCLK3: Bit in CCLKR register
Reset, Sleep, RetBusOff: Bits in C0CTLR register
State_Reset, State_BusOff: Bits in C0STR register

Figure 19.13 Transition between Operational Modes

19.5.1 CAN Reset/Initialization Mode

CAN reset/initialization mode is activated upon MCU reset or by setting the Reset bit in the C0CTLR register to 1. If the Reset bit is set to 1, check that the State_Reset bit in the C0STR register is set to 1. Entering CAN reset/initialization mode initiates the following functions by the module:

- CAN communication is impossible.
- When CAN reset/initialization mode is activated during an ongoing transmission in operation mode, the module suspends the mode transition until completion of the transmission (successful, arbitration loss, or error detection). Then, the State_Reset bit is set to 1, and CAN reset/initialization mode is activated.
- Registers C0MCTLj (j = 0 to 15), C0STR, C0ICR, C0IDR, C0RECR, C0TECR, and C0TSR are initialized. All these registers are locked to prevent CPU modification.
- Registers C0CTLR, C0CONR, C0GMR, C0LMAR, and C0LMBR, and the CAN0 message box retain their contents and are available for CPU access.

19.5.2 CAN Operation Mode

CAN operation mode is activated by setting the Reset bit in the C0CTLR register to 0. If the Reset bit is set to 0, check that the State_Reset bit in the C0STR register is set to 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode, the module initiates the following functions:

- The module's communication functions are released and it becomes an active node on the network and may transmit and receive CAN messages.
- Release the internal fault confinement logic including receive and transmit error counters. The module may leave CAN operation mode depending on the error counts.

Within CAN operation mode, the module may be in three different sub modes, depending on which type of communication functions are performed:

- Module idle : The modules receive and transmit sections are inactive.
- Module receives : The module receives a CAN message sent by another node.
- Module transmits : The module transmits a CAN message. The module may receive its own message simultaneously when the LoopBack bit in the C0CTLR register = 1 (Loop back mode enabled).

Figure 19.14 shows the Sub Modes of CAN Operation Mode.

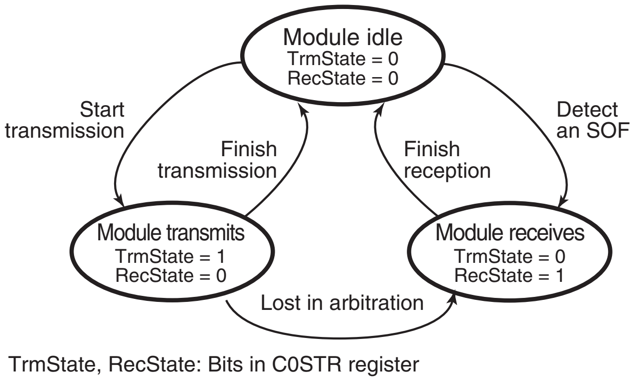


Figure 19.14 Sub Modes of CAN Operation Mode

19.5.3 CAN Sleep Mode

CAN sleep mode is activated by setting the Sleep bit to 1 in the C0CTLR register. It should never be activated from the CAN operation mode but only via CAN reset/initialization mode.

Entering CAN sleep mode instantly stops the clock supply to the module and thereby reduces power dissipation.

19.5.4 CAN Interface Sleep Mode

CAN interface sleep mode is activated by setting the CCLK3 bit in the CCLKR register to 1. It should never be activated but only via CAN sleep mode.

Entering CAN interface sleep mode instantly stops the clock supply to the CPU Interface in the module and thereby reduces power dissipation.

19.5.5 Bus Off State

The bus off state is entered according to the fault confinement rules of the CAN specification. When returning to CAN operation mode from the bus off state, the module has the following two cases.

In this time, the value of any CAN registers, except registers C0STR, C0RECR and C0TECR, does not change.

(1) When 11 consecutive recessive bits are detected 128 times

The module enters instantly into error active state and the CAN communication becomes possible immediately.

(2) When the RetBusOff bit in the C0CTLR register = 1 (Force return from buss off)

The module enters instantly into error active state, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected.

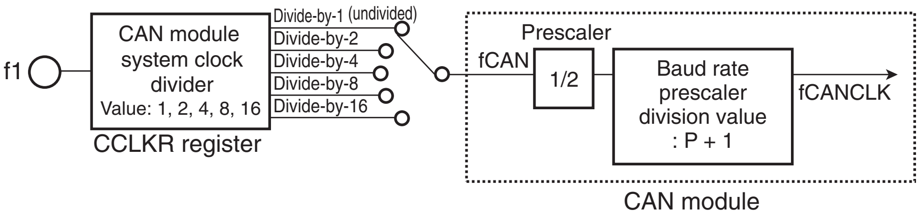
19.6 CAN Module System Clock Configuration

The M16C/6N Group (M16C/6NL, M16C/6NN) has a CAN module system clock select circuit.

Configuration of the CAN module system clock can be done through manipulating the CCLKR register and the BRP bit in the C0CONR register.

For the CCLKR register, refer to **8. Clock Generation Circuit**.

Figure 19.15 shows the CAN Module System Clock Generation Circuit Block Diagram.



fCAN : CAN module system clock
P : The value written in the BRP bit in the C0CONR register. P = 0 to 15
fCANCLK : CAN communication clock fCANCLK = fCAN/2(P + 1)

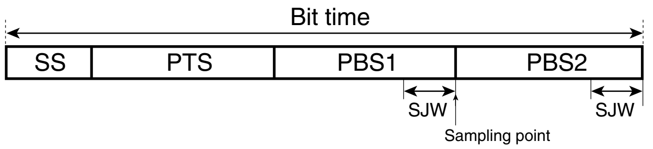
Figure 19.15 CAN Module System Clock Generation Circuit Block Diagram

19.7 Bit Timing Configuration

The bit time consists of the following four segments:

- Synchronization segment (SS)
  This serves for monitoring a falling edge for synchronization.
- Propagation time segment (PTS)
  This segment absorbs physical delay on the CAN network which amounts to double the total sum of delay on the CAN bus, the input comparator delay, and the output driver delay.
- Phase buffer segment 1 (PBS1)
  This serves for compensating the phase error. When the falling edge of the bit falls later than expected, the segment can become longer by the maximum of the value defined in SJW.
- Phase buffer segment 2 (PBS2)
  This segment has the same function as the phase buffer segment 1. When the falling edge of the bit falls earlier than expected, the segment can become shorter by the maximum of the value defined in SJW.

Figure 19.16 shows the Bit Timing.



The range of each segment: Bit time = 8 to 25Tq
SS = 1Tq
PTS = 1Tq to 8Tq
PBS1 = 2Tq to 8Tq
PBS2 = 2Tq to 8Tq
SJW = 1Tq to 4Tq

Configuration of PBS1 and PBS2: PBS1 ≥ PBS2
PBS1 ≥ SJW
PBS2 ≥ 2 when SJW = 1
PBS2 ≥ SJW when 2 ≤ SJW ≤ 4

Figure 19.16 Bit Timing

19.8 Bit-rate

Bit-rate depends on f1, the division value of the CAN module system clock, the division value of the baud rate prescaler, and the number of Tq of one bit.

Table 19.2 shows the Examples of Bit-rate.

Table 19.2 Examples of Bit-rate

Bit-rate	24 MHz	20 MHz	16 MHz	10 MHz	8 MHz
1 Mbps	12 Tq (1)	10 Tq (1)	8 Tq (1)	–	–
500 kbps	8 Tq (3) 12 Tq (2) 24 Tq (1)	10 Tq (2) 20 Tq (1) –	8 Tq (2) 16 Tq (1) –	10 Tq (1) – –	8 Tq (1) – –
125 kbps	8 Tq (12) 12 Tq (8) 16 Tq (6) 24 Tq (4)	8 Tq (10) 10 Tq (8) 16 Tq (5) 20 Tq (4)	8 Tq (8) 16 Tq (4) – –	8 Tq (5) 10 Tq (4) 20 Tq (2) –	8 Tq (4) 16 Tq (2) – –
83.3 kbps	8 Tq (18) 12 Tq (12) 16 Tq (9) 24 Tq (6)	8 Tq (15) 10 Tq (12) 20 Tq (6) –	8 Tq (12) 16 Tq (6) – –	10 Tq (6) 20 Tq (3) – –	8 Tq (6) 16 Tq (3) – –
33.3 kbps	10 Tq (36) 12 Tq (30) 20 Tq (18) 24 Tq (15)	10 Tq (30) 20 Tq (15) – –	8 Tq (30) 10 Tq (24) 16 Tq (15) 20 Tq (12)	10 Tq (15) – – –	8 Tq (15) 10 Tq (12) 20 Tq (6) –

NOTE:

1. The number in ( ) indicates a value of “fCAN division value” multiplied by “baud rate prescaler division value”.

19.8.1 Calculation of Bit-rate

$$\frac{\text{f1}}{2 \times \text{“fCAN division value}^{(1)}\text{”} \times \text{“baud rate prescaler division value}^{(2)}\text{”} \times \text{“number of Tq of one bit”}}$$

NOTES:

1. fCAN division value = 1, 2, 4, 8, 16
   fCAN division value: a value selected in the CCLKR register
2. Baud rate prescaler division value = P + 1 (P: 0 to 15)
   P: a value selected in the BRP bit in the C0CONR register

19.9 Acceptance Filtering Function and Masking Function

These functions serve the users to select and receive a facultative message. Registers C0GMR, C0LMAR, and C0LMBR can perform masking to the standard ID and the extended ID of 29 bits. The C0GMR register corresponds to slots 0 to 13, the C0LMAR register corresponds to slot 14, and the C0LMBR register corresponds to slot 15. The masking function becomes valid to 11 bits or 29 bits of a received ID according to the value in the corresponding slot of the C0IDR register upon acceptance filtering operation. When the masking function is employed, it is possible to receive a certain range of IDs.

Figure 19.17 shows the Correspondence of Mask Registers and Slots, Figure 19.18 shows the Acceptance Function.

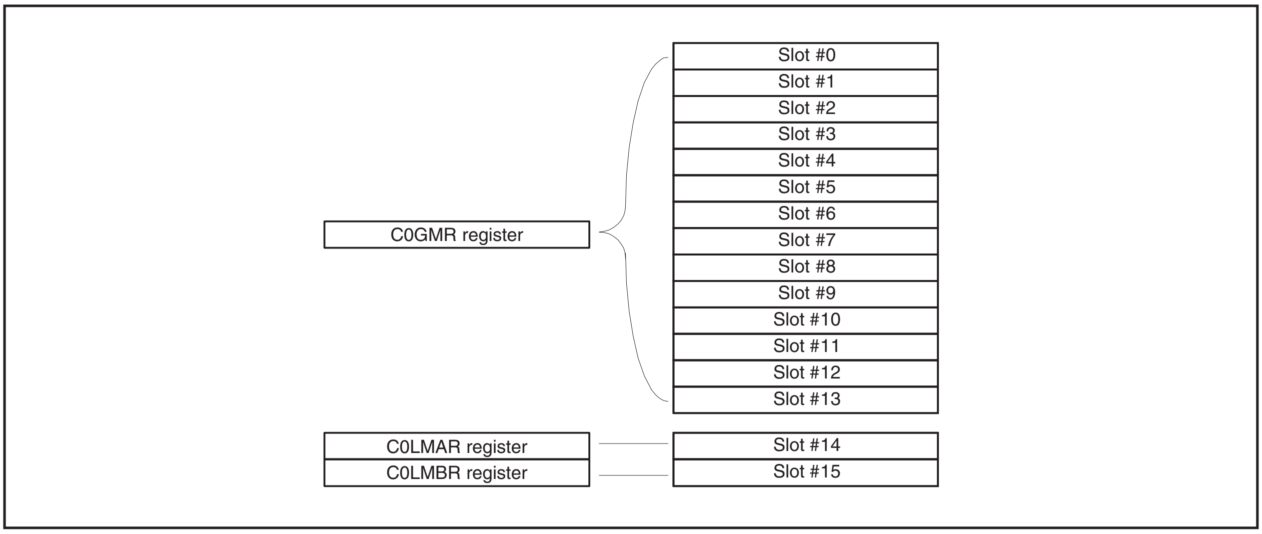


Figure 19.17 Correspondence of Mask Registers to Slots

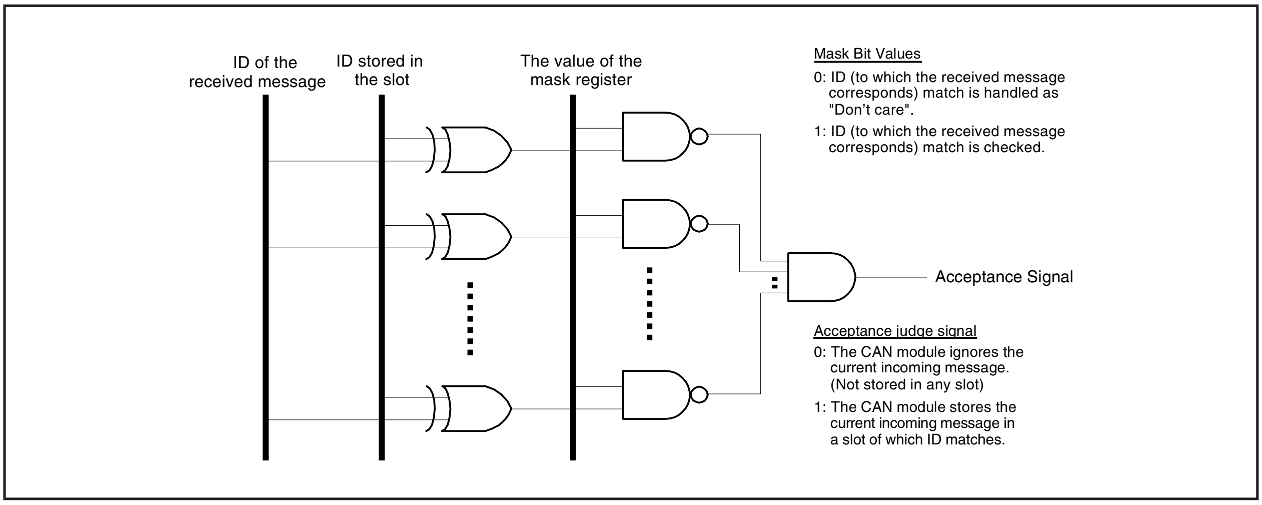

ID of the received message
ID stored in the slot
The value of the mask register
Mask Bit Values
0: ID (to which the received message corresponds) match is handled as "Don't care".
1: ID (to which the received message corresponds) match is checked.
Acceptance Signal
Acceptance judge signal
0: The CAN module ignores the current incoming message. (Not stored in any slot)
1: The CAN module stores the current incoming message in a slot of which ID matches.

Figure 19.18 Acceptance Function

When using the acceptance function, note the following points.

(1) When one ID is defined in two slots, the one with a smaller number alone is valid.

(2) When it is configured that slots 14 and 15 receive all IDs with Basic CAN mode, slots 14 and 15 receive all IDs which are not stored into slots 0 to 13.

19.10 Acceptance Filter Support Unit (ASU)

The acceptance filter support unit has a function to judge valid/invalid of a received ID through table search. The IDs to receive are registered in the data table; a received ID is stored in the C0AFS register, and table search is performed with a decoded received ID. The acceptance filter support unit can be used for the IDs of the standard frame only.

The acceptance filter support unit is valid in the following cases.

- When the ID to receive cannot be masked by the acceptance filter.
  (Example) IDs to receive: 078h, 087h, 111h
- When there are too many IDs to receive; it would take too much time to filter them by software.

Figure 19.19 shows the Write/Read of C0AFS Register in Word Access.

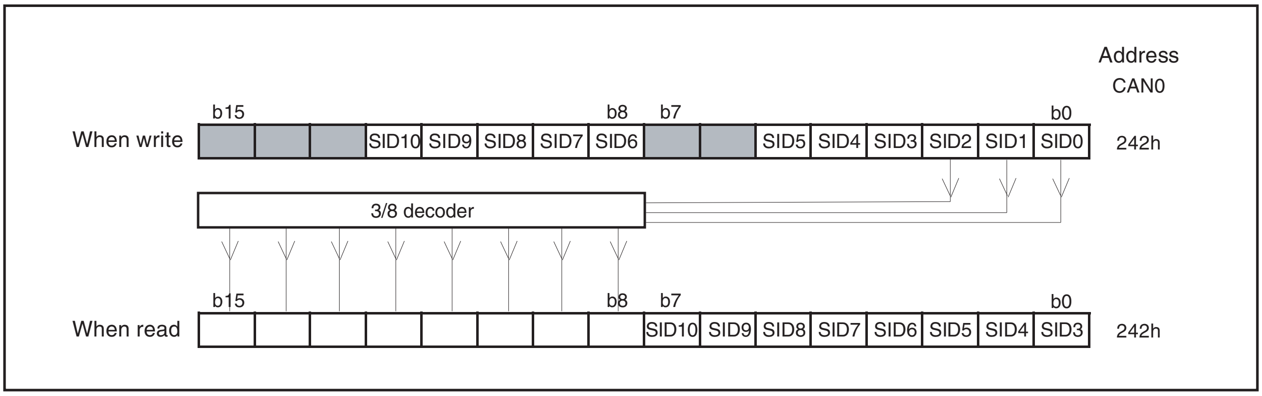


Figure 19.19 Write/read of C0AFS Register in Word Access

19.11 Basic CAN Mode

When the BasicCAN bit in the C0CTLR register is set to 1 (Basic CAN mode enabled), slots 14 and 15 correspond to Basic CAN mode. In normal operation mode, each slot can handle only one type message at a time, either a data frame or a remote frame by setting C0MCTLj regisrer (j = 0 to 15). However, in Basic CAN mode, slots 14 and 15 can receive both types of message at the same time.

When slots 14 and 15 are defined as reception slots in Basic CAN mode, received messages are stored in slots 14 and 15 alternately.

Which type of message has been received can be checked by the RemActive bit in the C0MCTLj register. Figure 19.20 shows the Slots 14 and 15 Operation in Basic CAN Mode.

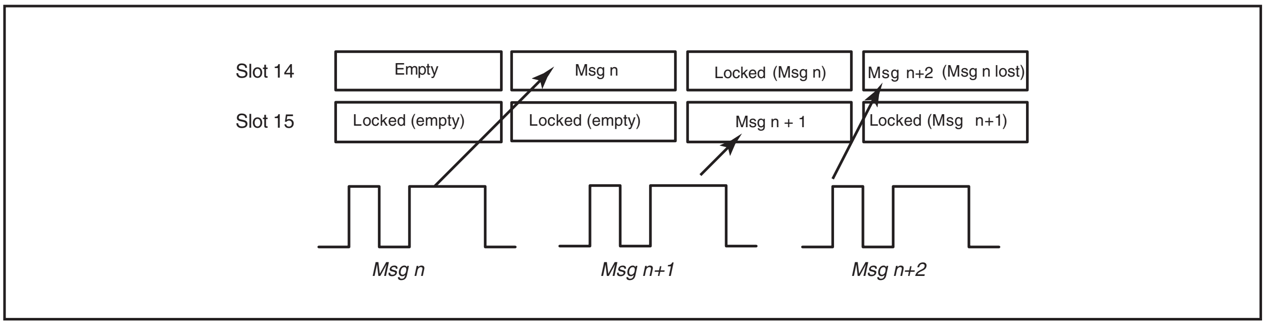


Figure 19.20 Slots 14 and 15 Operation in Basic CAN Mode

When using Basic CAN mode, note the following points.

(1) Setting of Basic CAN mode has to be done in CAN reset/initialization mode.

(2) Select the same ID for slots 14 and 15. Also, setting of registers C0LMAR and C0LMBR has to be the same.

(3) Define slots 14 and 15 as reception slot only.

(4) There is no protection available against message overwrite. A message can be overwritten by a new message.

(5) Slots 0 to 13 can be used in the same way as in normal CAN operation mode.

19.12 Return from Bus Off Function

When the protocol controller enters bus off state, it is possible to make it forced return from bus off state by setting the RetBusOff bit in the C0CTLR register to 1 (force return from bus off). At this time, the error state changes from bus off state to error active state. If the RetBusOff bit is set to 1, registers C0RECR and C0TECR are initialized and the State_BusOff bit in the C0STR register is set to 0 (CAN module is not in error bus off state). However, registers of the CAN module such as C0CONR register and the content of each slot are not initialized.

19.13 Time Stamp Counter and Time Stamp Function

When the C0TSR register is read, the value of the time stamp counter at the moment is read. The period of the time stamp counter reference clock is the same as that of 1 bit time that is configured by the C0CONR register. The time stamp counter functions as a free run counter.

The 1 bit time period can be divided by 1 (undivided), 2, 4 or 8 to produce the time stamp counter reference clock. Use the TSPreScale bit in the C0CTLR register to select the divide-by-n value.

The time stamp counter is equipped with a register that captures the counter value when the protocol controller regards it as a successful reception. The captured value is stored when a time stamp value is stored in a reception slot.

19.14 Listen-Only Mode

When the RXOnly bit in the C0CTLR register is set to 1, the module enters Listen-only mode.

In Listen-only mode, no transmission, such as data frames, error frames, and ACK response, is performed to bus.

When Listen-only mode is selected, do not request the transmission.

19.15 Reception and Transmission

Table 19.3 lists the CAN Reception and Transmission Mode Configuration.

Table 19.3 CAN Reception and Transmission Mode Configuration

TrmReq	RecReq	Remote	RspLock	Communication Mode of Slot
0	0	-	-	Communication environment configuration mode: configure the communication mode of the slot.
0	1	0	0	Configured as a reception slot for a data frame.
1	0	1	0	Configured as a transmission slot for a remote frame. (At this time the RemActive = 1.) After completion of transmission, this functions as a reception slot for a data frame. (At this time the RemActive = 0.) However, when an ID that matches on the CAN bus is detected before remote frame transmission, this immediately functions as a reception slot for a data frame.
1	0	0	0	Configured as a transmission slot for a data frame.
0	1	1	1/0	Configured as a reception slot for a remote frame. (At this time the RemActive = 1.) After completion of reception, this functions as a transmission slot for a data frame. (At this time the RemActive = 0.) However, transmission does not start as long as RspLock bit remains 1; thus no automatic response. Response (transmission) starts when the RspLock bit is set to 0.

TrmReq, RecReq, Remote, RspLock, RemActive, RspLock: Bits in C0MCTLj register (j = 0 to 15)

When configuring a slot as a reception slot, note the following points.

(1) Before configuring a slot as a reception slot, be sure to set the C0MCTLj register to 00h.

(2) A received message is stored in a slot that matches the condition first according to the result of reception mode configuration and acceptance filtering operation. Upon deciding in which slot to store, the smaller the number of the slot is, the higher priority it has.

(3) In normal CAN operation mode, when a CAN module transmits a message of which ID matches, the CAN module never receives the transmitted data. In loop back mode, however, the CAN module receives back the transmitted data. In this case, the module does not return ACK.

When configuring a slot as a transmission slot, note the following points.

(1) Before configuring a slot as a transmission slot, be sure to set the C0MCTLj registers to 00h.

(2) Set the TrmReq bit in the C0MCTLj register to 0 (not transmission slot) before rewriting a transmission slot.

(3) A transmission slot should not be rewritten when the TrmActive bit in the C0MCTLj register is 1 (transmitting).
If it is rewritten, an indeterminate data will be transmitted.

19.15.1 Reception

Figure 19.21 shows the Timing of Receive Data Frame Sequence. Figure 19.21 shows the behavior of the module when receiving two consecutive CAN messages, that fit into the slot of the shown C0MCTLj register (j = 0 to 15) and leads to losing/overwriting of the first message.

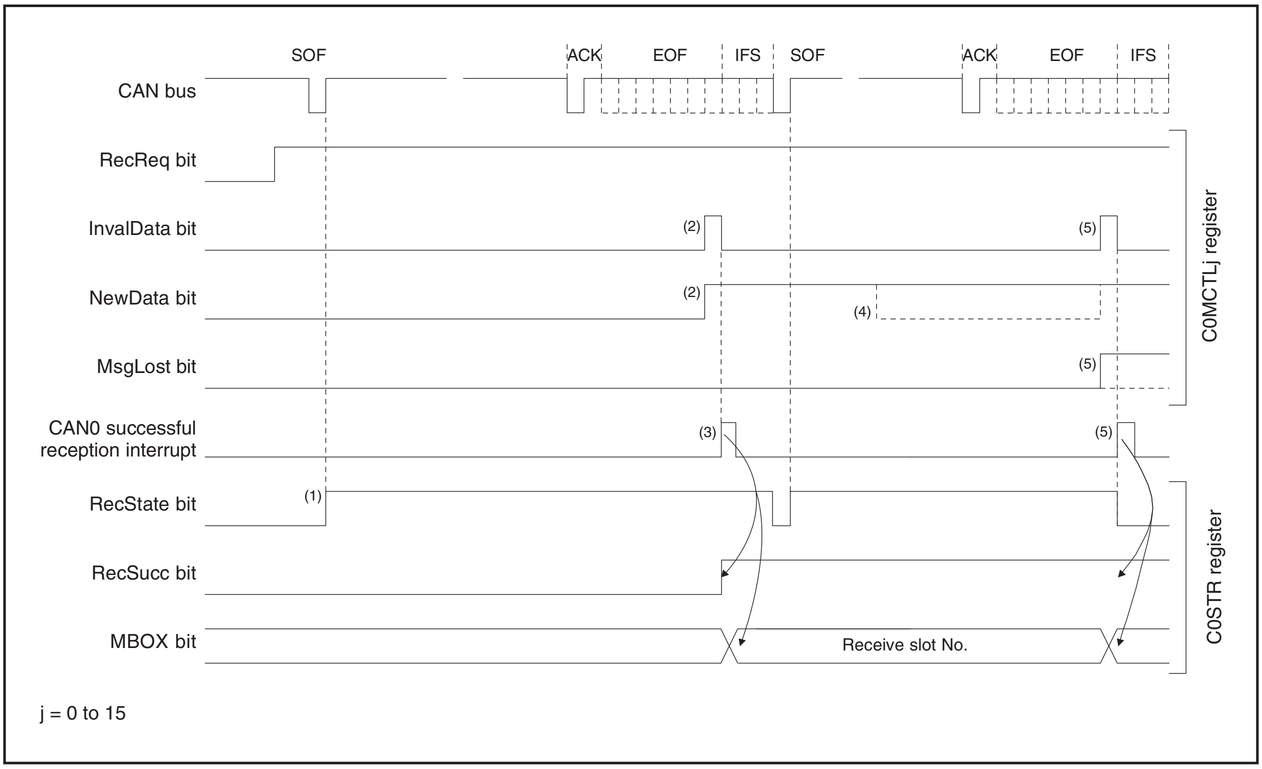


Figure 19.21 Timing of Receive Data Frame Sequence

(1) On monitoring a SOF on the CAN bus the RecState bit in the C0STR register becomes 1 (CAN module is receiver) immediately, given the module has no transmission pending.

(2) After successful reception of the message, the NewData bit in the C0MCTLj register of the receiving slot becomes 1 (stored new data in slot). The InvalData bit in the C0MCTLj register becomes 1 (message is being updated) at the same time and the InvalData bit becomes 0 (message is valid) again after the complete message was transferred to the slot.

(3) When the interrupt enable bit in the C0ICR register of the receiving slot = 1 (interrupt enabled), the CAN0 successful reception interrupt request is generated and the MBOX bit in the C0STR register is changed. It shows the slot number where the message was stored and the RecSucc bit in the C0STR register is active.

(4) Read the message out of the slot after setting the New Data bit to 0 (the content of the slot is read or still under processing by the CPU) by a program.

(5) When next CAN message is received before the NewData bit is set to 0 by a program or a receive request to a slot is canceled, the MsgLost bit in the C0MCTLj register is set to 1 (message has been overwritten). The new received message is transferred to the slot. Generating of an interrupt request and change of the C0STR register are same as in 3).

19.15.2 Transmission

Figure 19.22 shows the Timing of Transmit Sequence.

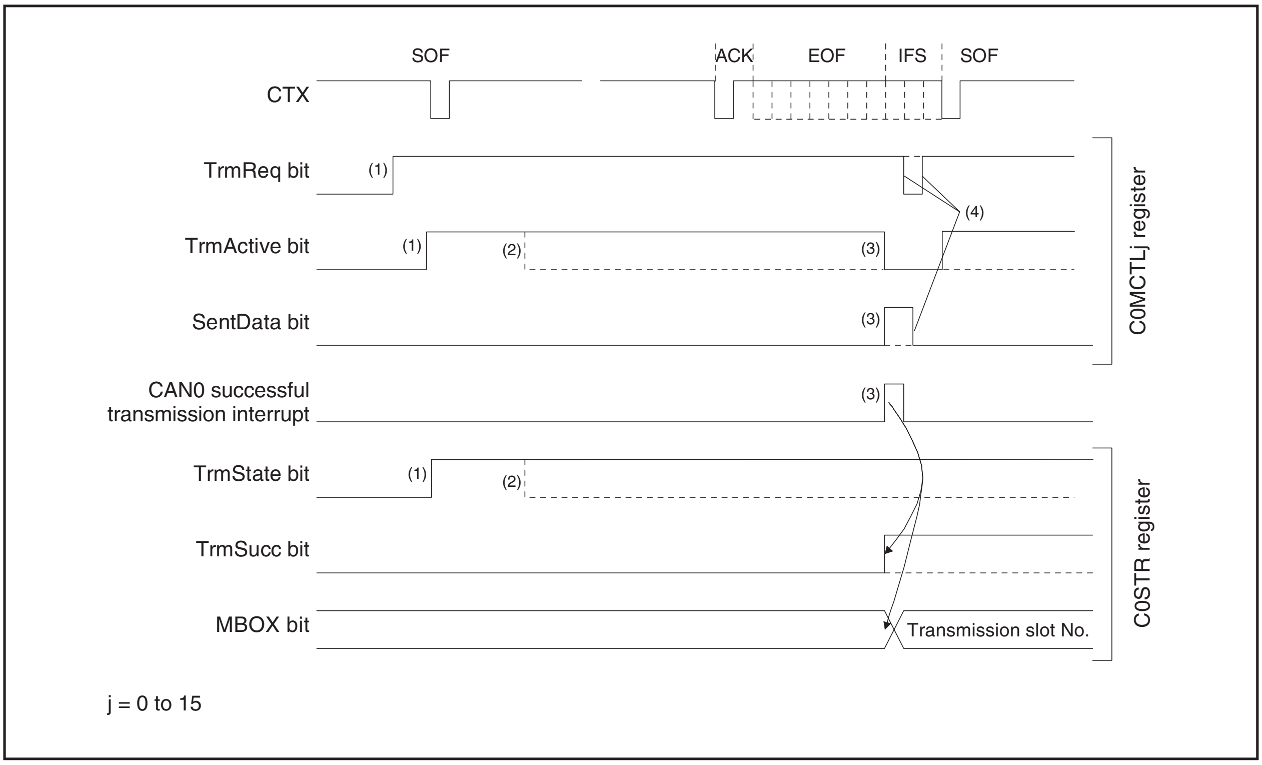


Figure 19.22 Timing of Transmit Sequence

(1) If the TrmReq bit in the C0MCTLj register (j = 0 to 15) is set to 1 (transmission slot) in the bus idle state, the TrmActive bit in the C0MCTLj register and the TrmState bit in the C0STR register are set to 1 (transmitting/transmitter), and CAN module starts the transmission.

(2) If the arbitration is lost after the CAN module starts the transmission, bits TrmActive and TrmState are set to 0.

(3) If the transmission has been successful without lost in arbitration, the SentData bit in the C0MCTLj register is set to 1 (transmission is successfully completed) and TrmActive bit is set to 0 (waiting for bus idle or completion of arbitration). And when the interrupt enable bits in the C0ICR register = 1 (Interrupt enabled), CAN0 successful transmission interrupt request is generated and the MBOX (the slot number which transmitted the message) and TrmSucc bit in the C0STR register are changed.

(4) When starting the next transmission, set bits SentData and TrmReq to 0. And set the TrmReq bit to 1 after checking that bits SentData and TrmReq are set to 0.

19.16 CAN Interrupt

The CAN module provides the following CAN interrupts.

- CAN0 successful reception interrupt
- CAN0 successful transmission interrupt
- CAN0 error interrupt: Error passive state
  Error bus off state
  Bus error (this feature can be disabled separately)
- CAN0 wake-up interrupt

When the CPU detects the CAN0 successful reception/transmission interrupt request, the MBOX bit in the C0STR register must be read to determine which slot has generated the interrupt request.

20. Programmable I/O Ports

The programmable input/output ports (hereafter referred to simply as I/O ports) consist of 87 lines P0 to P10 in the 100-pin version and consist of 113 lines P0 to P14 in the 128-pin version. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high every 4 lines. P8_5 is an input-only port and does not have a pull-up resistor. Port P8_5 shares the pin with $\overline{\text{NMI}}$, so that the $\overline{\text{NMI}}$ input level can be read from the P8_5 bit in the P8 register.

Table 20.1 lists the I/O ports Pin Number of Each Package. Figures 20.1 to 20.5 show the I/O ports. Figure 20.6 shows the I/O pins.

Each pin functions as an I/O port, a peripheral function input/output pin or a bus control pin.

For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, SI/O4 output or D/A converter output pin, set the direction bit for that pin to 0 (input mode). Any pin used as an output pin for peripheral functions other than the SI/O4 and D/A converter is directed for output no matter how the corresponding direction bit is set.

When using any pin as a bus control pin, refer to **7.2 Bus Control**.

Table 20.1 I/O Ports Pin Number of Each Package

	128-pin Version	100-pin Version
I/O ports	P0_0 to P0_7 P1_0 to P1_7 P2_0 to P2_7 P3_0 to P3_7 P4_0 to P4_7 P5_0 to P5_7 P6_0 to P6_7 P7_0 to P7_7 P8_0 to P8_4, P8_6, P8_7 (P8_5 is an input port) P9_0 to P9_7 P10_0 to P10_7 P11_0 to P11_7 P12_0 to P12_7 P13_0 to P13_7 P14_0, P14_1	P0_0 to P0_7 P1_0 to P1_7 P2_0 to P2_7 P3_0 to P3_7 P4_0 to P4_7 P5_0 to P5_7 P6_0 to P6_7 P7_0 to P7_7 P8_0 to P8_4, P8_6, P8_7 (P8_5 is an input port) P9_0 to P9_7 P10_0 to P10_7
Total	113 pins	87 pins

20.1 PDi Register (100-pin Version: i = 0 to 10, 128-pin Version: i = 0 to 13)

Figure 20.7 shows the PDi Register.

This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

During memory expansion and microprocessor modes, the PDi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}/\overline{\text{WR}}$, $\overline{\text{WRH}}/\overline{\text{BHE}}$, ALE, $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, and BCLK) cannot be modified.

No direction register bit for P8_5 is available.

20.2 Pi Register (100-pin Version: i = 0 to 10, 128-pin Version: i = 0 to 13), PC14 Register

Figure 20.8 shows the Pi Register.

Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the input/output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.

For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

During memory expansion and microprocessor modes, the Pi registers for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}/\overline{\text{WR}}$, $\overline{\text{WRH}}/\overline{\text{BHE}}$, ALE, $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, and BCLK) cannot be modified.

About the port P14 (128-pin version), Figure 20.8 shows the PC14 Register.

20.3 PURj Register (100-pin Version: j = 0 to 2, 128-pin Version: j = 0 to 3)

Figures 20.9 and 20.10 show the PURj Register.

The PURj register bits can be used to select whether or not to pull the corresponding port high in 4-bit unit. The port selected to be pulled high has a pull-up resistor connected to it when the direction bit is set for input mode.

However, the pull-up control register has no effect on P0 to P3, P4_0 to P4_3, and P5 during memory expansion and microprocessor modes. Although the register contents can be modified, no pull-up resistors are connected.

When using the ports P11 to P14, set the PUR37 bit in the PUR3 register to 1 (P11 to P14 are usable).

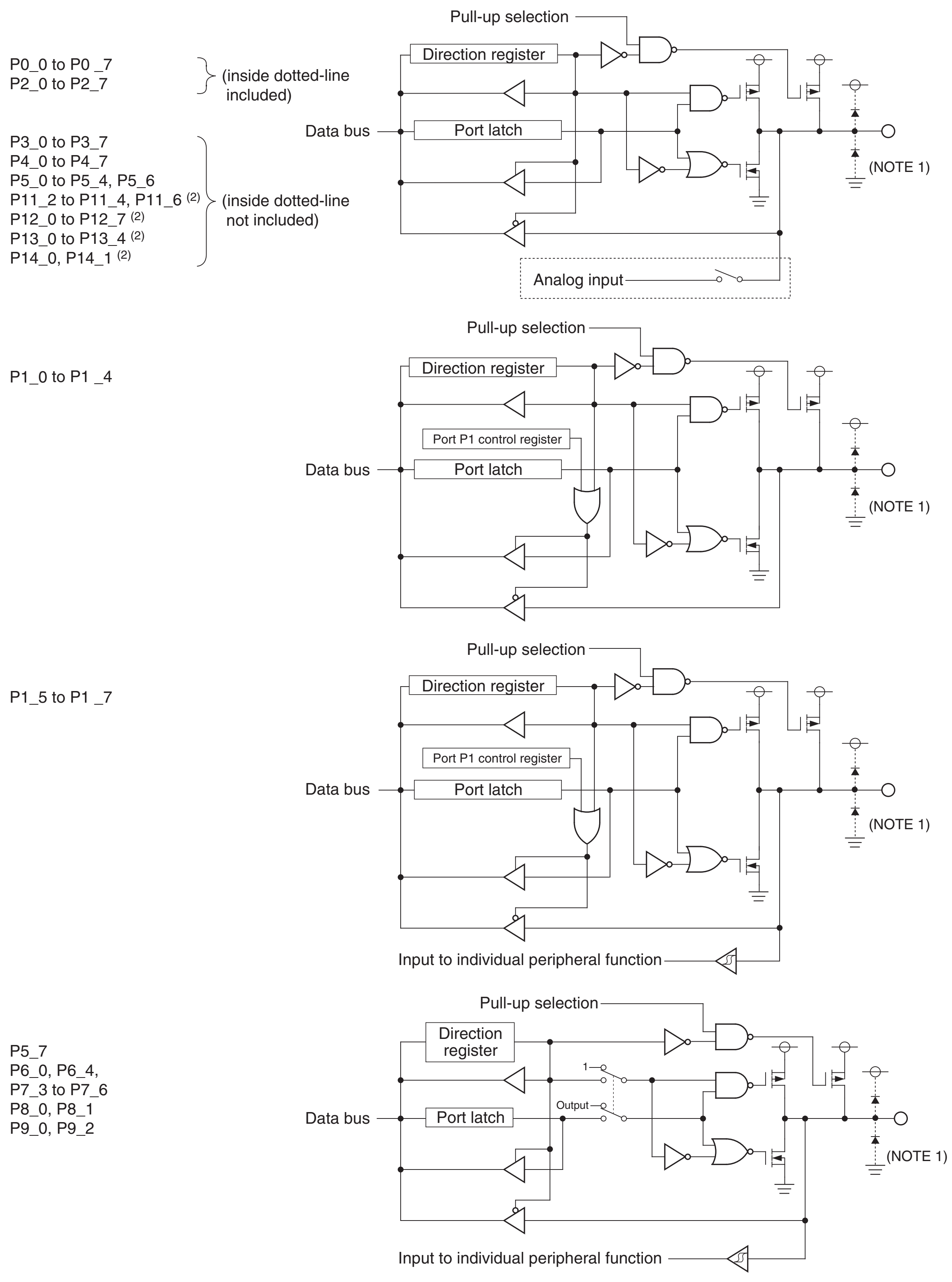
20.4 PCR Register

Figure 20.11 shows the PCR Register.

When the P1 register is read after setting the PCR0 bit in the PCR register to 1, the corresponding port latch can be read no matter how the PD1 register is set.

Table 20.2 lists the Unassigned Pin Handling in Single-chip Mode and Table 20.3 lists the Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode.

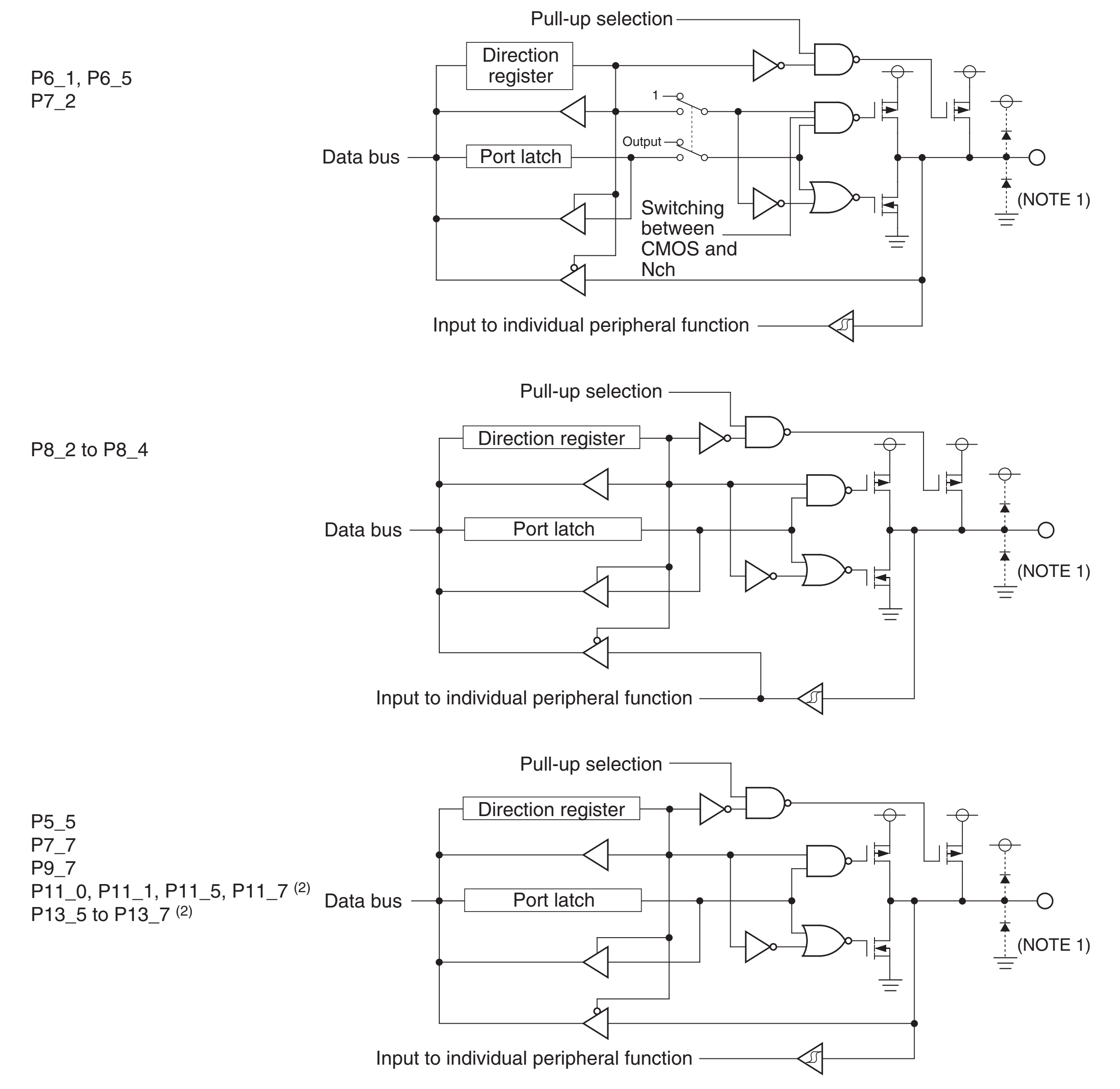
Figure 20.12 shows the Unassigned Pin Handling.



NOTES:

1. ┈┈◀┈┈ Symbolizes a parasitic diode.
   Make sure the input voltage on each port will not exceed VCC.
2. P11 to P14 are only in the 128-pin version.

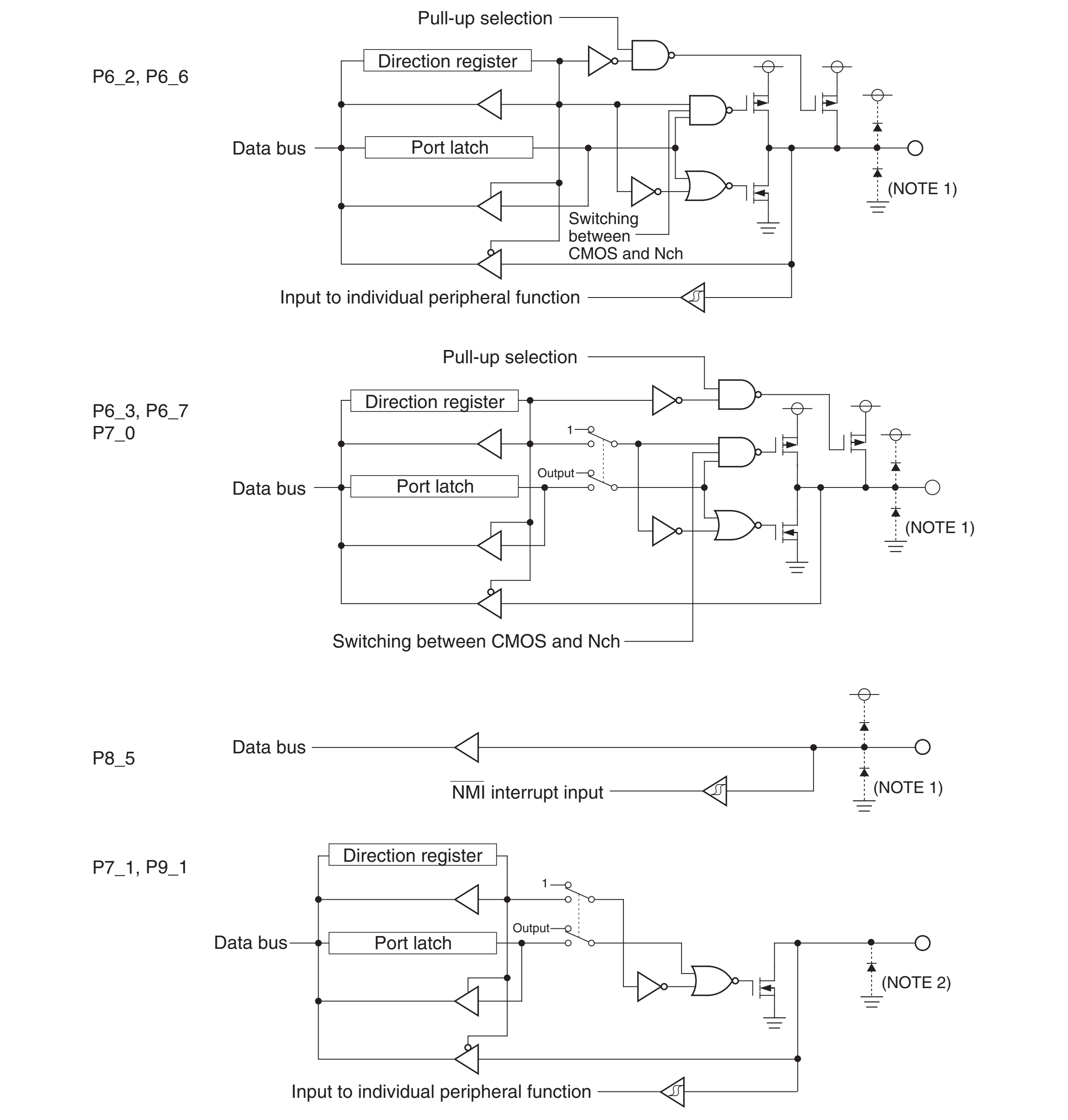
Figure 20.1 I/O Ports (1)



NOTES:

1. Symbolizes a parasitic diode.
   Make sure the input voltage on each port will not exceed VCC.
2. P11 and P13 are only in the 128-pin version.

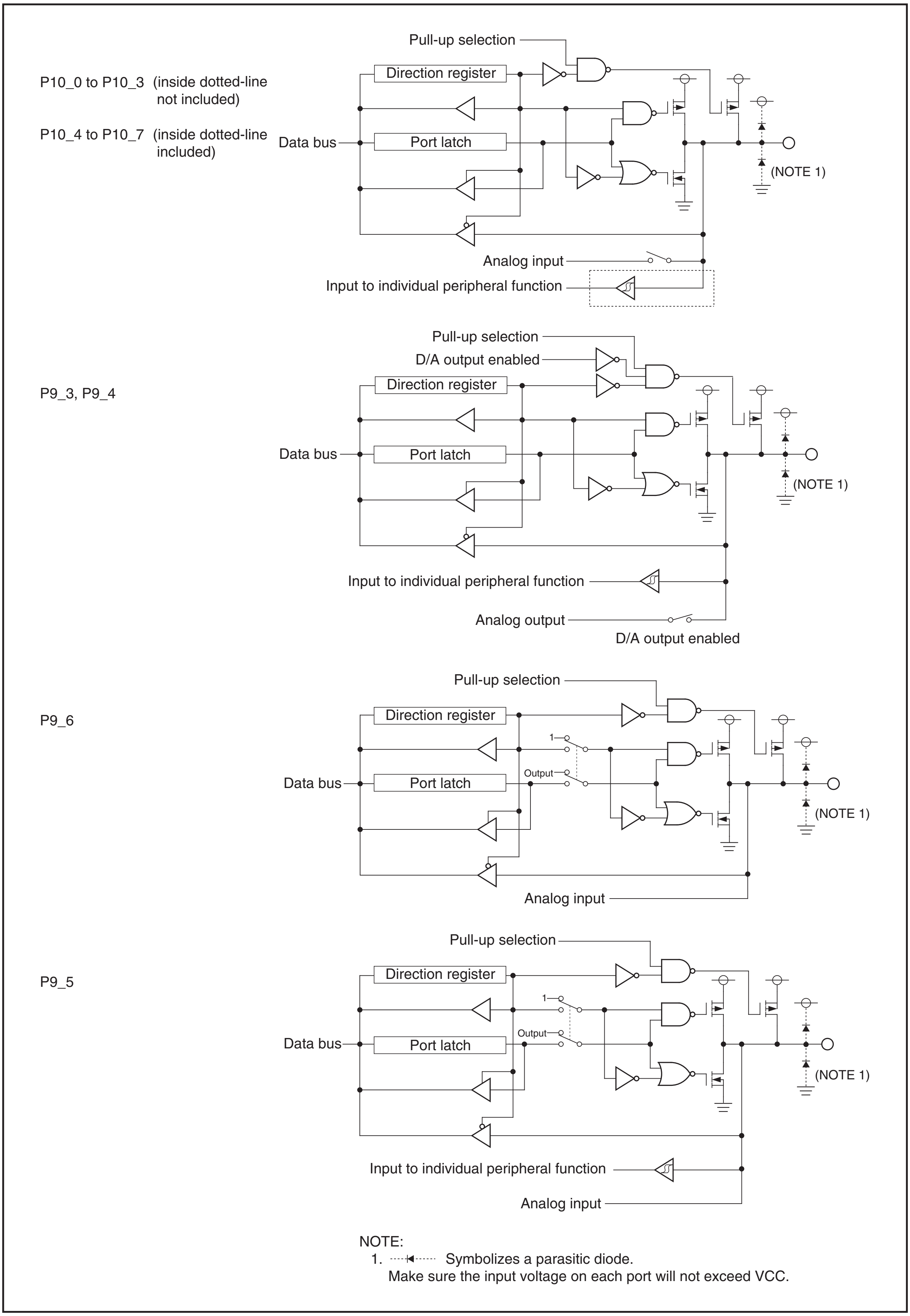
Figure 20.2 I/O Ports (2)



NOTES:

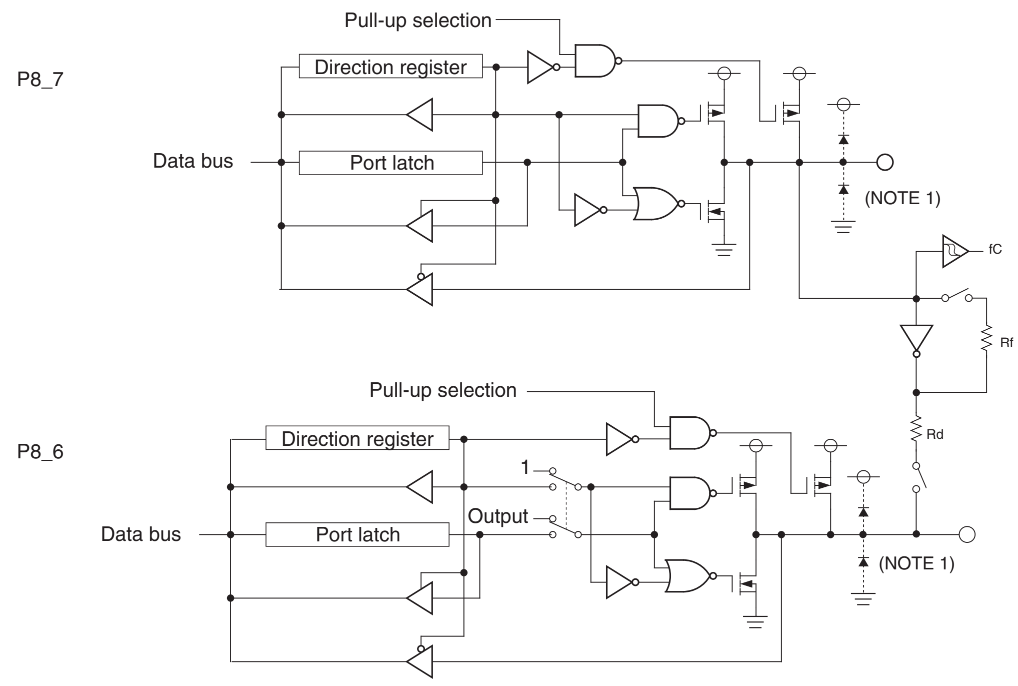
1. ┄┄|◀┄┄ Symbolizes a parasitic diode.
   Make sure the input voltage on each port will not exceed VCC.
2. ┄┄|◀┄┄ Symbolizes a parasitic diode.

Figure 20.3 I/O Ports (3)



NOTE:
1. ····◄···· Symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed VCC.

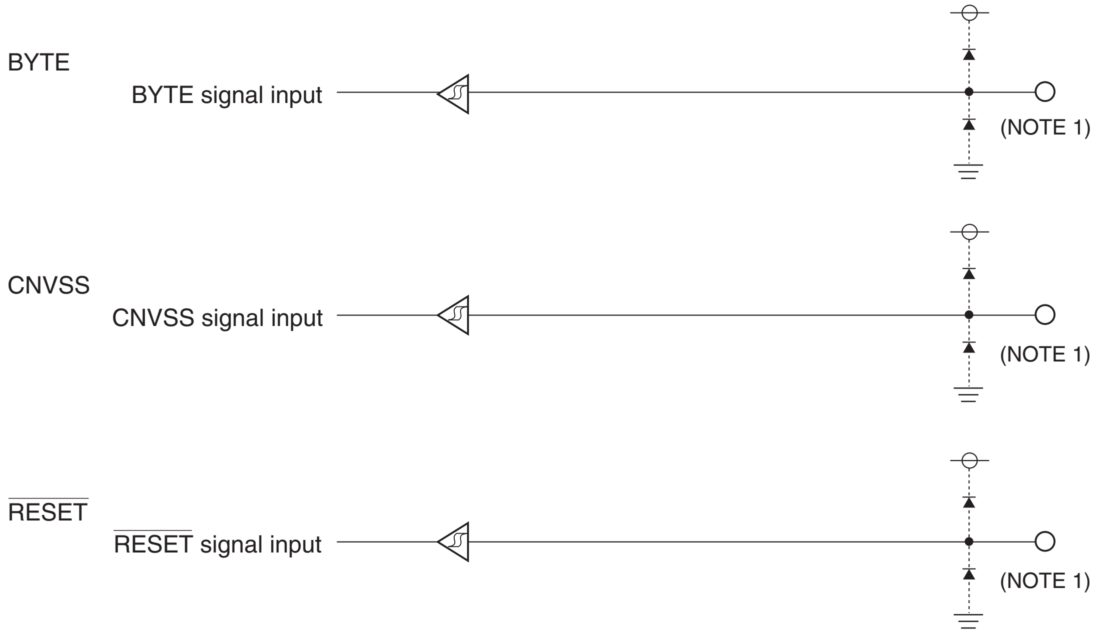
Figure 20.4 I/O Ports (4)



NOTE:

1. ·····▸|····· Symbolizes a parasitic diode.
   Make sure the input voltage on each port will not exceed VCC.

Figure 20.5 I/O Ports (5)

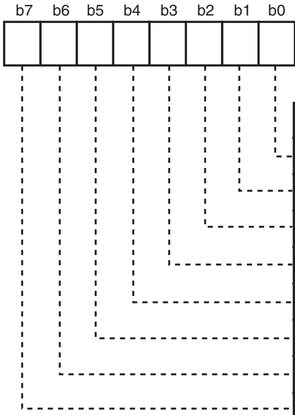


NOTE:

1. ·····▸|····· Symbolizes a parasitic diode.
   Make sure the input voltage on each port will not exceed VCC.

Figure 20.6 I/O Pins

Port Pi Direction Register (i = 0 to 7, 9 to 13) (1) (2) (3)



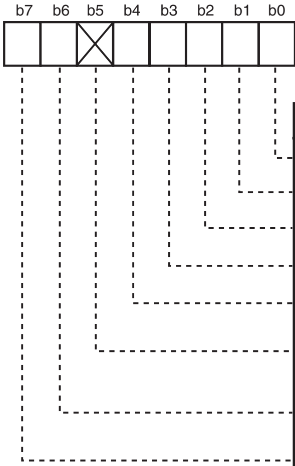
Symbol	Address	After Reset
PD0 to PD3	03E2h, 03E3h, 03E6h, 03E7h	00h
PD4 to PD7	03EAh, 03EBh, 03EEh, 03EFh	00h
PD9 to PD12 (4)	03F3h, 03F6h, 03F7h, 03FAh	00h
PD13 (4)	03FBh	00h

Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi_0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PDi_1	Port Pi_1 direction bit		RW
PDi_2	Port Pi_2 direction bit		RW
PDi_3	Port Pi_3 direction bit		RW
PDi_4	Port Pi_4 direction bit		RW
PDi_5	Port Pi_5 direction bit		RW
PDi_6	Port Pi_6 direction bit		RW
PDi_7	Port Pi_7 direction bit		RW

NOTES:

1. Make sure registers PD7 and PD9 are written to by the next instruction after setting the PRC2 bit in the PRCR register to 1 (write enabled).
2. During memory expansion and microprocessor modes, the PDi register for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}/\overline{\text{WR}}$, $\overline{\text{WRH}}/\overline{\text{BHE}}$, ALE, $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, and BCLK) cannot be modified.
3. When using the ports P11 to P13, set the PU37 bit in the PUR3 register to 1 (usable).
4. Registers PD11 to PD13 are only in the 128-pin version.

Port P8 Direction Register



Symbol	Address	After Reset
PD8	03F2h	00X00000b

Bit Symbol	Bit Name	Function	RW
PD8_0	Port P8_0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_1	Port P8_1 direction bit		RW
PD8_2	Port P8_2 direction bit		RW
PD8_3	Port P8_3 direction bit		RW
PD8_4	Port P8_4 direction bit		RW
- (b5)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		-
PD8_6	Port P8_6 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD8_7	Port P8_7 direction bit		RW

Figure 20.7 Registers PD0 to PD13

Port Pi Register (i = 0 to 7, 9 to 13) (1) (2) (3)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
P0 to P3	03E0h, 03E1h, 03E4h, 03E5h	Undefined
P4 to P7	03E8h, 03E9h, 03ECh, 03EDh	Undefined
P9 to P12 (4)	03F1h, 03F4h, 03F5h, 03F8h	Undefined
P13 (4)	03F9h	Undefined

Bit Symbol	Bit Name	Function	RW
Pi_0	Port Pi_0 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register. 0 : "L" level 1 : "H" level	RW
Pi_1	Port Pi_1 bit		RW
Pi_2	Port Pi_2 bit		RW
Pi_3	Port Pi_3 bit		RW
Pi_4	Port Pi_4 bit		RW
Pi_5	Port Pi_5 bit		RW
Pi_6	Port Pi_6 bit		RW
Pi_7	Port Pi_7 bit		RW

NOTES:
1. Since P7_1 and P9_1 are N channel open-drain ports, the data is high-impedance.
2. During memory expansion and microprocessor modes, the Pi register for the pins functioning as bus control pins (A0 to A19, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{RD}}$, $\overline{\text{WRL}}/\overline{\text{WR}}$, $\overline{\text{WRH}}/\overline{\text{BHE}}$, ALE, $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, and BCLK) cannot be modified.
3. When using the ports P11 to P13, set the PU37 bit in the PUR3 register to 1 (usable). If this bit is set to 0 (unusable), registers P11 to P13 are set to 00h.
4. Registers P11 to P13 are only in the 128-pin version.

Port P8 Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
P8	03F0h	Undefined

Bit symbol	Bit name	Function	RW
P8_0	Port P8 _0 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register. (except for P8_5.) 0 : "L" level 1 : "H" level	RW
P8_1	Port P8 _1 bit		RW
Pi8_2	Port P8 _2 bit		RW
P8_3	Port P8 _3 bit		RW
P8_4	Port P8 _4 bit		RW
P8_5	Port P8 _5 bit		RO
P8_6	Port P8 _6 bit		RW
P8_7	Port P8 _7 bit		RW

Port P14 Control Register (128-pin version) (1)

b7 b6 b5 b4 b3 b2 b1 b0

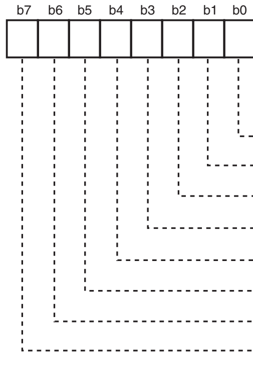
Symbol	Address	After Reset
PC14	03DEh	XX00XXXXb

Bit Symbol	Bit Name	Function	RW
P140	Port P14_0 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register. 0 : "L" level 1 : "H" level	RW
P141	Port P14_1 bit		RW
- (b3-b2)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–
PD140	Port P14_0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PD141	Port P14_1 direction bit		RW
- (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined.		–

NOTE:
1. When using the port P14, set the PU37 bit in the PUR3 register to 1 (usable).

Figure 20.8 Registers P0 to P13, and PC14

Pull-up Control Register 0 (1)



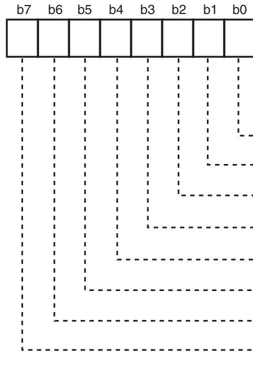
Symbol	Address	After Reset
PUR0	03FCh	00h

Bit Symbol	Bit Name	Function	RW
PU00	P0_0 to P0_3 pull-up	0 : Not pulled high 1 : Pulled high (2)	RW
PU01	P0_4 to P0_7 pull-up		RW
PU02	P1_0 to P1_3 pull-up		RW
PU03	P1_4 to P1_7 pull-up		RW
PU04	P2_0 to P2_3 pull-up		RW
PU05	P2_4 to P2_7 pull-up		RW
PU06	P3_0 to P3_3 pull-up		RW
PU07	P3_4 to P3_7 pull-up		RW

NOTES:

1. During memory expansion and microprocessor modes, the pins are not pulled high although their corresponding register contents can be modified.
2. The pin for which this bit is 1 (pulled high) and the direction bit is 0 (input mode) is pulled high.

Pull-up Control Register 1



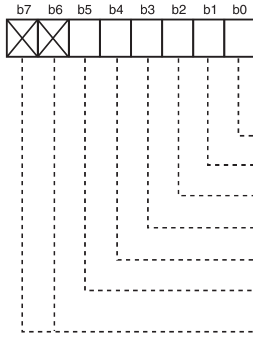
Symbol	Address	After Reset (1)
PUR1	03FDh	00000000b 00000010b

Bit Symbol	Bit Name	Function	RW
PU10	P4_0 to P4_3 pull-up (2)	0 : Not pulled high 1 : Pulled high (5)	RW
PU11	P4_4 to P4_7 pull-up (3)		RW
PU12	P5_0 to P5_3 pull-up (2)		RW
PU13	P5_4 to P5_7 pull-up (2)		RW
PU14	P6_0 to P6_3 pull-up		RW
PU15	P6_4 to P6_7 pull-up		RW
PU16	P7_0, P7_2, and P7_3 pull-up (4)		RW
PU17	P7_4 to P7_7 pull-up		RW

NOTES:

1. The values after hardware reset is as follows:
   00000000b when input on CNVSS pin is "L".
   00000010b when input on CNVSS pin is "H".
   The values after software reset, watchdog timer reset and oscillation stop detection reset are as follows:
   00000000b when bits PM 01 to PM00 in the PM0 register are 00b (single-chip mode).
   00000010b when bits PM 01 to PM00 are 01b (memory expansion mode) or 11b (microprocessor mode).
2. During memory expansion and microprocessor modes, the pins are not pulled high although their corresponding register contents can be modified.
3. If bits PM01 to PM00 are set to 01b (memory expansion mode) or 11b (microprocessor mode) in a program during single-chip mode, the PU11 bit becomes 1.
4. The P7_1 pin does not have pull-up.
5. The pin for which this bit is 1 (pulled high) and the direction bit is 0 (input mode) is pulled high.

Pull-up Control Register 2



Symbol	Address	After Reset
PUR2	03FEh	00h

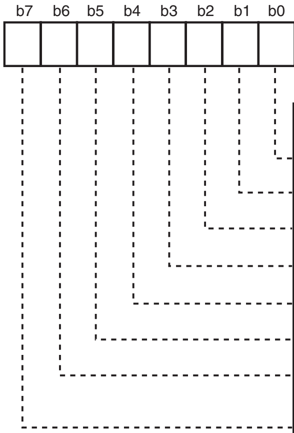
Bit Symbol	Bit Name	Function	RW
PU20	P8_0 to P8_3 pull-up	0 : Not pulled high 1 : Pulled high (3)	RW
PU21	P8_4, P8_6, and P8_7 pull-up (1)		RW
PU22	P9_0, P9_2, and P9_3 pull-up (2)		RW
PU23	P9_4 to P9_7 pull-up		RW
PU24	P10_0 to P10_3 pull-up		RW
PU25	P10_4 to P10_7 pull-up		RW
- (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		-

NOTES:

1. The P8_5 pin does not have pull-up.
2. The P9_1 pin does not have pull-up.
3. The pin for which this bit is 1 (pulled high) and the direction bit is 0 (input mode) is pulled high.

Figure 20.9 Registers PUR0, PUR1, and PUR2

Pull-up Control Register 3 (128-pin version)



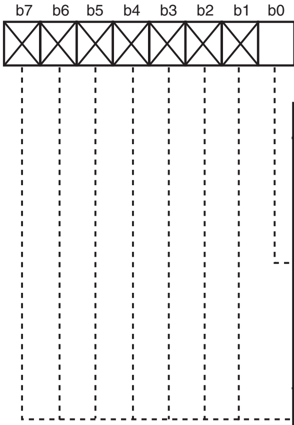
Symbol	Address	After Reset
PUR3	03DFh	00h

Bit Symbol	Bit Name	Function	RW
PU30	P11_0 to P11_3 pull-up	0 : Not pulled high 1 : Pulled high (1)	RW
PU31	P11_4 to P11_7 pull-up		RW
PU32	P12_0 to P12_3 pull-up		RW
PU33	P12_4 to P12_7 pull-up		RW
PU34	P13_0 to P13_3 pull-up		RW
PU35	P13_4 to P13_7 pull-up		RW
PU36	P14_0, P14_1 pull-up		RW
PU37	P11 to P14 enabling bit	0 : Unusable (2) 1 : Usable	RW

NOTES:
1. The pin for which this bit is 1 (pulled high) and the direction bit is 0 (input mode) is pulled high.
2. If the PU37 bit is set to 0 (unusable), registers P11 to P14 are set to 00h.

Figure 20.10 PUR3 Register

Port Control Register



Symbol	Address	After Reset
PCR	03FFh	00h

Bit Symbol	Bit Name	Function	RW
PCR0	Port P1 control bit	Operation performed when the P1 register is read 0 : When the port is set for input, the input levels of P1_0 to P1_7 pins are read. When set for output, the port latch is read. 1 : The port latch is read regardless of whether the port is set for input or output.	RW
- (b7-b1)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		-

Figure 20.11 PCR Register

Table 20.2 Unassigned Pin Handling in Single-chip Mode

Pin Name	Connection
Ports P0 to P7, P8_0 to P8_4, P8_6, P8_7, P9 to P14 (5)	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. (1) (2) (3)
XOUT (4)	Open
$\overline{\text{NMI}}$(P8_5)	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

NOTES:

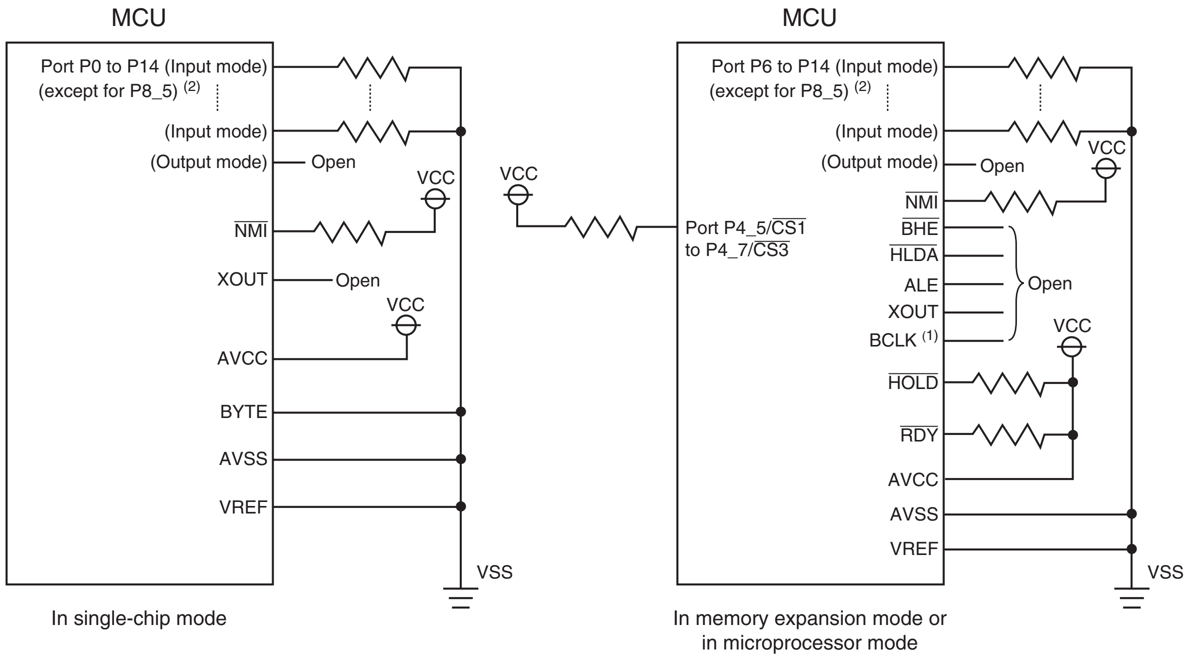
1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes undefined, causing the power supply current to increase while the port remains in input mode.
   Furthermore, by considering a possibility that the contents of the direction registers may change due to noise or program runaway caused by noise, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
2. Make sure the unused pins are processed with the shortest possible wiring from the MCU pins (2 cm or less).
3. When the ports P7_1 and P9_1 are set for output mode, make sure a low-level signal is output from the pins.
   The ports P7_1 and P9_1 are N-channel open-drain outputs.
4. With external clock input to XIN pin.
5. The ports P11 to P14 are only in the 128-pin version. When not using all of pins P11 to P14 may be left open by setting the PU37 bit in the PUR3 register to 0 (P11 to P14 unusable), without causing any problem.

Table 20.3 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode

Pin Name	Connection
Ports P6, P7, P8_0 to P8_4, P8_6, P8_7, P9 to P14 (7)	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open. (1) (2) (3) (4)
P4_5/$\overline{\text{CS1}}$ to P4_7/$\overline{\text{CS3}}$	Connect to VCC via a resistor (pulled high) by setting the corresponding direction bit in the PD4 register for $\overline{\text{CSi}}$ (i = 1 to 3) to 0 (input mode) and the $\overline{\text{CSi}}$ bit in the CSR register to 0 (chip select disabled).
$\overline{\text{BHE}}$, ALE, $\overline{\text{HLDA}}$, XOUT (5), BCLK (6)	Open
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, $\overline{\text{NMI}}$(P8_5)	Connect via resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

NOTES:

1. When setting the port for output mode and leave it open, be aware that the port remains in input mode until it is switched to output mode in a program after reset. For this reason, the voltage level on the pin becomes indeterminate, causing the power supply current to increase while the port remains in input mode.
   Furthermore, by considering a possibility that the contents of the direction registers may change due to noise or program runaway caused by noise, it is recommended that the contents of the direction registers be periodically reset in software, for the increased reliability of the program.
2. Make sure the unused pins are processed with the shortest possible wiring from the MCU pins (2 cm or less).
3. If the CNVSS pin has the VSS level applied to it, these pins are set for input ports until the processor mode is switched over in a program after reset. For this reason, the voltage levels on these pins become indeterminate, causing the power supply current to increase while they remain set for input ports.
4. When the ports P7_1 and P9_1 are set for output mode, make sure a low-level signal is output from the pins.
   The ports P7_1 and P9_1 are N-channel open-drain outputs.
5. With external clock input to XIN pin.
6. If the PM07 bit in the PM0 register is set to 1 (BCLK not output), connect this pin to VCC via a resistor (pulled high).
7. The ports P11 to P14 are only in the 128-pin version. When not using all of pins P11 to P14 may be left open by setting the PU37 bit in the PUR3 register to 0 (P11 to P14 unusable), without causing any problem.



NOTES:

1. If the PM07 bit in the PM0 register is set to 1 (BCLK not output), connect this pin to VCC via a resistor (pulled high).
2. The ports P11 to P14 are only in the 128-pin version. When not using all of pins P11 to p14 may be left open by setting the PU37 bit in the PUR3 register to 0 (P11 to P14 unusable), without causing any problem.

Figure 20.12 Unassigned Pins Handling

21. Flash Memory Version

Aside from the on-chip flash memory, the flash memory version MCU has the same functions as the masked ROM version.

In the flash memory version, the flash memory can perform in four rewrite mode: CPU rewrite mode, standard serial I/O mode, parallel I/O mode, and CAN I/O mode.

Table 21.1 lists the Flash Memory Version Specifications. See **Tables 1.1 and 1.2 Functions and Specifications**, for the items not listed in Table 21.1. Table 21.2 shows the Flash Memory Rewrite Modes Overview.

Table 21.1 Flash Memory Version Specifications

Item		Specifications
Flash memory rewrite mode		4 modes (CPU rewrite, standard serial I/O, parallel I/O, CAN I/O)
Erase block	User ROM area	See **Figure 21.1 Flash Memory Block Diagram**
	Boot ROM area	1 block (4 Kbytes) [1]
Program method		In units of word, in units of byte [2]
Erase method		Collective erase, block erase
Program and erase control method		Program and erase controlled by software command
Protect method		Lock bit protects each block
Number of commands		8 commands
Programming and erasure endurance [3]		100 times
ROM code protection		Parallel I/O, standard serial I/O, and CAN I/O modes are supported.

NOTES:

1. The boot ROM area contains standard serial I/O mode and CAN I/O mode rewrite control program which is stored in it when shipped from the factory. This area can only be rewritten in parallel I/O mode.
2. Can be programmed in byte units in only parallel I/O mode.
3. Definition of programming and erasure endurance
   The programming and erasure endurance is defined to be per-block erasure endurance. For example, assume a case where a 4K-byte block A is programmed in 2,048 operations by writing one word at a time and erased thereafter.
   In this case, the block is reckoned as having been programmed and erased once.
   If a product is 100 times of programming and erasure endurance, each block in it can be erased up to 100 times.

Table 21.2 Flash Memory Rewrite Modes Overview

Flash Memory Rewrite Mode	CPU Rewrite Mode [1]	Standard Serial I/O Mode	Parallel I/O Mode	CAN I/O Mode
Function	The user ROM area is rewritten when the CPU executes software commands. EW0 mode: Rewrite in areas other than flash memory [2] EW1 mode: Can be rewritten in the flash memory	The user ROM area is rewritten using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART [3]	The boot ROM and user ROM areas are rewritten using a dedicated parallel programmer.	The user ROM area is rewritten busing a dedicated CAN programmer.
Areas which can be rewritten	User ROM area	User ROM area	User ROM area Boot ROM area	User ROM area
Operating mode	Single-chip mode Memory expansion mode (EW0 mode) Boot mode (EW0 mode)	Boot mode	Parallel I/O mode	Boot mode
ROM programmer	None	Serial programmer	Parallel programmer	CAN programmer

NOTES:

1. The PM13 bit remains set to 1 while the FMR01 bit in the FMR0 register = 1 (CPU rewrite mode enabled). The PM13 bit is reverted to its original value by setting the FMR01 bit to 0 (CPU rewrite mode disabled). However, if the PM13 bit is changed during CPU rewrite mode, its changed value is not reflected until after the FMR01 bit is set to 0.
2. When in CPU rewrite mode, bits PM10 and PM13 in the PM1 register are set to 1. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1.
3. When using standard serial I/O mode 2, make sure a main clock input oscillation frequency is set to 5 MHz, 10 MHz, or 16 MHz.

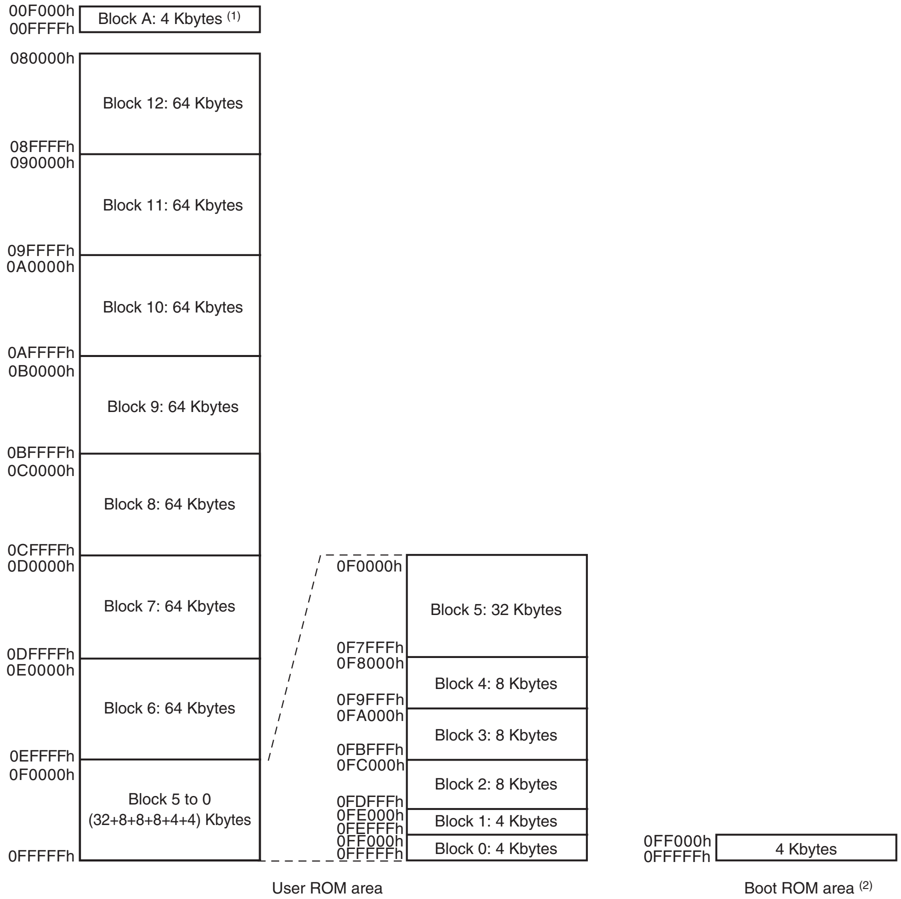
21.1 Memory Map

The flash memory contains the user ROM area and the boot ROM area. The user ROM area has space to store the MCU operating program in single-chip mode or memory expansion mode and a separate 4-Kbyte space as the block A.

Figure 21.1 shows the Flash Memory Block Diagram.

The user ROM area is divided into several blocks, each of which can be protected (locked) against programming or erasure. The user ROM area can be rewritten in CPU rewrite, standard serial I/O mode, parallel I/O mode, and CAN I/O mode. Block A is enabled for use by setting the PM10 bit in the PM1 register to 1 (block A enabled. $\overline{CS2}$ area at addresses 10000h to 26FFFh).

The boot ROM area is located at the same addresses as the user ROM area. It can only be rewritten in parallel I/O mode (refer to **21.1.1 Boot Mode**). A program in the boot ROM area is executed after a hardware reset occurs while an “H ” signal is applied to pins CNVSS and P5_0 and an “L” signal is applied to the P5_5 pin (refer to **21.1.1 Boot Mode**). A program in the user ROM area is executed after a hardware reset occurs while an “L” signal is applied to the CNVSS pin. However, the boot ROM area cannot be read.



* Shown here is a block diagram during single-chip mode.

NOTES:

1. Block A can be made usable by setting the PM10 bit in the PM1 register to 1 (block A enabled, addresses 10000h to 26FFFh for $\overline{CS2}$ area).
   Block A cannot be erased by the erase all unlocked block command. Use the block erase command to erase it.
2. The boot ROM area can only be rewritten in parallel I/O mode.
3. To specify a block, use an even address in that block.

Figure 21.1 Flash Memory Block Diagram

21.1.1 Boot Mode

The MCU enters boot mode when a hardware reset occurs while an “H ” signal is applied to pins CNVSS and P5_0 and an “L ” signal is applied to the P5_5 pin. A program in the boot ROM area is executed. In boot mode, the FMR05 bit in the FMR0 register selects access to the boot ROM area or the user ROM area. The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment. The boot ROM area can be rewritten in parallel I/O mode only. If given rewrite control program using erase-write mode (EW0 mode) is written in the boot ROM area, the flash memory can be rewritten according to the system implemented.

21.2 Functions to Prevent Flash Memory from Rewriting

The flash memory has the ROM code protect function for parallel I/O mode and the ID code check function for standard serial I/O mode and CAN I/O mode to prevent the flash memory from reading or rewriting.

21.2.1 ROM Code Protect Function

The ROM code protect function inhibits the flash memory from being read or rewritten during parallel I/O mode. Figure 21.2 shows the ROMCP Register. The ROMCP register is located in the user ROM area. The ROM code protect function is enabled when the ROMCR bits are set to other than 11b. In this case, set the bit 5 to bit 0 to 111111b.

When exiting ROM code protect, erase the block including the ROMCP register by CPU rewrite mode, standard serial I/O mode, or CAN I/O mode.

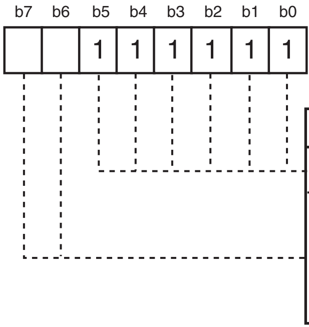
21.2.2 ID Code Check Function

Use the ID code check function in standard serial I/O mode and CAN I/O mode. The ID code sent from the serial programmer is compared with the ID code written in the flash memory for a match. If the ID codes do not match, commands sent from the serial programmer are not accepted. However, if the four bytes of the reset vector are FFFFFFFFh, ID codes are not compared, allowing all commands to be accepted.

The ID codes are 7-byte data stored consecutively, starting with the first byte, into addresses 0FFFDFh, 0FFFE3h, 0FFFEBh, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. The flash memory must have a program with the ID codes set in these addresses.

Figure 21.3 shows the Addresses for ID Code Stored.

ROM Code Protect Control Address [5]

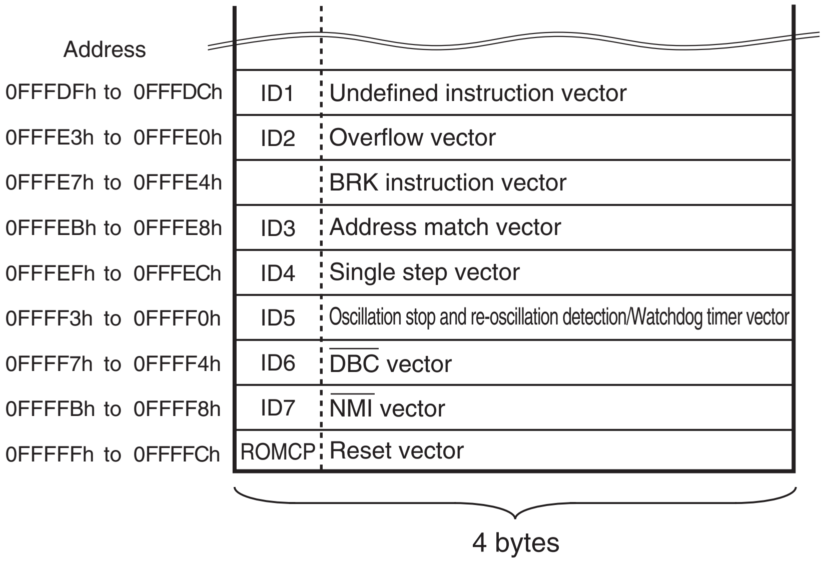


Symbol	Address	Value when Shipped
ROMCP	0FFFFFh	FFh [1]

Bit Symbol	Bit Name	Function	RW
(b5-b0)	Reserved bits	Set to 1	RW
ROMCP1	ROM code protect level 1 set bit [1] [2] [3] [4]	b7 b6 0 0 : ROM code protection active 0 1 : ROM code protection active 1 0 : ROM code protection active 1 1 : ROM code protection inactive	RW RW

NOTES:
1. The ROMCP address is set to FFh when a block, including the ROMCP address, is erased.
2. When the ROM code protection is active by the ROMCP1 bit setting, the flash memory is protected against reading or rewriting in parallel I/O mode.
3. Set bits 5 to 0 to 111111b when the ROMCP1 bit is set to a value other than 11b. If bits 5 to 0 are set to values other than 111111b, the ROM code protection may not become active by setting the ROMCP1 bit to a value other than 11b.
4. To make the ROM code protection inactive, erase a block including the ROMCP address in CPU rewrite mode, standard serial I/O mode, or CAN I/O mode.
5. When a value of the ROMCP address is 00h or FFh, the ROM code protect function is disabled.

Figure 21.2 ROMCP Register



Address		
0FFFDFh to 0FFFDCh	ID1	Undefined instruction vector
0FFFE3h to 0FFFE0h	ID2	Overflow vector
0FFFE7h to 0FFFE4h		BRK instruction vector
0FFFEBh to 0FFFE8h	ID3	Address match vector
0FFFEFh to 0FFFECh	ID4	Single step vector
0FFFF3h to 0FFFF0h	ID5	Oscillation stop and re-oscillation detection/Watchdog timer vector
0FFFF7h to 0FFFF4h	ID6	$\overline{\text{DBC}}$ vector
0FFFFBh to 0FFFF8h	ID7	$\overline{\text{NMI}}$ vector
0FFFFFh to 0FFFFCh	ROMCP	Reset vector

Figure 21.3 Address for ID Code Stored

21.3 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. The user ROM area can be rewritten with the MCU is mounted on a board without using a parallel, serial or CAN programmer.
In CPU rewrite mode, only the user ROM area shown in Figure 21.1 can be rewritten. The boot ROM area cannot be rewritten. Program and the block erase command are executed only in the user ROM area.
Erase-write 0 (EW0) mode and erase-write 1 (EW1) mode are provided as CPU rewrite mode.
Table 21.3 lists the differences between EW0 and EW1 Modes.

Table 21.3 EW0 Mode and EW1 Mode

Item	EW0 Mode	EW1 Mode
Operating mode	• Single-chip mode • Memory expansion mode • Boot mode	Single-chip mode
Space where rewrite control program can be placed	• User ROM area • Boot ROM area	User ROM area
Space where rewrite control program can be executed	The rewrite control program must be transferred to any space other than the flash memory (e.g., RAM) before being executed (2)	The rewrite control program can be executed in the user ROM area
Space which can be rewritten	User ROM area	User ROM area However, this excludes blocks with the rewrite control program
Software command restriction	None	• Program and block erase commands cannot be executed in a block having the rewrite control program. • Erase all unlocked block command cannot be executed when the lock bit in a block having the rewrite control program is set to 1 (unlocked) or when the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled). • Read status register command cannot be used.
Modes after program or erasing	Read status register mode	Read array mode
CPU status during auto-programming and auto-erasure	Operating	Maintains hold state (I/O ports maintains the state before the command was executed) (1)
Flash memory status detection	•Read bits FMR00, FMR06, and FMR07 in the FMR0 register by program •Execute the read status register command to read bits SR7, SR5, and SR4 in the status register	Read bits FMR00, FMR06, and FMR07 in the FMR0 register by program

NOTES:
1. Do not generate an interrupts (except $\overline{\text{NMI}}$ interrupt) and DMA transfer.
2. When in CPU rewrite mode, bits PM10 and PM13 in the PM1 register are set to 1. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1..

21.3.1 EW0 Mode

The MCU enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to 1 (CPU rewrite mode enabled) and is ready to accept commands. EW0 mode is selected by setting the FMR11 bit in the FMR1 register to 0. To set the FMR01 bit to 1, set to 1 after first writing 0.

The software commands control programming and erasing. The FMR0 register or the status register indicates whether a program or erase operation is completed as expected or not.

21.3.2 EW1 Mode

EW1 mode is selected by setting FMR11 bit to 1 (by writing 0 and then 1 in succession) after setting the FMR01 bit to 1 (by writing 0 and then 1 in succession). (Both bits must be set to 0 first before setting to 1.)

The FMR0 register indicates whether or not a program or erase operation has been completed as expected. The status register cannot be read in EW1 mode.

When an erase/program operation is initiated the CPU halts all program execution until the operation is completed or erase-suspend is requested.

21.3.3 Registers FMR0 and FMR1

Figure 21.4 shows Registers FMR0 and FMR1.

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
			0				

Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{\text{BY}}$ status flag	0 : Busy (being written or erased) (1) 1 : Ready	RO
FMR01	CPU rewrite mode select bit (2)	0 : CPU rewrite mode disabled 1 : CPU rewrite mode enabled	RW
FMR02	Lock bit disable select bit (3)	0: Lock bit enabled 1: Lock bit disabled	RW
FMSTP	Flash memory stop bit (4) (5)	0 Flash memory operation enabled 1: Flash memory operation stops (placed in low power dissipation mode, flash memory initialized)	RW
- (b4)	Reserved bit	Set to 0	RW
FMR05	User ROM area select bit (4) (Effective in only boot mode)	0 : Boot ROM area is accessed 1 : User ROM area is accessed	RW
FMR06	Program status flag (6)	0 : Terminated normally 1 : Terminated in error	RO
FMR07	Erase status flag (6)	0 : Terminated normally 1 : Terminated in error	RO

NOTES:

1. This status includes writing or reading with the lock bit program or read lock bit status command.
2. To set this bit to 1, write 0 and then 1 in succession. Make sure no interrupts or no DMA transfers will occur before writing 1 after writing 0.
   Write to this bit when the $\overline{\text{NMI}}$ pin is in the high state. Also, while in EW0 mode, write to this bit from a program in other than the flash memory.
   Enter read array mode and set this bit to 0.
3. To set this bit to 1, write 0 and then 1 in succession when the FMR01 bit = 1. Make sure no interrupts or no DMA transfers will occur before writing 1 after writing 0.
4. Write to this bit from a program in other than the flash memory.
5. Effective when the FMR01 bit = 1 (CPU rewrite mode). If the FMR01 bit = 0, although the FMSTP bit can be set to 1 by writing 1 in a program, the flash memory is neither placed in low power dissipation state nor initialized.
6. This bit is set to 0 by executing the clear status command.

Flash Memory Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
0		0	0				

Symbol	Address	After Reset
FMR1	01B5h	0X00XX0Xb

Bit Symbol	Bit Name	Function	RW
- (b0)	Reserved bit	When read, the content is undefined.	RO
FMR11	EW1 mode select bit (1)	0 : EW0 mode 1 : EW1 mode	RW
- (b3-b2)	Reserved bits	When read, the content is undefined.	RO
- (b5-b4)	Reserved bits	Set to 0	RW
FMR16	Lock bit status flag	0 : Lock 1 : Unlock	RO
- (b7)	Reserved bit	Set to 0	RW

NOTE:

1. To set this bit to 1, write 0 and then 1 in succession when the FMR01 bit in the FMR0 register = 1. Make sure no interrupts or no DMA transfers will occur before writing 1 after writing 0.
   Write to this bit when the $\overline{\text{NMI}}$ pin is in the high state.
   Both the FMR01 and FMR11 bits are set to 0 by setting the FMR01 bit to 0.

Figure 21.4 Registers FMR0 and FMR1

21.3.3.1 FMR00 Bit

This bit indicates the operating status of the flash memory. It is set to 0 while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to 1.

21.3.3.2 FMR01 Bit

The MCU can accept commands when the FMR01 bit is set to 1 (CPU rewrite mode). Set the FMR05 bit to 1 (user ROM area access) as well if in boot mode.

21.3.3.3 FMR02 Bit

The lock bit is disabled by setting the FMR02 bit to 1 (lock bit disabled). (Refer to **21.3.6 Data Protect Function**.) The lock bit is enabled by setting the FMR02 bit to 0 (lock bit enabled).

The FMR02 bit does not change the lock bit status but disables the lock bit function. If the block erase or erase all unlocked block command is executed when the FMR02 bit is set to 1, the lock bit status changes 0 (locked) to 1 (unlocked) after command execution is completed.

21.3.3.4 FMSTP Bit

The FMSTP bit resets the flash memory control circuits and minimizes power consumption in the flash memory. Access to the flash memory is disabled when the FMSTP bit is set to 1 (flash memory operation stops). Set the FMSTP bit by program in a space other than the flash memory.

Set the FMSTP bit to 1 if one of the followings occurs:

- A flash memory access error occurs while erasing or programming in EW0 mode (FMR00 bit does not switch back to 1 (ready))
- Low power dissipation mode or on-chip oscillator low power dissipation mode is entered

Use the following the procedure to change the FMSTP bit setting.

(1) Set the FMSTP bit to 1

(2) Set tps (the wait time to stabilize flash memory circuit)

(3) Set the FMSTP bit to 0

(4) Set tps (the wait time to stabilize flash memory circuit)

Figure 21.7 shows the Processing Before and After Low Power Dissipation Mode or On-chip Oscillator Low Power Dissipation Mode. Follow the procedure on this flow chart.

When entering stop or wait mode, the flash memory is automatically turned off. When exiting stop or wait mode, the flash memory is turned back on. The FMR0 register does not need to be set.

21.3.3.5 FMR05 Bit

This bit selects the boot ROM or user ROM area in boot mode. Set to 0 to access (read) the boot ROM area or to 1 (user ROM access) to access (read, write or erase) the user ROM area.

21.3.3.6 FMR06 Bit

This is a read-only bit indicating the status of an auto-program operation. The FMR06 bit is set to 1 when a program error occurs; otherwise, it is set to 0. Refer to **21.3.8 Full Status Check**.

21.3.3.7 FMR07 Bit

This is a read-only bit indicating the status of an auto-erase operation. The FMR07 bit is set to 1 when an erase error occurs; otherwise, it is set to 0. For details, refer to **21.3.8 Full Status Check**.

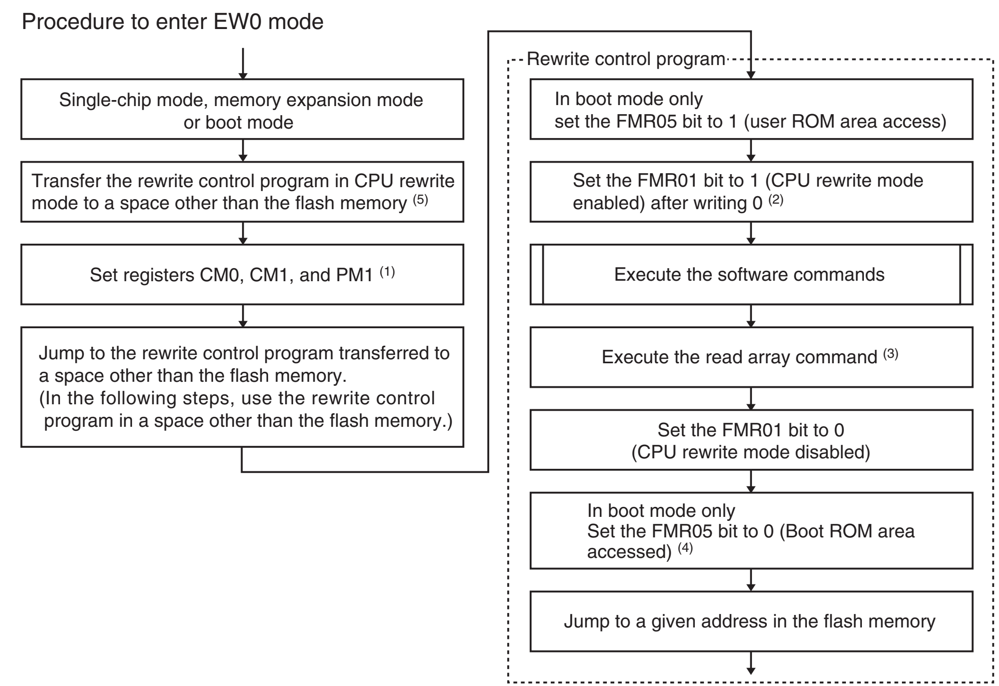
21.3.3.8 FMR11 Bit

EW0 mode is entered by setting the FMR11 bit to 0 (EW0 mode).
EW1 mode is entered by setting the FMR11 bit to 1 (EW1 mode).

21.3.3.9 FMR16 Bit

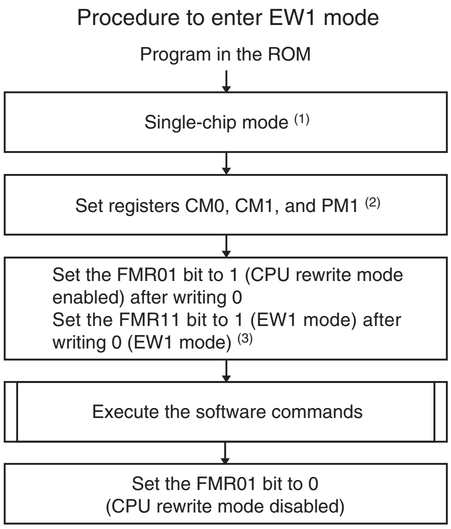
This is a read-only bit indicating the execution result of the read lock bit status command. When the block, where the read lock bit status command is executed, is locked, the FMR16 bit is set to 0.
When the block, where the read lock bit status command is executed, is unlocked, the FMR16 bit is set to 1.

Figure 21.5 shows the Setting and Resetting of EW0 Mode. Figure 21.6 show the Setting and Resetting of EW1 Mode.


Procedure to enter EW0 mode
Single-chip mode, memory expansion mode or boot mode
Transfer the rewrite control program in CPU rewrite mode to a space other than the flash memory (5)
Set registers CM0, CM1, and PM1 (1)
Jump to the rewrite control program transferred to a space other than the flash memory.
(In the following steps, use the rewrite control program in a space other than the flash memory.)
Rewrite control program
In boot mode only
set the FMR05 bit to 1 (user ROM area access)
Set the FMR01 bit to 1 (CPU rewrite mode enabled) after writing 0 (2)
Execute the software commands
Execute the read array command (3)
Set the FMR01 bit to 0
(CPU rewrite mode disabled)
In boot mode only
Set the FMR05 bit to 0 (Boot ROM area accessed) (4)
Jump to a given address in the flash memory

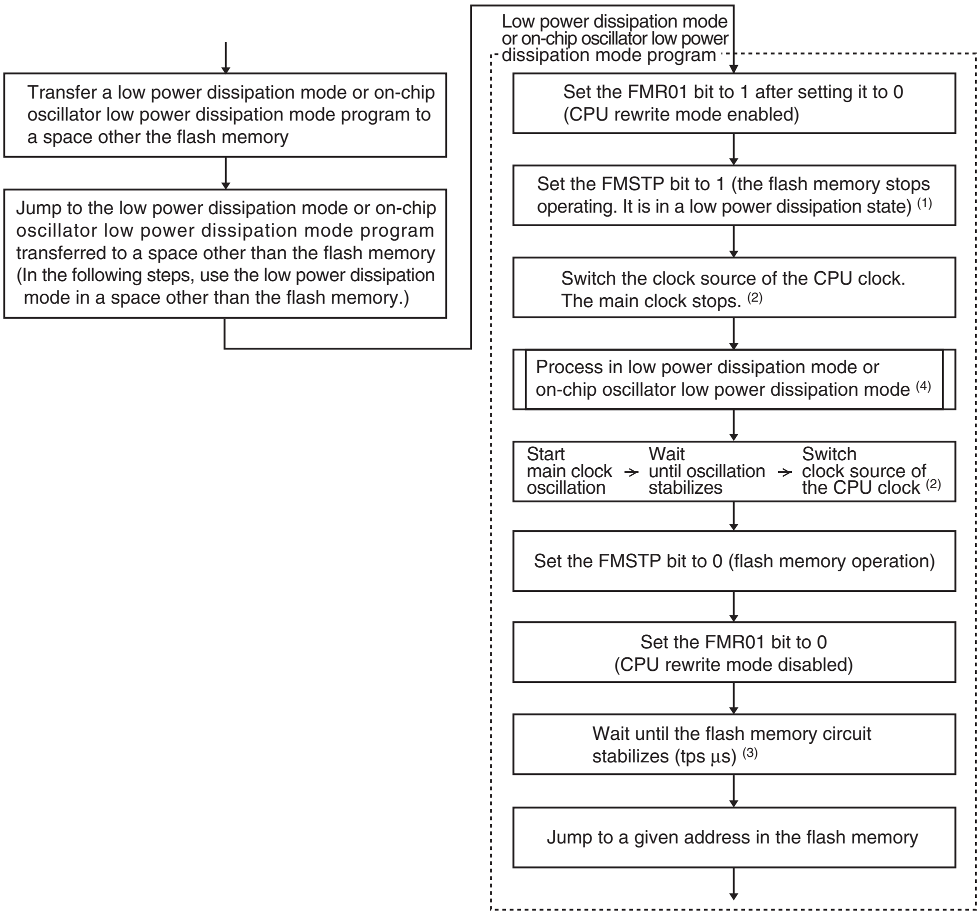
NOTES:
1. In CPU rewrite mode, set the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register to CPU clock frequency of 10 MHz or less. Set the PM17 bit in the PM1 register to 1 (with wait state).
2. Set the FMR01 bit to 1 immediately after setting it to 0. Do not generate an interrupts or DMA transfer between setting the bit to 0 and setting it to 1.
   Set the bit to 0 if setting to 0. Set this bit in a space other than the flash memory while the $\overline{\text{NMI}}$ pin is held "H".
3. Exit CPU rewrite mode after executing the read array command.
4. When CPU rewrite mode is exited while the FMR05 bit is set to 1, the user ROM area can be accessed.
5. When in CPU rewrite mode, bits PM10 and PM13 in the PM1 register are set to 1. The rewrite control program can only be executed in the internal RAM or in an external area that is enabled for use when the PM13 bit = 1.

Figure 21.5 Setting and Resetting of EW0 Mode


Procedure to enter EW1 mode
Program in the ROM
Single-chip mode (1)
Set registers CM0, CM1, and PM1 (2)
Set the FMR01 bit to 1 (CPU rewrite mode enabled) after writing 0
Set the FMR11 bit to 1 (EW1 mode) after writing 0 (EW1 mode) (3)
Execute the software commands
Set the FMR01 bit to 0
(CPU rewrite mode disabled)

NOTES:
1. In EW1 mode, do not enter the memory expansion mode or boot mode.
2. In CPU rewrite mode, set the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register to CPU clock frequency of 10 MHz or less. Set the PM17 bit in the PM1 register to 1 (with wait state).
3. Set the FMR01 bit to 1 immediately after setting it to 0. Do not generate an interrupt or a DMA transfer between setting the bit to 0 and setting it to 1.
   Set the FMR11 bit to 1 immediately after setting it to 0 while the FMR01 bit is set to 1.
   Do not generate an interrupt or a DMA transfer between setting the FMR11 bit to 0 and setting it to 1.
   Set bits FMR01 and FMR11 while "H" is applied to the $\overline{\text{NMI}}$ pin.

Figure 21.6 Setting and Resetting of EW1 Mode


Transfer a low power dissipation mode or on-chip oscillator low power dissipation mode program to a space other the flash memory
Jump to the low power dissipation mode or on-chip oscillator low power dissipation mode program transferred to a space other than the flash memory (In the following steps, use the low power dissipation mode in a space other than the flash memory.)
Low power dissipation mode or on-chip oscillator low power dissipation mode program
Set the FMR01 bit to 1 after setting it to 0 (CPU rewrite mode enabled)
Set the FMSTP bit to 1 (the flash memory stops operating. It is in a low power dissipation state) (1)
Switch the clock source of the CPU clock. The main clock stops. (2)
Process in low power dissipation mode or on-chip oscillator low power dissipation mode (4)
Start main clock oscillation
Wait until oscillation stabilizes
Switch clock source of the CPU clock (2)
Set the FMSTP bit to 0 (flash memory operation)
Set the FMR01 bit to 0 (CPU rewrite mode disabled)
Wait until the flash memory circuit stabilizes (tps µs) (3)
Jump to a given address in the flash memory

NOTES:

1. Set the FMSTP bit in the FMR0 register to 1 after setting the FMR01 bit in the FMR0 register to 1 (CPU rewrite mode enabled).
2. Wait until clock stabilizes to switch clock source of the CPU clock to the main clock or sub clock.
3. Add tps µs wait time by program. Do not access the flash memory during this wait time.
4. Before entering wait mode or stop mode, be sure to set the FMR01 bit to 0 (CPU rewrite mode disabled).

Figure 21.7 Processing Before and After Low Power Dissipation Mode or On-chip Oscillator Low Power Dissipation Mode

21.3.4 Notes on CPU Rewrite Mode

21.3.4.1 Operating Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), set the CM11 bit in the CM1 register to 0 (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register. Also, set the PM17 bit in the PM1 register to 1 (with wait state).

21.3.4.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

21.3.4.3 Interrupts (EW0 Mode)

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available since registers FMR0 and FMR1 are forcibly reset when either interrupt request is generated. Allocate the jump addresses for each interrupt service routines to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ or watchdog timer interrupt request is generated. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

21.3.4.4 Interrupts (EW1 Mode)

- Do not acknowledge any interrupts with vectors in the relocatable vector table or address match interrupt during auto-programming or auto-erasure.
- Do not use the watchdog timer interrupt.
- The $\overline{\text{NMI}}$ interrupt is available since registers FMR0 and FMR1 are forcibly reset when the interrupt request is generated. Allocate the jump address for the interrupt service routine to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ interrupt request is generated. Execute the rewrite program again after exiting the interrupt service routine.

21.3.4.5 How to Access

To set the FMR01, FMR02 or FMR11 bit to 1, write 1 after first setting the bit to 0. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to 0 and the instruction to set the bit to 1. Set the bit while an “H” signal is applied to the $\overline{\text{NMI}}$ pin.

21.3.4.6 Rewriting in User ROM Area (EW0 Mode)

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory cannot be rewritten because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area while in standard serial I/O mode, parallel I/O mode, or CAN I/O mode.

21.3.4.7 Rewriting in User ROM Area (EW1 Mode)

Avoid rewriting any block in which the rewrite control program is stored.

21.3.4.8 DMA Transfer

In EW1 mode, do not perform a DMA transfer while the FMR00 bit in the FMR0 register is set to 0 (auto-programming or auto-erasure).

21.3.4.9 Writing Command and Data

Write commands and data to even addresses in the user ROM area.

21.3.4.10 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

21.3.4.11 Stop Mode

When entering stop mode, execute the instruction which sets the CM10 bit to 1 (stop mode) after setting the FMR01 bit to 0 (CPU rewrite mode disabled) and disabling the DMA transfer.

21.3.4.12 Low Power Dissipation Mode and On-chip Oscillator Low Power Dissipation Mode

If the CM05 bit is set to 1 (main clock stopped), do not execute the following commands:

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program
- Read lock bit status

21.3.5 Software Commands

Software commands are described below. The command code and data must be read and written in 16-bit unit, to and from even addresses in the user ROM area. When writing command code, the high-order 8 bits (D15 to D8) are ignored.

Table 21.4 lists the Software Commands.

Table 21.4 Software Commands

Software Command	First Bus Cycle			Second Bus Cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read array	Write	×	xxFFh	-	-	-
Read status register	Write	×	xx70h	Read	×	SRD
Clear status register	Write	×	xx50h	-	-	-
Program	Write	WA	xx40h	Write	WA	WD
Block erase	Write	×	xx20h	Write	BA	xxD0h
Erase all unlocked block [(1)]	Write	×	xxA7h	Write	×	xxD0h
Lock bit program	Write	BA	xx77h	Write	BA	xxD0h
Read lock bit status	Write	×	xx71h	Write	BA	xxD0h

SRD:data in the SRD register (D7 to D0)

WA: Address to be written (The address specified in the first bus cycle is the same even address as the address specified in the second bus cycle.)

WD: 16-bit write data

BA: Highest-order block address (must be an even address)

×: Given even address in the user ROM area

xx: High-order 8 bits of command code (ignored)

NOTE:

1. Blocks 0 to 12 can be erased by the erase all unlocked block command.
   Block A cannot be erased. The block erase command must be used to erase the block A.

21.3.5.1 Read Array Command (FFh)

The read array command reads the flash memory.

By writing command code xxFFh in the first bus cycle, read array mode is entered. Content of a specified address can be read in 16-bit unit after the next bus cycle.

The MCU remains in read array mode until another command is written. Therefore, contents from multiple addresses can be read consecutively.

21.3.5.2 Read Status Register Command (70h)

The read status register command reads the status register (refer to **21.3.7 Status Register (SRD Register)** for detail).

By writing command code xx70h in the first bus cycle, the status register can be read in the second bus cycle. Read an even address in the user ROM area.

Do not execute this command in EW1 mode.

21.3.5.3 Clear Status Register Command (50h)

The clear status register command clears the status register.

By writing xx50h in the first bus cycle, bits FMR07 to FMR06 in the FMR0 register are set to 00b and bits SR5 to SR4 in the status register are set to 00b.

21.3.5.4 Program Command (40h)

The program command writes 2-byte data to the flash memory.

By writing xx40h in the first bus cycle and data to the write address in the second bus cycle, an auto-program operation (data program and verify) will start. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether an auto-program operation has been completed. The FMR00 bit is set to 0 (busy) during auto-programming and to 1 (ready) when an auto-program operation is completed.

After the completion of an auto-program operation, the FMR06 bit in the FMR0 register indicates whether or not the auto-program operation has been completed as expected. (Refer to **21.3.8 Full Status Check.**)

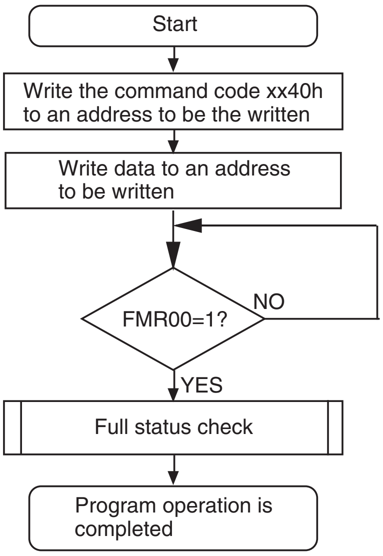
An address that is already written cannot be altered or rewritten.

Figure 21.8 shows a flow chart of the Program Command.

The lock bit protects each block from being programmed inadvertently. (Refer to **21.3.6 Data Protect Function.**)

In EW1 mode, do not execute this command on the block where the rewrite control program is allocated.

In EW0 mode, the MCU enters read status register mode as soon as an auto-program operation starts. The status register can be read. The SR7 bit in the status register is set to 0 at the same time an auto-program operation starts. It is set to 1 when auto-program operation is completed. The MCU remains in read status register mode until the read array command is written. After completion of an auto-program operation, the status register indicates whether or not the auto-program operation has been completed as expected.


Start
Write the command code xx40h to an address to be the written
Write data to an address to be written
FMR00=1?
NO
YES
Full status check
Program operation is completed

NOTE:
1. Write the command code and data to even addresses.

Figure 21.8 Program Command

21.3.5.5 Block Erase Command

The block erase command erases each block.

By writing xx20h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, an auto-erase operation (erase and verify) will start in the specified block.

The FMR00 bit in the FMR0 register indicates whether an auto-erase operation has been completed.

The FMR00 bit is set to 0 (busy) during auto-erasure and to 1 (ready) when the auto-erase operation is completed.

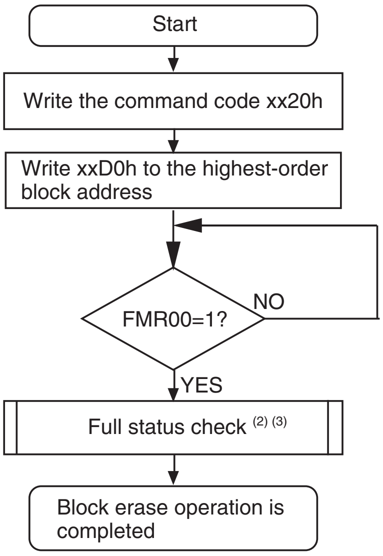
After the completion of an auto-erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto-erase operation has been completed as expected. (Refer to **21.3.8 Full Status Check**.)

Figure 21.9 shows a flow chart of the Block Erase Command.

The lock bit protects each block from being programmed inadvertently. (Refer to **21.3.6 Data Protect Function**.)

In EW1 mode, do not execute this command on the block where the rewrite control program is allocated.

In EW0 mode, the MCU enters read status register mode as soon as an auto-erase operation starts. The status register can be read. The SR7 bit in the status register is set to 0 at the same time an auto-erase operation starts. It is set to 1 when an auto-erase operation is completed. The MCU remains in read status register mode until the read array command or read lock bit status command is written. Also execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.


Start
Write the command code xx20h
Write xxD0h to the highest-order block address
FMR00=1?
NO
YES
Full status check (2) (3)
Block erase operation is completed

NOTES:
1. Write the command code and data to even addresses.
2. Refer to **Figure 21.12 Full Status Check and Handling Procedure for Each Error**.
3. Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.

Figure 21.9 Block Erase Command

21.3.5.6 Erase All Unlocked Block

The erase all unlocked block command erases all blocks except the block A.

By writing xxA7h in the first bus cycle and xxD0h in the second bus cycle, an auto-erase (erase and verify) operation will run continuously in all blocks except the block A.

The FMR00 bit in the FMR0 register indicates whether an auto-erase operation has been completed.

After the completion of an auto-erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto-erase operation has been completed as expected.

The lock bit can protect each block from being programmed inadvertently. (Refer to **21.3.6 Data Protect Function**.)

In EW1 mode, do not execute this command when the lock bit for any block storing the rewrite control program is set to 1 (unlocked) or when the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled).

In EW0 mode, the MCU enters read status register mode as soon as an auto-erase operation starts. The status register can be read. The SR7 bit in the status register is set to 0 (busy) at the same time an auto-erase operation starts. It is set to 1 (ready) when an auto-erase operation is completed. The MCU remains in read status register mode until the read array command or read lock bit status command is written.

Only blocks 0 to 12 can be erased by the erase all unlocked block command. The block A cannot be erased. Use the block erase command to erase the block A.

21.3.5.7 Lock Bit Program Command

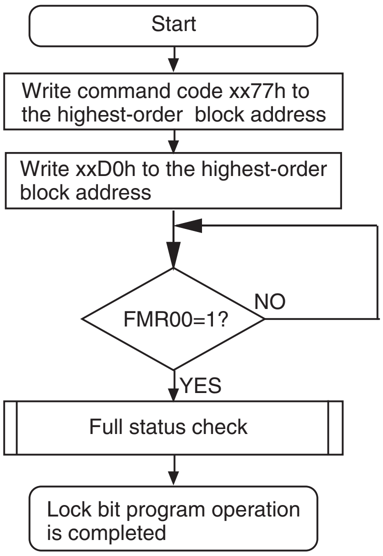
The lock bit program command sets the lock bit for a specified block to 0 (locked).

By writing xx77h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, the lock bit for the specified block is set to 0. The address value specified in the first bus cycle must be the same highest-order even address of a block specified in the second bus cycle.

Figure 21.10 shows a flow chart of the Lock Bit Program Command. Execute read lock bit status command to read lock bit state (lock bit data).

The FMR00 bit in the FMR0 register indicates whether a lock bit program operation is completed.

Refer to **21.3.6 Data Protect Function** for details on lock bit functions and how to set it to 1 (unlocked).


Start
Write command code xx77h to the highest-order block address
Write xxD0h to the highest-order block address
FMR00=1?
NO
YES
Full status check
Lock bit program operation is completed

NOTE:
1. Write the command code and data to even addresses.

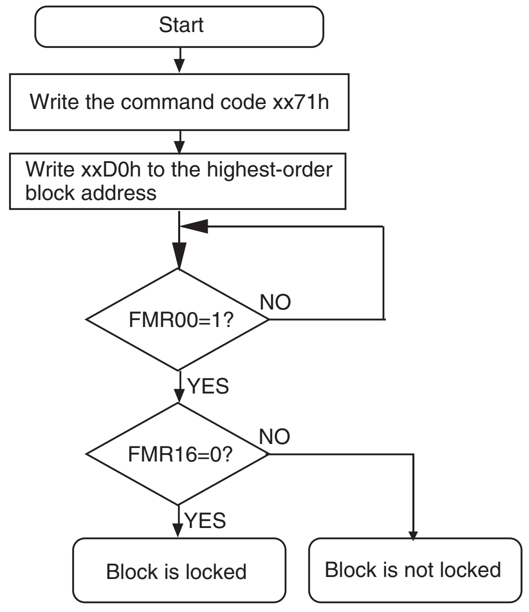
Figure 21.10 Lock Bit Program Command

21.3.5.8 Read Lock Bit Status Command (71h)

The read lock bit status command reads the lock bit state of a specified block.

By writing xx71h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, the FMR16 bit in the FMR1 register stores information on whether or not the lock bit of a specified block is locked. Read the FMR16 bit after the FMR00 bit in the FMR0 register is set to 1 (ready).

Figure 21.11 shows a flow chart of the Read Lock Bit Status Command.


Start
Write the command code xx71h
Write xxD0h to the highest-order block address
FMR00=1?
NO
YES
FMR16=0?
NO
YES
Block is locked
Block is not locked

NOTE:
1. Write the command code and data to even addresses.

Figure 21.11 Read Lock Bit Status Command

21.3.6 Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is enabled by setting the FMR02 bit in the FMR0 register to 0 (lock bit enabled). The lock bit allows each block to be individually protected (locked) against program and erase. This helps prevent data from being inadvertently written to or erased from the flash memory.

- When the lock bit status is set to 0, the block is locked (block is protected against program and erase).
- When the lock bit status is set to 1, the block is not locked (block can be programmed or erased).

The lock bit status is set to 0 (locked) by executing the lock bit program command and to 1 (unlocked) by erasing the block. The lock bit status cannot be set to 1 by any commands.
The lock bit status can be read by the read lock bit status command.

The lock bit function is disabled by setting the FMR02 bit to 1 (lock bit disabled). All blocks are unlocked. However, individual lock bit status remains unchanged. The lock bit function is enabled by setting the FMR02 bit to 0. Lock bit status is retained.
If the block erase or erase all unlocked block command is executed while the FMR02 bit is set to 1, the target block or all blocks are erased regardless of lock bit status. The lock bit status of each block are set to 1 after an erase operation is completed.
Refer to **21.3.5 Software Commands** for details on each command.

21.3.7 Status Register (SRD Register)

The status register indicates the operating status of the flash memory and whether or not an erase or program operation is completed as expected. Bits FMR00, FMR06, and FMR07 in the FMR0 register indicate status register states.
Table 21.5 shows the Status Register.
In EW0 mode, the status register can be read when the followings occur.

- Given even address in the user ROM area is read after writing the read status register command.
- Given even address in the user ROM area is read from when the program, block erase, erase all unlocked block, or lock bit program command is executed until when the read array command is executed.

21.3.7.1 Sequencer Status (Bits SR7 and FMR00)

The sequencer status indicates the operating status of the flash memory. It is set to 0 while the program, block erase, erase all unlocked block, lock bit program, or read lock bit status command is being executed; otherwise, it is set to 1.

21.3.7.2 Erase Status (Bits SR5 and FMR07)

Refer to **21.3.8 Full Status Check**.

21.3.7.3 Program Status (Bits SR4 and FMR06)

Refer to **21.3.8 Full Status Check**.

Table 21.5 Status Register

Bits in Status Register	Bits in FMR0 Register	Status Name	Contents		Value after Reset
			0	1	
SR0 (D0)	-	Reserved	-	-	-
SR1 (D1)	-	Reserved	-	-	-
SR2 (D2)	-	Reserved	-	-	-
SR3 (D3)	-	Reserved	-	-	-
SR4 (D4)	FMR06	Program status	Terminated normally	Terminated in error	0
SR5 (D5)	FMR07	Erase status	Terminated normally	Terminated in error	0
SR6 (D6)	-	Reserved	-	-	-
SR7 (D7)	FMR00	Sequencer status	Busy	Ready	1

D0 to D7: These data bus are read when the read status register command is executed.

NOTE:

1. Bits FMR06 (SR4) and FMR07 (SR5) are set to 0 by executing the clear status register command. When the FMR06 bit (SR4) or FMR07 bit (SR5) is set to 1, the program, block erase, erase all unlocked block and lock bit program commands are not accepted.

21.3.8 Full Status Check

If an error occurs when a program or erase operation is completed, the FMR06, FMR07 bits in the FMR0 register are set to 1, indicating a specific error. Therefore, execution results can be confirmed by checking these bits (full status check).

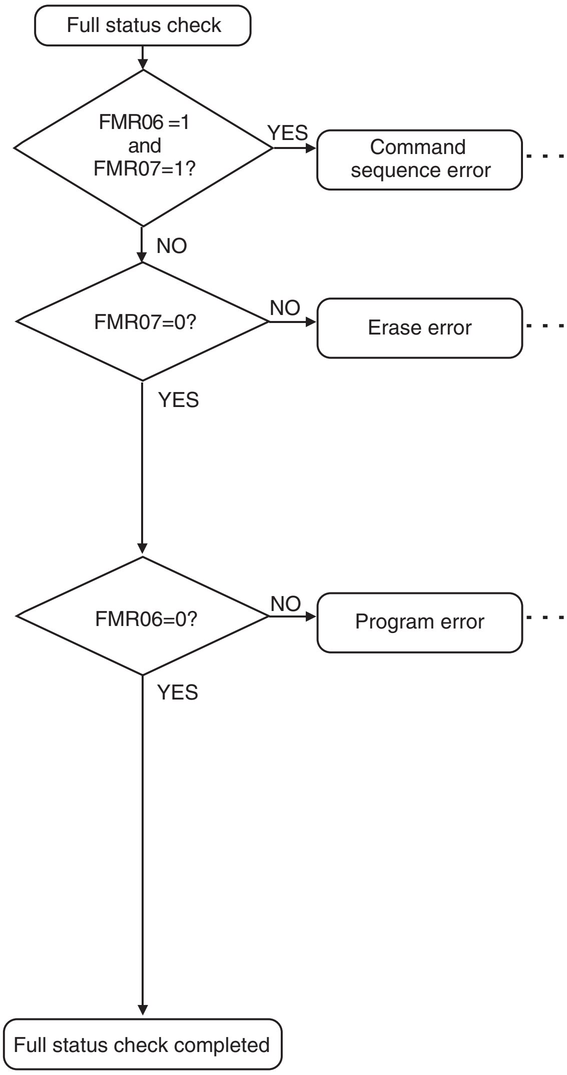
Table 21.6 lists the Errors and FMR0 Register Status. Figure 21.12 shows a flow chart of the Full Status Check and Handling Procedure for Each Error.

Table 21.6 Errors and FMR0 Register Status

FRM00 Register (Status Register) Status		Error	Error Occurrence Conditions
FMR07 Bit (SR5)	FMR06 Bit (SR4)		
1	1	Command Sequence error	• Command is written incorrectly • A value other than xxD0h or xxFFh is written in the second bus cycle of the lock bit program, block erase or erase all unlocked block command [1]
1	0	Erase error	• The block erase command is executed on a locked block [2] • The block erase or erase all unlocked block command is executed on an unlock block and auto-erase operation is not completed as expected
0	1	Program error	• The program command is executed on locked blocks [2] • The program command is executed on unlocked blocks and auto-program operation is not completed as expected • The lock bit program command is executed but program operation is not completed as expected

NOTES:

1. The flash memory enters read array mode by writing command code xxFFh in the second bus cycle of these commands. The command code written in the first bus cycle becomes invalid.
2. When the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled), no error occurs even under the conditions above.



Command sequence error:

(1) Execute the clear status register command and set bits SR4 and SR5 to 0 (completed as expected).
(2) Rewrite command and execute again.

Erase error:

(1) Execute the clear status register command and set the SR5 bit to 0.
(2) Execute the lock bit read status command. Set the FMR02 bit in the FMR0 register to 1 (lock bit disabled) if the lock bit in the block where the error occurred is set to 0 (locked).
(3) Execute the block erase or erase all unlocked block command again.
(4) Execute (1), (2), and (3) at least 3 times until an erase error is not generated.

NOTE: If similar error occurs, that block cannot be used.
If the lock bit is set to 1 (unlocked) in (2) above, that block cannot be used.

Program error:

[When a program operation is executed]
(1) Execute the clear status register command and set the SR4 bit to 0 (completed as expected).
(2) Execute the read lock bit status command and set the FMR02 bit to 1 if the lock bit in the block where the error occurred is set to 0.
(3) Execute the program command again.

NOTE: When a similar error occurs, that block cannot be used.
If the lock bit is set to 1 in (2) above, that block cannot be used.

[When a lock bit program operation is executed]
(1) Execute the clear status register command and set the SR4 bit to 0.
(2) Set the FMR02 bit to 1.
(3) Execute the block erase command to erase the block where the error occurred.
(4) Execute the lock bit program command again.

NOTE: If similar error occurs, that block cannot be used.

FMR06, FMR07: Bits in FMR0 register

NOTE:
1. When either FMR06 or FMR07 bit is set to 1 (terminated by error), the program, block erase, erase all unlocked block, lock bit program and read lock bit status commands cannot be accepted.
Execute the clear status register command before each command.

Figure 21.12 Full Status Check and Handling Procedure for Each Error

21.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/6N Group (M16C/6NL, M16C/6NN) can be used to rewrite the flash memory user ROM area in the MCU mounted on a board. For more information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual included with your serial programmer for instructions.

Table 21.7 lists the Pin Functions in Standard Serial I/O Mode. Figures 21.13 and 21.14 show the Pin Connections in Standard Serial I/O Mode.

21.4.1 ID Code Check Function

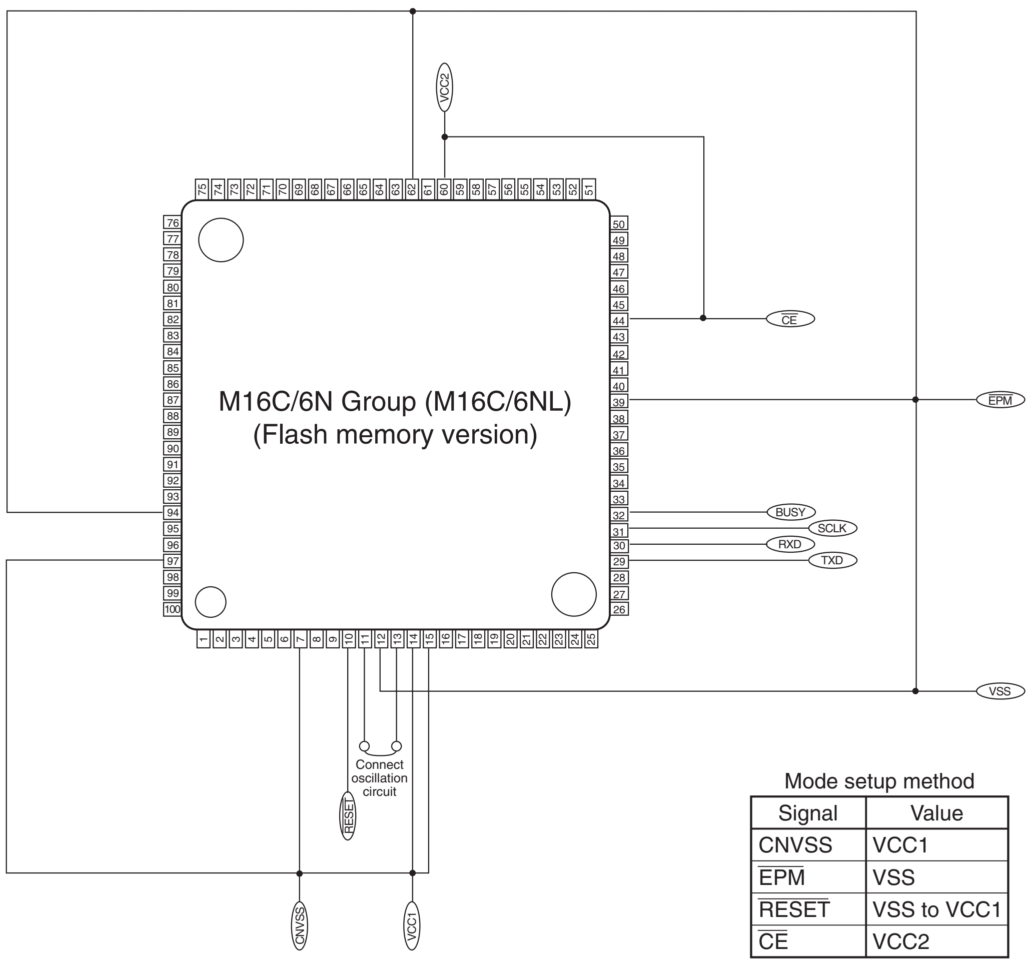
The ID code check function determines whether the ID codes sent from the serial programmer matches those written in the flash memory. (Refer to **21.2 Functions to Prevent Flash Memory from Rewriting**.)

Table 21.7 Pin Functions in Standard Serial I/O Mode

Pin	Name	I/O	Description
VCC1, VCC2, VSS	Power supply input		Apply the Flash Program, Erase Voltage to VCC1 pin and VCC2 to VCC2 pin. The VCC apply condition is that VCC2 = VCC1. Apply 0 V to VSS pin.
CNVSS	CNVSS	I	Connect to VCC1 pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, input 20 cycles or longer clock to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
BYTE	BYTE	I	Connect this pin to VCC1 or VSS.
AVCC, AVSS	Analog power supply input		Connect AVCC to VCC1 and AVSS to VSS, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for A/D and D/A converters from this pin.
P0_0 to P0_7	Input port P0	I	Input “H” or “L” level signal or open.
P1_0 to P1_7	Input port P1	I	Input “H” or “L” level signal or open.
P2_0 to P2_7	Input port P2	I	Input “H” or “L” level signal or open.
P3_0 to P3_7	Input port P3	I	Input “H” or “L” level signal or open.
P4_0 to P4_7	Input port P4	I	Input “H” or “L” level signal or open.
P5_0	$\overline{\text{CE}}$ input	I	Input “H” level signal.
P5_1 to P5_4, P5_6, P5_7	Input port P5	I	Input “H” or “L” level signal or open.
P5_5	$\overline{\text{EPM}}$ input	I	Input “L” level signal.
P6_0 to P6_3	Input port P6	I	Input “H” or “L” level signal or open.
P6_4/$\overline{\text{RTS1}}$	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
P6_5/CLK1	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin. Standard serial I/O mode 2: Input “L”.
P6_6/RXD1	RXD input	I	Serial data input pin
P6_7/TXD1	TXD output	O	Serial data output pin [1]
P7_0 to P7_7	Input port P7	I	Input “H” or “L” level signal or open.
P8_0 to P8_3, P8_6, P8_7	Input port P8	I	Input “H” or “L” level signal or open.
P8_4	P8_4 input	I	Input “L” level signal. [2]
P8_5/$\overline{\text{NMI}}$	$\overline{\text{NMI}}$ input	I	Connect this pin to VCC1.
P9_0 to P9_4, P9_7	Input port P9	I	Input “H” or “L” level signal or open.
P9_5/CRX0	CRX input	I	Input “H” or “L” level signal or connect to a CAN transceiver.
P9_6/CTX0	CTX output	O	Input “H” level signal, open or connect to a CAN transceiver.
P10_0 to P10_7	Input port P10	I	Input “H” or “L” level signal or open.
P11_0 to P11_7 [3]	Input port P11	I	Input “H” or “L” level signal or open.
P12_0 to P12_7 [3]	Input port P12	I	Input “H” or “L” level signal or open.
P13_0 to P13_7 [3]	Input port P13	I	Input “H” or “L” level signal or open.
P14_0, P14_1 [3]	Input port P14	I	Input “H” or “L” level signal or open.

NOTES:

1. When using standard serial I/O mode, It is necessary to input “H” to the TXD1(P6_7) pin while the $\overline{\text{RESET}}$ pin is “L”. Therefore, the internal pull-up is enabled for the TXD1(P6_7) pin while the $\overline{\text{RESET}}$ pin is “L”.
2. When using standard serial I/O mode, pins P0_0 to P0_7, P1_0 to P1_7 may become undefined while the P8_4 pin is “H” and the $\overline{\text{RESET}}$ pin is “L”. If this causes a problem, apply “L” to the P8_4 pin.
3. The pins P11 to P14 are only in the 128-pin version.

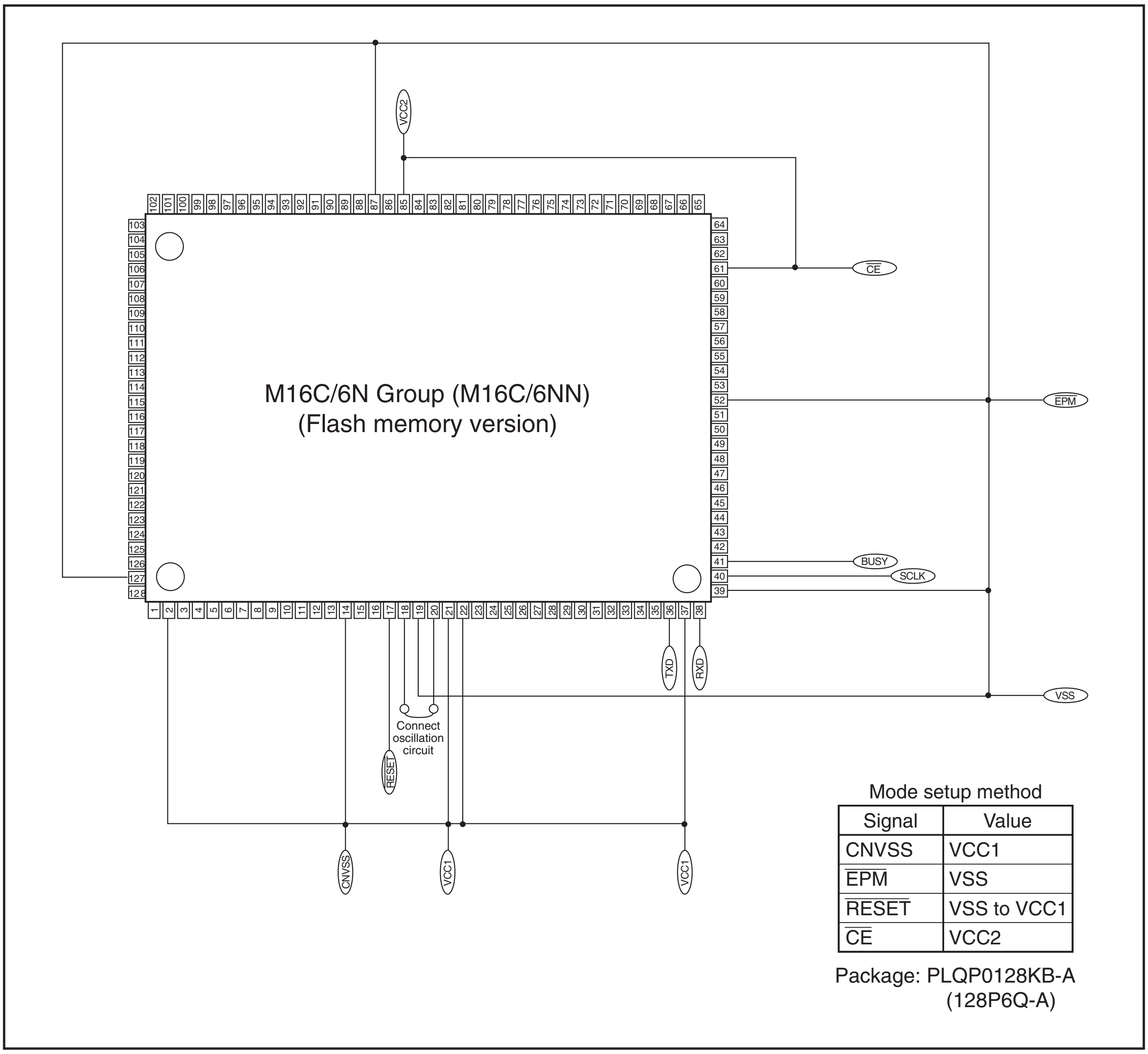


Mode setup method

Signal	Value
CNVSS	VCC1
$\overline{\text{EPM}}$	VSS
$\overline{\text{RESET}}$	VSS to VCC1
$\overline{\text{CE}}$	VCC2

Package: PLQP0100KB-A
(100P6Q-A)

Figure 21.13 Pin Connections in Standard Serial I/O Mode (1)



Mode setup method

Signal	Value
CNVSS	VCC1
$\overline{\text{EPM}}$	VSS
$\overline{\text{RESET}}$	VSS to VCC1
$\overline{\text{CE}}$	VCC2

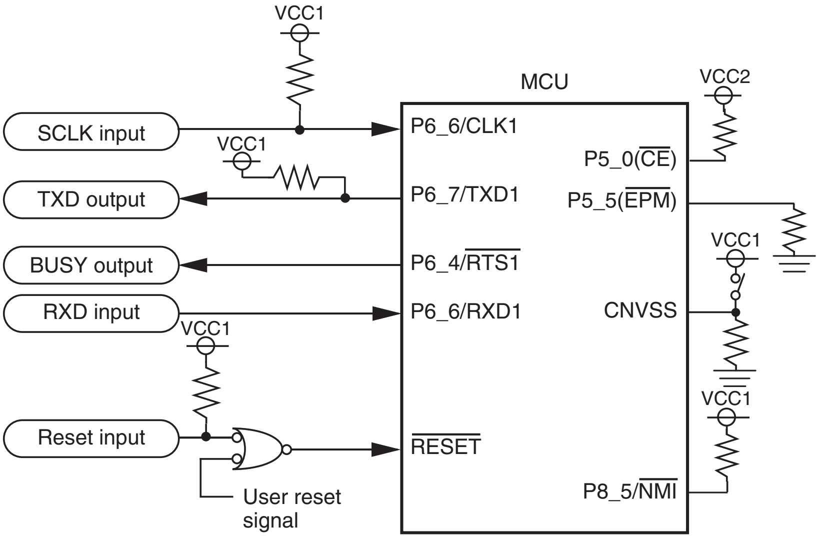
Package: PLQP0128KB-A
(128P6Q-A)

Figure 21.14 Pin Connections in Standard Serial I/O Mode (2)

21.4.2 Example of Circuit Application in Standard Serial I/O Mode

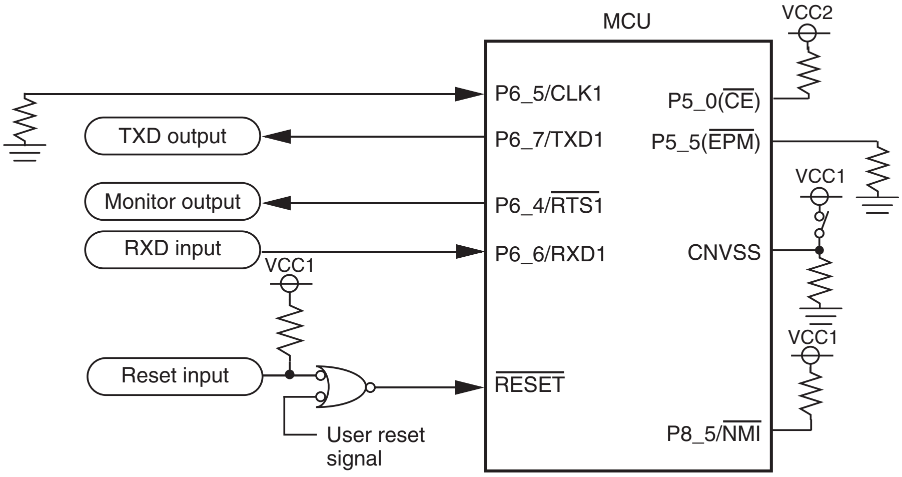
Figures 21.15 and 21.16 show the Circuit Application in Standard Serial I/O Mode 1 and Mode 2. Refer to the user's manual of your serial programmer to handle pins controlled by a serial programmer.

Note that when using standard serial I/O mode 2, make sure a main clock input oscillation frequency is set to 5 MHz, 10 MHz, or 16 MHz.



NOTES:
1. Control pins and external circuitry will vary according to programmer. For more information, refer to the programmer manual.
2. In this example, modes are switched between single-chip mode and standard serial I/O mode by controlling the CNVSS input with a switch.
3. If in standard standard serial I/O mode 1 there is a possibility that the user reset signal will go low during standard serial I/O mode, break the connection between the user reset signal and RESET pin by using, for example, a jumper switch.

Figure 21.15 Circuit Application in Standard Serial I/O Mode 1



NOTE:
1. In this example, modes are switched between single-chip mode and standard serial I/O mode by controlling the CNVSS input with a switch.

Figure 21.16 Circuit Application in Standard Serial I/O Mode 2

21.5 Parallel I/O Mode

In parallel I/O mode, the user ROM area and the boot ROM area can be rewritten by a parallel programmer supporting the M16C/6N Group (M16C/6NL, M16C/6NN). Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

21.5.1 User ROM and Boot ROM Areas

An erase block operation in the boot ROM area is applied to only one 4-Kbyte block. The rewrite control program in standard serial I/O and CAN I/O modes are written in the boot ROM area before shipment. Do not rewrite the boot ROM area if using the serial programmer.
In parallel I/O mode, the boot ROM area is located in addresses 0FF000h to 0FFFFFh. Rewrite this address range only if rewriting the boot ROM area. (Do not access addresses other than addresses 0FF000h to 0FFFFFh.)

21.5.2 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read and rewritten in parallel I/O mode. (Refer to **21.2 Functions to Prevent Flash Memory from Rewriting**.)

21.6 CAN I/O Mode

In CAN I/O mode, the CAN programmer supporting the M16C/6N Group (M16C/6NL, M16C/6NN) can be used to rewrite the flash memory user ROM area in the MCU mounted on a board. For more information about the CAN programmer, contact your CAN programmer manufacturer. Refer to the user's manual included with your CAN programmer for instructions.

Table 21.8 lists pin functions for CAN I/O mode. Figures 21.17 and 21.18 show pin connections in CAN I/O mode.

21.6.1 ID Code Check Function

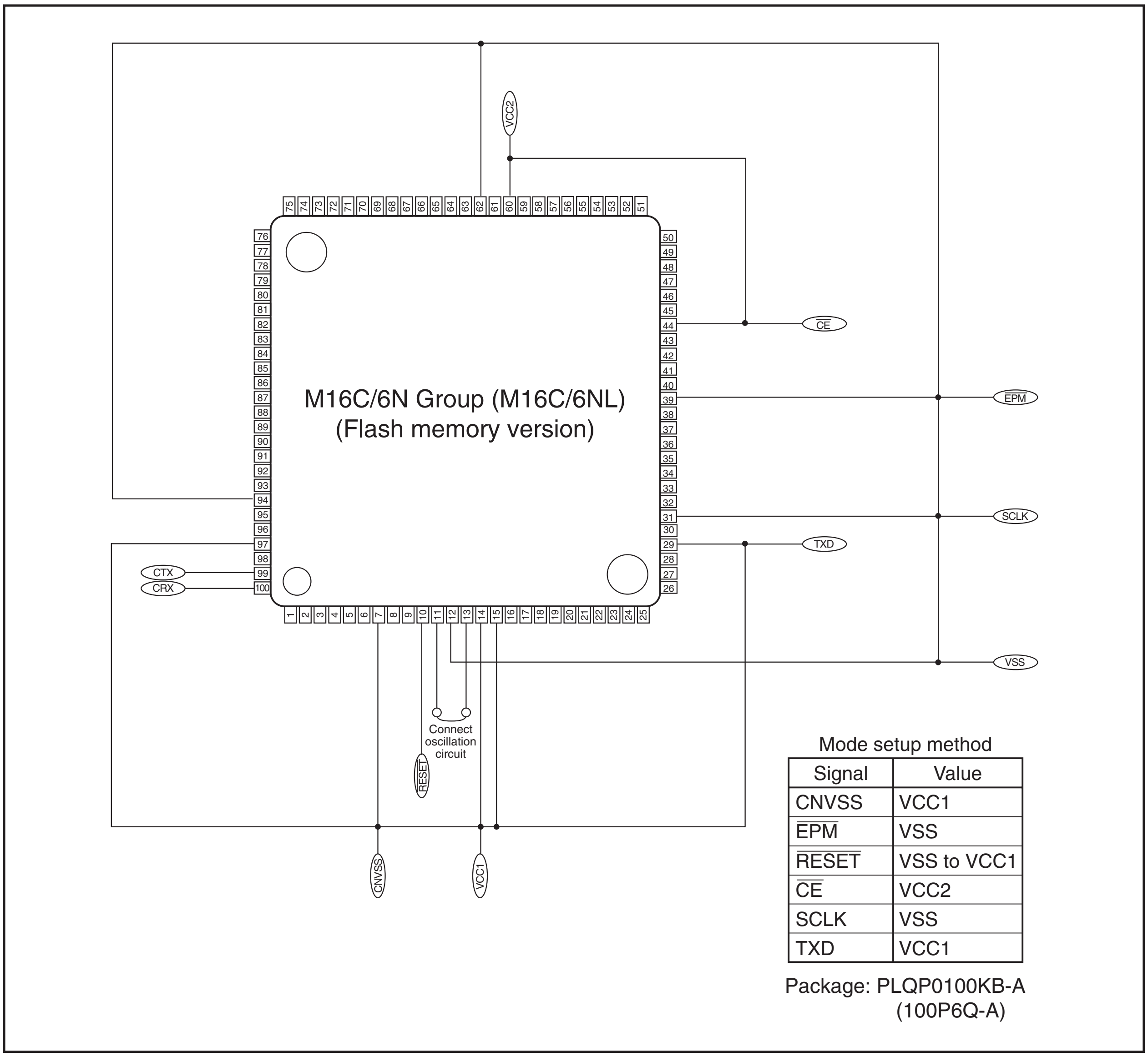
The ID code check function determines whether the ID codes sent from the CAN programmer matches those written in the flash memory. (Refer to **21.2 Functions to Prevent Flash Memory from Rewriting**.)

Table 21.8 Pin Functions for CAN I/O Mode

Pin	Name	I/O	Description
VCC1, VCC2, VSS	Power supply input		Apply the Flash Program, Erase Voltage to VCC1 pin and VCC2 to VCC2 pin. The VCC apply condition is that VCC2 = VCC1. Apply 0 V to VSS pin.
CNVSS	CNVSS	I	Connect to VCC1 pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is “L” level, input 20 cycles or longer clock to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
BYTE	BYTE	I	Connect this pin to VCC1 or VSS.
AVCC, AVSS	Analog power supply input		Connect AVCC to VCC1 and AVSS to VSS, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for A/D and D/A converters from this pin.
P0_0 to P0_7	Input port P0	I	Input “H” or “L” level signal or open.
P1_0 to P1_7	Input port P1	I	Input “H” or “L” level signal or open.
P2_0 to P2_7	Input port P2	I	Input “H” or “L” level signal or open.
P3_0 to P3_7	Input port P3	I	Input “H” or “L” level signal or open.
P4_0 to P4_7	Input port P4	I	Input “H” or “L” level signal or open.
P5_0	$\overline{\text{CE}}$ input	I	Input “H” level signal.
P5_1 to P5_4, P5_6, P5_7	Input port P5	I	Input “H” or “L” level signal or open.
P5_5	$\overline{\text{EPM}}$ input	I	Input “L” level signal.
P6_0 to P6_4, P6_6	Input port P6	I	Input “H” or “L” level signal or open.
P6_5/CLK1	SCLK input	I	Input “L” level signal.
P6_7/TXD1	TXD output	O	Input “H” level signal.
P7_0 to P7_7	Input port P7	I	Input “H” or “L” level signal or open.
P8_0 to P8_3, P8_6, P8_7	Input port P8	I	Input “H” or “L” level signal or open.
P8_4	P8_4 Input	I	Input “L” level signal. (1)
P8_5/$\overline{\text{NMI}}$	$\overline{\text{NMI}}$ input	I	Connect this pin to VCC1.
P9_0 to P9_4, P9_7	Input port P9	I	Input “H” or “L” level signal or open.
P9_5/CRX0	CRX input	I	Connect to a CAN transceiver.
P9_6/CTX0	CTX output	O	Connect to a CAN transceiver.
P10_0 to P10_7	Input port P10	I	Input “H” or “L” level signal or open.
P11_0 to P11_7 (2)	Input port P11	I	Input “H” or “L” level signal or open.
P12_0 to P12_7 (2)	Input port P12	I	Input “H” or “L” level signal or open.
P13_0 to P13_7 (2)	Input port P13	I	Input “H” or “L” level signal or open.
P14_0, P14_1 (2)	Input port P14	I	Input “H” or “L” level signal or open.

NOTES:

1. When using CAN I/O mode, pins P0_0 to P0_7, P1_0 to P1_7 may become undefined while the P8_4 pin is “H” and the $\overline{\text{RESET}}$ pin is “L”. If this causes a problem, apply “L” to the P8_4 pin.
2. The pins P11 to P14 are only in the 128-pin version.

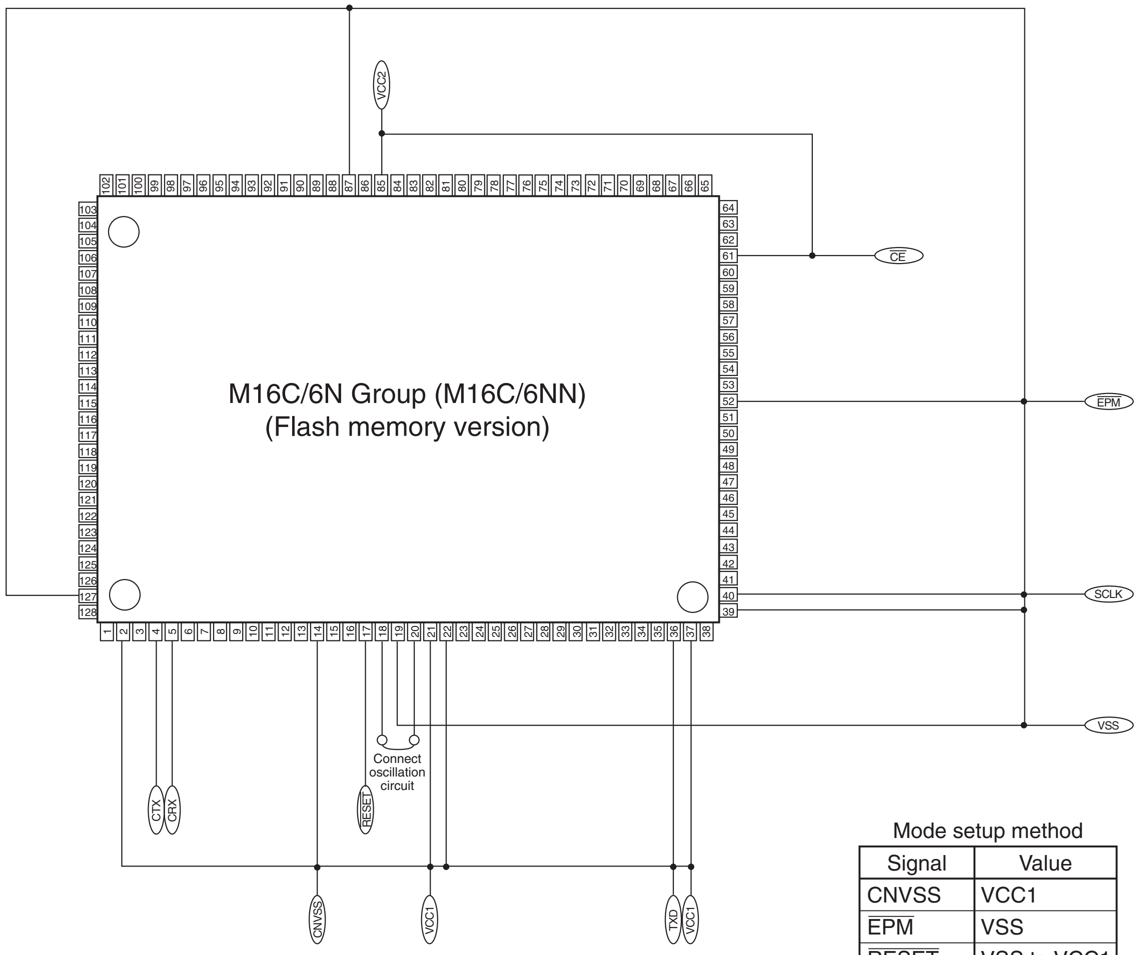


Mode setup method

Signal	Value
CNVSS	VCC1
EPM	VSS
RESET	VSS to VCC1
CE	VCC2
SCLK	VSS
TXD	VCC1

Package: PLQP0100KB-A
(100P6Q-A)

Figure 21.17 Pin Connections in CAN I/O Mode (1)



Mode setup method

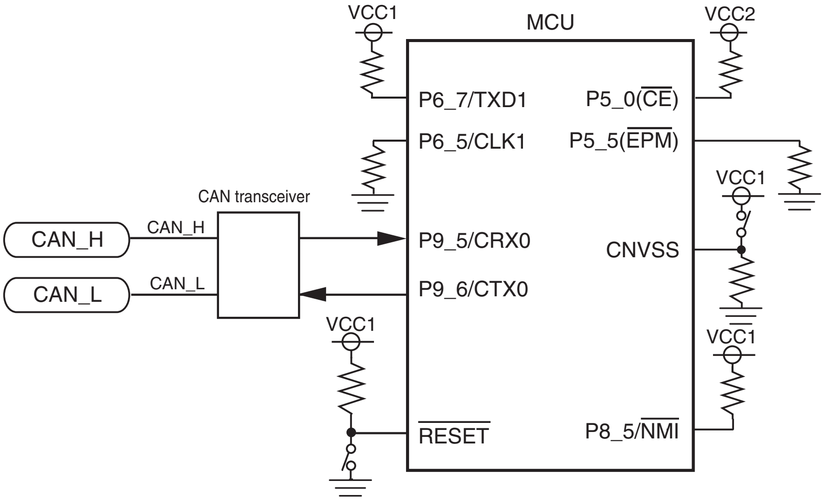
Signal	Value
CNVSS	VCC1
$\overline{\text{EPM}}$	VSS
$\overline{\text{RESET}}$	VSS to VCC1
$\overline{\text{CE}}$	VCC2
SCLK	VSS
TXD	VCC1

Package: PLQP0128KB-A
(128P6Q-A)

Figure 21.18 Pin Connections in CAN I/O Mode (2)

21.6.2 Example of Circuit Application in CAN I/O Mode

Figure 21.19 shows the Circuit Application in CAN I/O Mode. Refer to the user's manual of your CAN programmer to handle pins controlled by a CAN programmer.



NOTES:

1. Control pins and external circuitry will vary according to programmer. For more information, refer to the programmer manual.
2. In this example, modes are switched between single-chip mode and CAN I/O mode by controlling the CNVSS input with a switch.

Figure 21.19 Circuit Application in CAN I/O Mode

22. Electrical Characteristics

Table 22.1 Absolute Maximum Ratings

Symbol	Parameter		Condition	Rated Value	Unit
V_{CC}	Supply voltage (VCC1 = VCC2)		VCC = AVCC	–0.3 to 6.5	V
AV_{CC}	Analog supply voltage		VCC = AVCC	–0.3 to 6.5	V
V_I	Input voltage	$\overline{\text{RESET}}$, CNVSS, BYTE, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, VREF, XIN		–0.3 to VCC+0.3	V
		P7_1, P9_1		–0.3 to 6.5	V
V_O	Output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XOUT		–0.3 to VCC+0.3	V
		P7_1, P9_1		–0.3 to 6.5	V
P_d	Power dissipation		Topr = 25°C	700	mW
T_{opr}	Operating ambient temperature	During MCU operation		–40 to 85	°C
		During flash memory program and erase operation		0 to 60	
T_{stg}	Storage temperature			–65 to 150	°C

NOTE:

1. Ports P11 to P14 are only in the 128-pin version.

Table 22.2 Recommended Operating Conditions (1) (1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply voltage (VCC1 = VCC2)		3.0	5.0	5.5	V
AV_{CC}	Analog supply voltage			V_{CC}		V
V_{SS}	Supply voltage			0		V
AV_{SS}	Analog supply voltage			0		V
V_{IH}	HIGH input voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	0.8 V_{CC}		V_{CC}	V
		P7_1, P9_1	0.8 V_{CC}		6.5	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode)	0.8 V_{CC}		V_{CC}	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes)	0.5 V_{CC}		V_{CC}	
V_{IL}	LOW input voltage	P3_1 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	0		0.2 V_{CC}	V V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (During single-chip mode)	0		0.2 V_{CC}	V
		P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 (Data input during memory expansion and microprocessor modes)	0		0.16 V_{CC}	V
$I_{OH(peak)}$	HIGH peak output current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			–10.0	mA
$I_{OH(avg)}$	HIGH average output current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			–5.0	mA
$I_{OL(peak)}$	LOW peak output current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			10.0	mA
$I_{OL(avg)}$	LOW average output current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1			5.0	mA

NOTES:

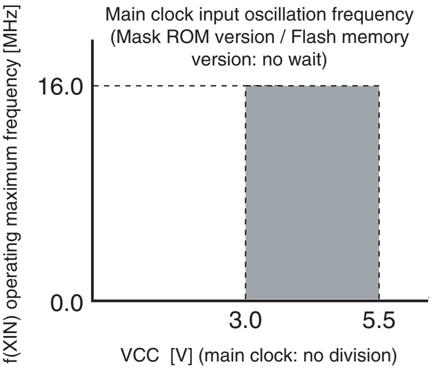
1. Referenced to VCC = 3.0 to 5.5 V at Topr = –40 to 85°C unless otherwise specified.
2. Average output current values during 100 ms period.
3. The total $I_{OL(peak)}$ for ports P0, P1, P2, P8_6, P8_7, P9, P10, P11, P14_0, and P14_1 must be 80 mA max.
   The total $I_{OL(peak)}$ for ports P3, P4, P5, P6, P7, P8_0 to P8_4, P12, and P13 must be 80 mA max.
   The total $I_{OH(peak)}$ for ports P0, P1, and P2 must be –40 mA max.
   The total $I_{OH(peak)}$ for ports P3, P4, P5, P12, and P13 must be –40 mA max.
   The total $I_{OH(peak)}$ for ports P6, P7, and P8_0 to P8_4 must be –40 mA max.
   The total $I_{OH(peak)}$ for ports P8_6, P8_7, P9, P10, P11, P14_0, and P14_1 must be –40 mA max.
4. P11 to P14 are only in the 128-pin version.

Table 22.3 Recommended Operating Conditions (2) [1]

Symbol	Parameter				Standard			Unit
					Min.	Typ.	Max.	
f(XIN)	Main clock input oscillation frequency [2] [3] [4]	No wait	Mask ROM version Flash memory version	VCC = 3.0 to 5.5 V	0		16	MHz
f(XCIN)	Sub clock oscillation frequency					32.768	50	kHz
f(Ring)	On-chip oscillation frequency					1		MHz
f(PLL)	PLL clock oscillation frequency				16		24	MHz
f(BCLK)	CPU operation clock			VCC = 3.0 to 5.5 V	0		24	MHz
$t_{su(PLL)}$	PLL frequency synthesizer stabilization wait time						20	ms
$f_{(ripple)}$	Power supply ripple allowable frequency (VCC)						10	kHz
$V_{P-P(ripple)}$	Power supply ripple allowable amplitude voltage			VCC = 5 V			0.5	V
				VCC = 3 V			0.3	
$V_{CC(\|\Delta V/\Delta T\|)}$	Power supply ripple rising/falling gradient			VCC = 5 V			0.3	V/ms
				VCC = 3 V			0.3	

NOTES:

1. Referenced to VCC = 3.0 to 5.5 V at Topr = –40 to 85°C unless otherwise specified.
2. Relationship between main clock oscillation frequency and supply voltage is shown right.
3. Execute program/erase of flash memory by VCC = 3.3 ± 0.3 V or VCC = 5.0 ± 0.5 V.
4. When using 16 MHz and over, use PLL clock. PLL clock oscillation frequency which can be used is 16 MHz, 20 MHz or 24 MHz.



$f_{(ripple)}$
Power supply ripple allowable frequency (VCC)

$V_{P-P(ripple)}$
Power supply ripple allowable amplitude voltage

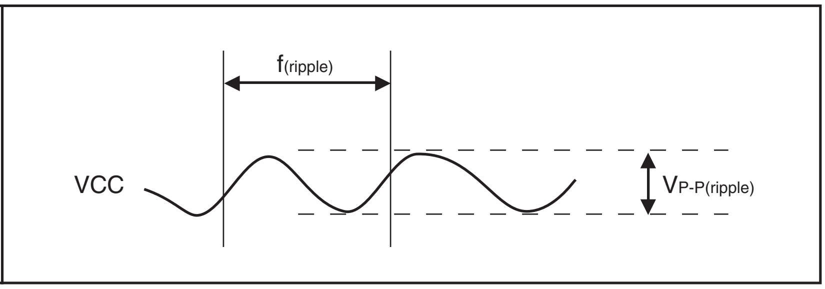


Figure 22.1 Voltage Fluctuation Timing

Table 22.4 Electrical Characteristics (1) [1]

VCC = 5V

Symbol	Parameter			Measuring Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		I_{OH} = –5 mA	V_{CC}-2.0		V_{CC}	V
V_{OH}	HIGH output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		I_{OH} = –200 μA	V_{CC}-0.3		V_{CC}	V
V_{OH}	HIGH output voltage	XOUT	HIGHPOWER	I_{OH} = –1 mA	3.0		V_{CC}	V
			LOWPOWER	I_{OH} = –0.5 mA	3.0		V_{CC}	
	HIGH output voltage	XCOUT	HIGHPOWER	With no load applied		2.5		V
			LOWPOWER	With no load applied		1.6		
V_{OL}	LOW output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		I_{OL} = 5 mA			2.0	V
V_{OL}	LOW output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		I_{OL} = 200 μA			0.45	V
V_{OL}	LOW output voltage	XOUT	HIGHPOWER	I_{OL} = 1 mA			2.0	V
			LOWPOWER	I_{OL} = 0.5 mA			2.0	
	LOW output voltage	XCOUT	HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, TA0IN to TA4IN, TB0IN to TB5IN, $\overline{INT0}$ to $\overline{INT8}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK6, TA0OUT to TA4OUT, $\overline{KI0}$ to $\overline{KI3}$, RXD0 to RXD2, SIN3 to SIN6			0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$			0.2		2.5	V
I_{IH}	HIGH input current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, $\overline{RESET}$, CNVSS, BYTE		V_I = 5 V			5.0	μA
I_{IL}	LOW input current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, $\overline{RESET}$, CNVSS, BYTE		V_I = 0 V			–5.0	μA
R_{PULLUP}	Pull-up resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		V_I = 0 V	30	50	170	kΩ
R_{fXIN}	Feedback resistance		XIN			1.5		MΩ
R_{fXCIN}	Feedback resistance		XCIN			15		MΩ
V_{RAM}	RAM retention voltage			At stop mode	2.0			V

NOTES:

1. Referenced to VCC = 4.2 to 5.5 V, VSS = 0 V at Topr = –40 to 85°C, f(BCLK) = 24 MHz unless otherwise specified.
2. P11 to P14, $\overline{INT6}$ to $\overline{INT8}$, CLK5, CLK6, SIN5, and SIN6 are only in the 128-pin version.

Table 22.5 Electrical Characteristics (2) [1]

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power supply current (VCC = 3.0 to 5.5 V)	In single-chip mode, the output pins are open and other pins are VSS.	Mask ROM	f(BCLK) = 24 MHz, PLL operation, No division		19	33	mA
				On-chip oscillation, No division		1		mA
			Flash memory	f(BCLK) = 24 MHz, PLL operation, No division		21	35	mA
				On-chip oscillation, No division		1.8		mA
			Flash memory program	f(BCLK) = 10 MHz, VCC = 5 V		15		mA
			Flash memory erase	f(BCLK) = 10 MHz, VCC = 5 V		25		mA
			Mask ROM	f(BCLK) = 32 kHz, Low power dissipation mode, ROM [2]		25		µA
			Flash memory	f(BCLK) = 32 kHz, Low power dissipation mode, RAM [2]		25		µA
				f(BCLK) = 32 kHz, Low power dissipation mode, Flash memory [2]		420		µA
			Mask ROM Flash memory	On-chip oscillation, Wait mode		50		µA
				f(BCLK) = 32 kHz, Wait mode [3], Oscillation capacity High		8.5		µA
				f(BCLK) = 32 kHz, Wait mode [3], Oscillation capacity Low		3.0		µA
				Stop mode, Topr = 25°C		0.8	3.0	µA

NOTES:

1. Referenced to VCC = 3.0 to 5.5 V, VSS = 0 V at Topr = –40 to 85°C, f(BCLK) = 24 MHz unless otherwise specified.
2. This indicates the memory in which the program to be executed exists.
3. With one timer operated using fC32.

Table 22.6 A/D Conversion Characteristics [1]

Symbol	Parameter		Measuring Condition		Standard			Unit
					Min.	Typ.	Max.	
–	Resolution		VREF = VCC				10	Bit
INL	Integral nonlinearity error	10 bits	VREF = VCC = 5 V	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±3	LSB
				External operation amp connection mode			±7	LSB
			VREF = VCC = 3.3 V	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±5	LSB
				External operation amp connection mode			±7	LSB
		8 bits	VREF = AVCC = VCC = 3.3 V				±2	LSB
–	Absolute accuracy	10 bits	VREF = VCC = 5 V	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±3	LSB
				External operation amp connection mode			±7	LSB
			VREF = VCC = 3.3 V	ANEX0, ANEX1 input, AN0 to AN7 input, AN0_0 to AN0_7 input, AN2_0 to AN2_7 input			±5	LSB
				External operation amp connection mode			±7	LSB
		8 bits	VREF = AVCC = VCC = 3.3 V				±2	LSB
DNL	Differential nonlinearity error						±1	LSB
–	Offset error						±3	LSB
–	Gain error						±3	LSB
R_{LADDER}	Resistor ladder		VREF = VCC		10		40	kΩ
t_{CONV}	10-bit conversion time, sample & hold available		VREF = VCC = 5 V, φAD = 10 MHz		3.3			µs
	8-bit conversion time, sample & hold available		VREF = VCC = 5 V, φAD = 10 MHz		2.8			µs
t_{SAMP}	Sampling time				0.3			µs
V_{REF}	Reference voltage				2.0		V_{CC}	V
V_{IA}	Analog input voltage				0		V_{REF}	V

NOTES:

1. Referenced to VCC = AVCC = VREF = 3.3 to 5.5 V, VSS = AVSS = 0 V, –40 to 85°C unless otherwise specified.
2. φAD frequency must be 10 MHz or less.
3. When sample & hold is disabled, φAD frequency must be 250 kHz or more in addition to a limit of NOTE 2.
   When sample & hold is enabled, φAD frequency must be 1 MHz or more in addition to a limit of NOTE 2.

Table 22.7 D/A conversion Characteristics [1]

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
t_{su}	Setup time				3	µs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference power supply input current	(NOTE 2)			1.5	mA

NOTES:

1. Referenced to VCC = AVCC = VREF = 3.3 to 5.5 V, VSS = AVSS = 0 V, –40 to 85°C unless otherwise specified.
2. This applies when using one D/A converter, with the DAi register (i = 0, 1) for the unused D/A converter set to 00h. The resistor ladder of the A/D converter is not included. Also, the I_{VREF} will flow even if VREF is disconnected by the ADCON1 register.

Table 22.8 Flash Memory Version Electrical Characteristics [1]

Symbol	Parameter		Standard Min.	Standard Typ.	Standard Max.	Unit
-	Programming and erasure endurance [2]		100			cycle
-	Word program time (VCC = 5.0 V)			25	200	µs
-	Lock bit program time			25	200	µs
-	Block erase time (VCC = 5.0 V)	4-Kbyte block		0.3	4	s
		8-Kbyte block		0.3	4	s
		32-Kbyte block		0.5	4	s
		64-Kbyte block		0.8	4	s
-	Erase all unlocked blocks time				4 × n [3]	s
tps	Flash memory circuit stabilization wait time				15	µs

NOTES:

1. Referenced to VCC = 4.5 to 5.5 V, 3.0 to 3.6 V, Topr = 0 to 60°C unless otherwise specified.
2. Programming and erasure endurance refers to the number of times a block erase can be performed.
   If the programming and erasure endurance is n (n = 100), each block can be erased n times.
   For example, if a 4-Kbyte block A is erased after writing 1 word data 2,048 times, each to a different address, this counts as one programming and erasure endurance. Data cannot be written to the same address more than once without erasing the block (rewrite prohibited).
3. n denotes the number of blocks to erase.

Table 22.9 Flash Memory Version Program/Erase Voltage and Read Operation Voltage Characteristics (at Topr = 0 to 60°C)

Flash Program, Erase Voltage	Flash Read Operation Voltage
VCC = 3.3 ± 0.3 V or 5.0 ± 0.5 V	VCC = 3.0 to 5.5 V

Table 22.10 Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measuring Condition	Standard Min.	Standard Typ.	Standard Max.	Unit
$t_{d(P-R)}$	Time for internal power supply stabilization during powering-on	VCC = 3.0 to 5.5 V			2	ms
$t_{d(R-S)}$	STOP release time				150	µs
$t_{d(W-S)}$	Low power dissipation mode wait mode release time				150	µs

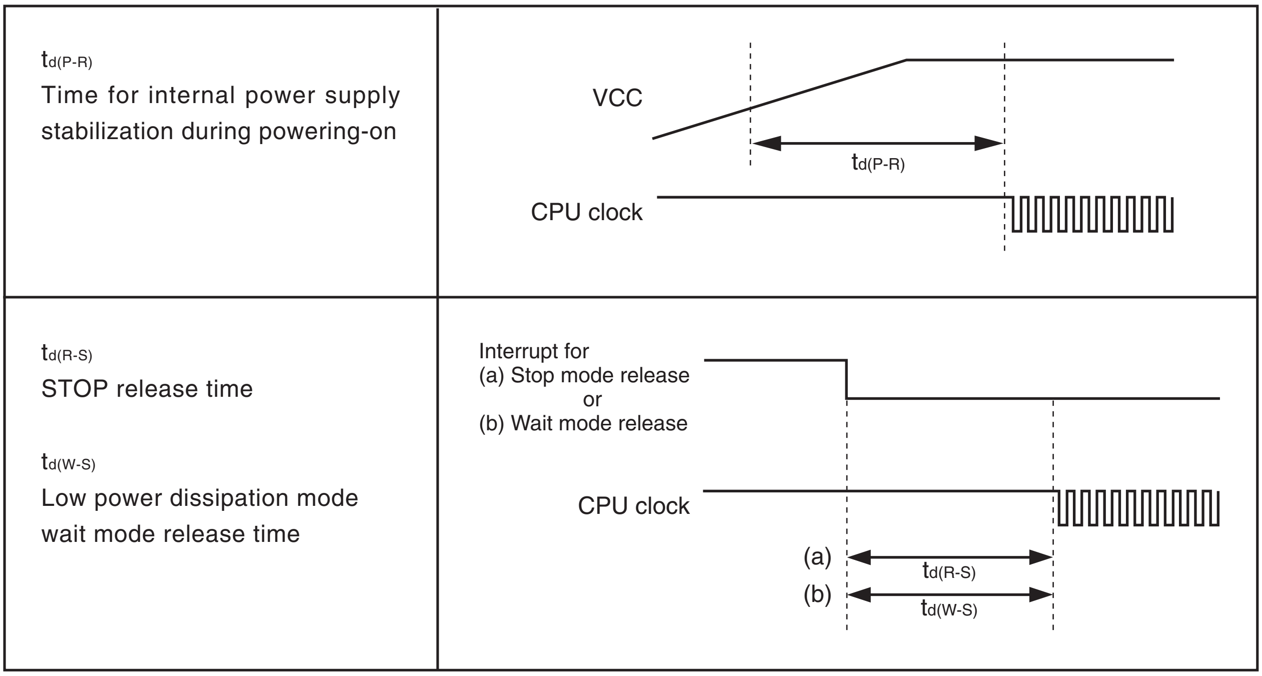


Figure 22.2 Power Supply Circuit Timing Diagram

Timing Requirements

VCC = 5 V

(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)

Table 22.11 External Clock Input (XIN Input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_C	External clock input cycle time	62.5		ns
$t_{w(H)}$	External clock input HIGH pulse width	25		ns
$t_{w(L)}$	External clock input LOW pulse width	25		ns
t_r	External clock rise time		15	ns
t_f	External clock fall time		15	ns

Table 22.12 Memory Expansion Mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (for setting with no wait)		(NOTE 1)	ns
$t_{ac2(RD-DB)}$	Data input access time (for setting with wait)		(NOTE 2)	ns
$t_{ac3(RD-DB)}$	Data input access time (when accessing multiplexed bus area)		(NOTE 3)	ns
$t_{su(DB-RD)}$	Data input setup time	40		ns
$t_{su(RDY-BCLK)}$	$\overline{\text{RDY}}$ input setup time	30		ns
$t_{su(HOLD-BCLK)}$	$\overline{\text{HOLD}}$ input setup time	40		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{\text{RDY}}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{\text{HOLD}}$ input hold time	0		ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 45 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 45 \text{ [ns]}$$ n is “2” for 1-wait setting, “3” for 2-wait setting and “4” for 3-wait setting.

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 45 \text{ [ns]}$$ n is “2” for 2-wait setting, “3” for 3-wait setting.

Timing Requirements **VCC = 5 V**

(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)

Table 22.13 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	100		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	40		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	40		ns

Table 22.14 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	400		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	200		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	200		ns

Table 22.15 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	200		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	100		ns

Table 22.16 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	100		ns

Table 22.17 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	400		ns

Table 22.18 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	800		ns
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	200		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	200		ns

Timing Requirements **VCC = 5 V**

(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)

Table 22.19 Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	80		ns

Table 22.20 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 22.21 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 22.22 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{C(AD)}$	$\overline{\text{ADTRG}}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ input LOW pulse width	125		ns

Table 22.23 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TXDi output delay time		80	ns
$t_{h(C-Q)}$	TXDi hold time	0		ns
$t_{su(D-C)}$	RXDi input setup time	70		ns
$t_{h(C-D)}$	RXDi input hold time	90		ns

Table 22.24 External Interrupt $\overline{\text{INTi}}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input LOW pulse width	250		ns

Switching Characteristics **VCC = 5 V**

(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)

Table 22.25 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 22.3		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (in relation to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (in relation to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (rin relation to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			15	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (in relation to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (in relation to BCLK) (3)		4		ns
$t_{d(DB-WR)}$	Data output delay time (in relation to WR)		(NOTE 2)		ns
$t_{h(WR-DB)}$	Data output hold time (rin relation to WR) (3)		(NOTE 1)		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time			40	ns

NOTES:

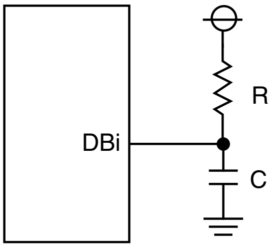
1. Calculated according to the BCLK frequency as follows:

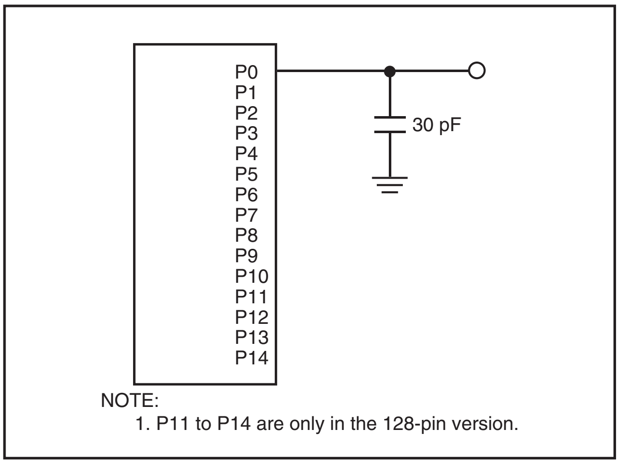
$$\frac{0.5 \times 10^9}{f(BCLK)} - 10\ [ns]$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40\ [ns] \qquad f(BCLK) \text{ is 12.5 MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in
$t = -CR \times \ln(1 - V_{OL} / V_{CC})$
by a circuit of the right figure.
For example, when $V_{OL} = 0.2\,V_{CC}$, C = 30 pF, R =1 kΩ, hold time of output "L" level is
$t = -30\ pF \times 1\ k\Omega \times \ln(1 - 0.2\,V_{CC} / V_{CC}) = 6.7\ ns$.





NOTE:
1. P11 to P14 are only in the 128-pin version.

Figure 22.3 Port P0 to P14 Measurement Circuit

Switching Characteristics

VCC = 5 V

(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)

Table 22.26 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 22.3		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (in relation to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (in relation to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (in relation to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			15	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (in relation to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (rin relation to BCLK) [3]		4		ns
$t_{d(DB-WR)}$	Data output delay time (in relation to WR)		(NOTE 2)		ns
$t_{h(WR-DB)}$	Data output hold time (in relation to WR) [3]		(NOTE 1)		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time			40	ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

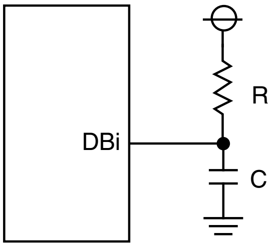
$$\frac{0.5 \times 10^9}{f(BCLK)} - 10\ [ns]$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 40\ [ns]$$

n is “1” for 1-wait setting, “2” for 2-wait setting and “3” for 3-wait setting.
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in
$t = – CR \times \ln (1 – V_{OL} / V_{CC})$
by a circuit of the right figure.
For example, when $V_{OL} = 0.2\ V_{CC}$, C = 30 pF, R =1 kΩ, hold time of output “L” level is
$t = – 30\ pF \times 1\ k\Omega \times \ln (1 – 0.2\ V_{CC} / V_{CC}) = 6.7\ ns.$



Switching Characteristics **VCC = 5 V**

(Referenced to VCC = 5 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)

Table 22.27 Memory Expansion Mode and Microprocessor Mode (for 2- to 3-wait setting, external area access and multiplexed bus selection)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 22.3		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (in relation to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (in relation to RD)		(NOTE 1)		ns
$t_{h(WR-AD)}$	Address output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (in relation to BCLK)		4		ns
$t_{h(RD-CS)}$	Chip select output hold time (in relation to RD)		(NOTE 1)		ns
$t_{h(WR-CS)}$	Chip select output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (in relation to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (in relation to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (in relation to WR)		(NOTE 2)		ns
$t_{h(WR-DB)}$	Data output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time			40	ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (in relation to BCLK)			15	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (in relation to BCLK)		–4		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (in relation to Address)		(NOTE 3)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (in relation to Address)		(NOTE 4)		ns
$t_{d(AD-RD)}$	RD signal output delay from the end of Address		0		ns
$t_{d(AD-WR)}$	WR signal output delay from the end of Address		0		ns
$t_{dZ(RD-AD)}$	Address output floating start time			8	ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 40 \text{ [ns]}$$ n is “2” for 2-wait setting, “3” for 3-wait setting.

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 25 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 15 \text{ [ns]}$$

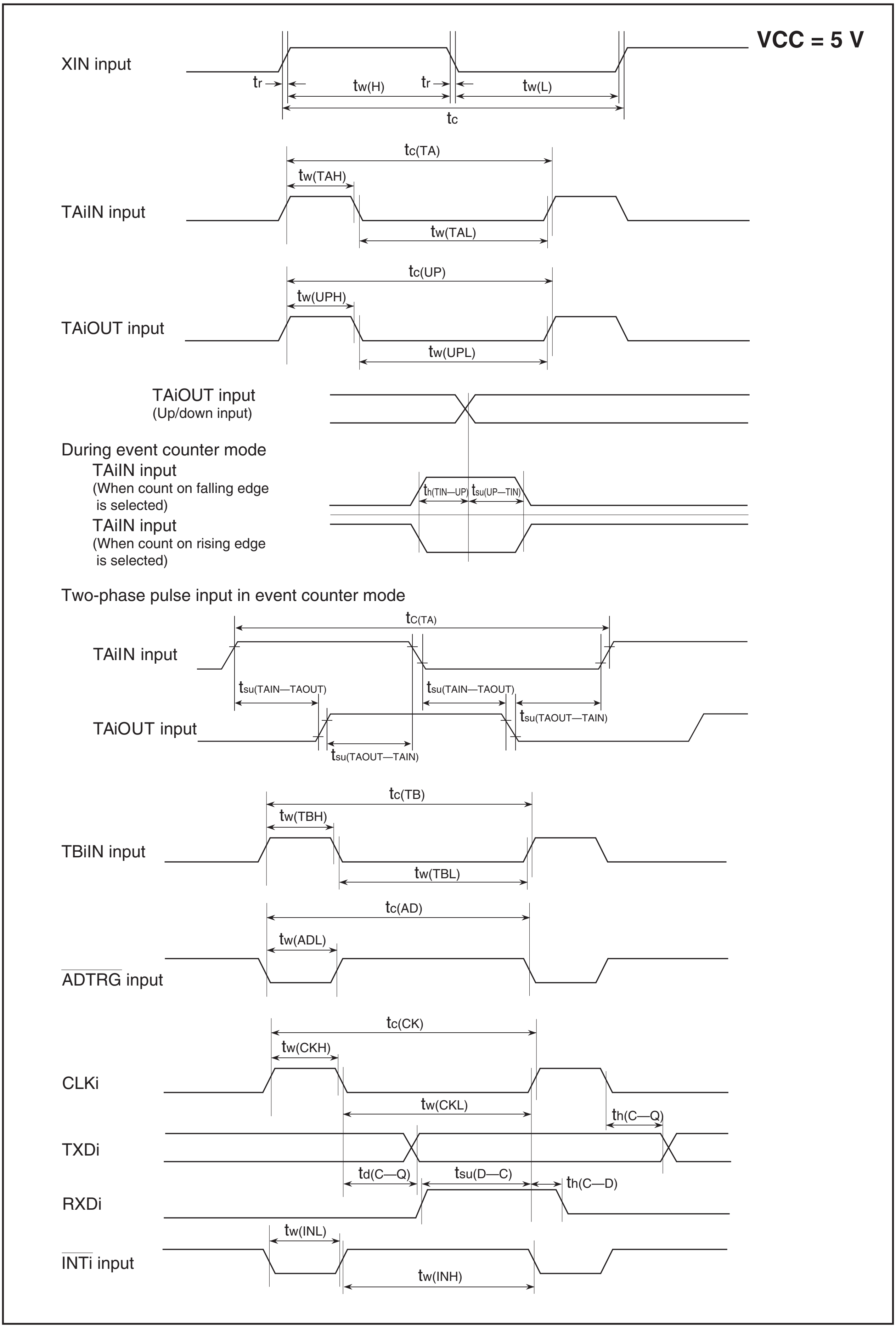
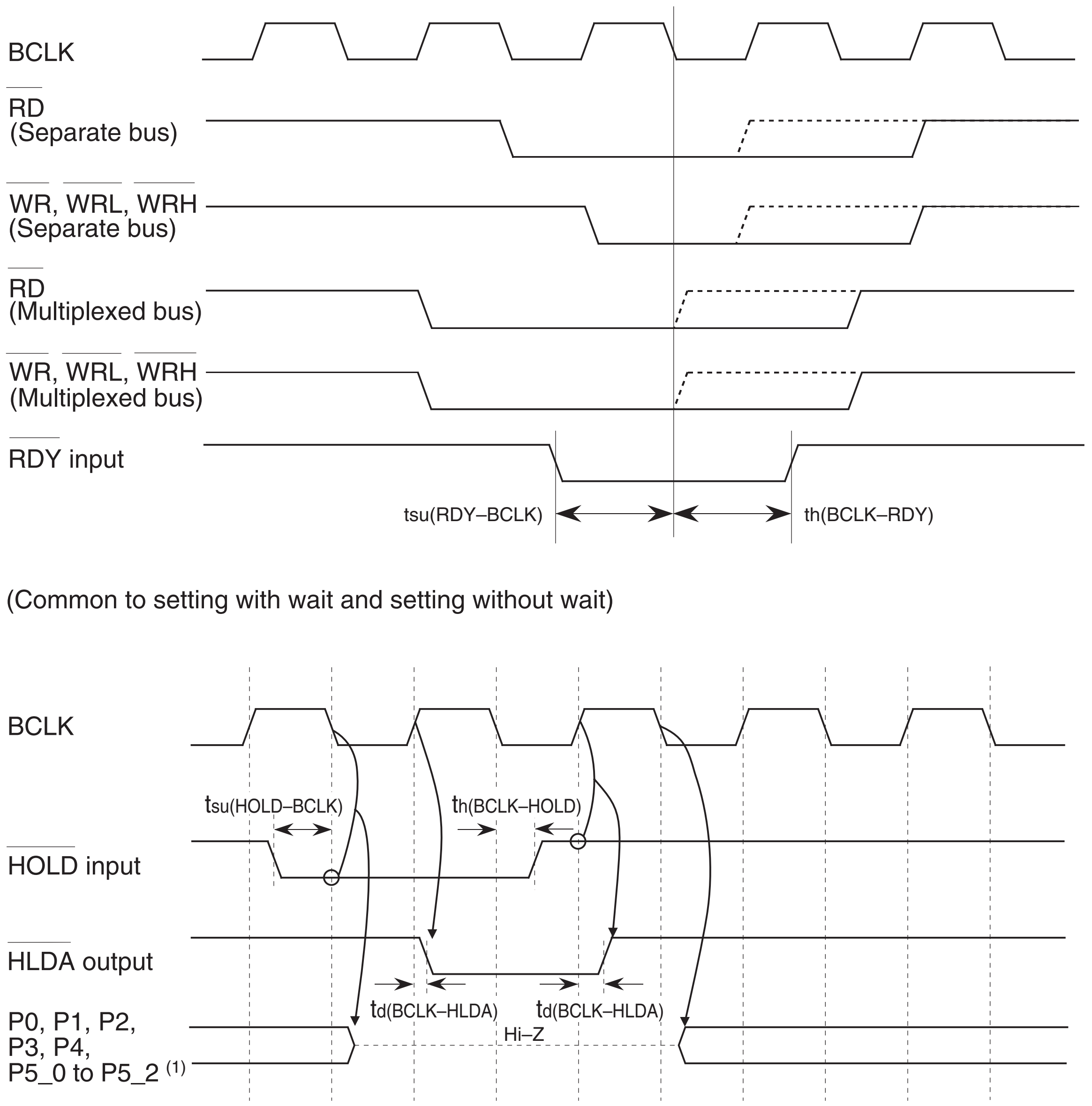


Figure 22.4 Timing Diagram (1)

Memory Expansion Mode and Microprocessor Mode **VCC = 5 V**

(Effective for setting with wait)



NOTE:

1. The above pins are set to high-impedance regardless of the input level of the BYTE pin, the PM06 bit in the PM0 register, and the PM11 bit in the PM1 register.

Measuring conditions :

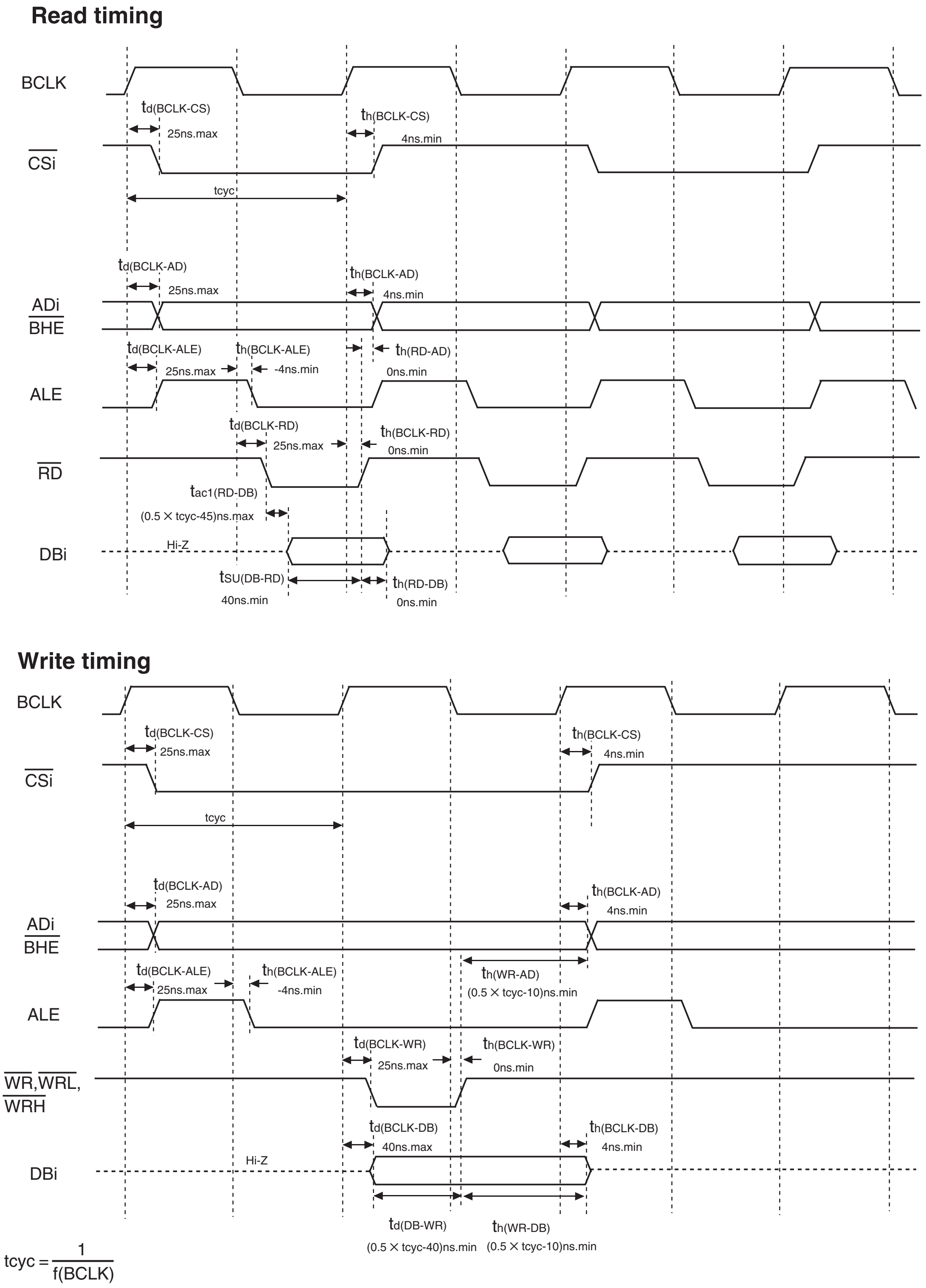
- VCC = 5 V
- Input timing voltage : Determined with V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage: Determined with V_{OL} = 2.5 V, V_{OH} = 2.5 V

Figure 22.5 Timing Diagram (2)

Memory Expansion Mode and Microprocessor Mode

VCC = 5 V

(For setting with no wait)



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.0 V
- Output timing voltage : V_{OL} = 0.4 V, V_{OH} = 2.4 V

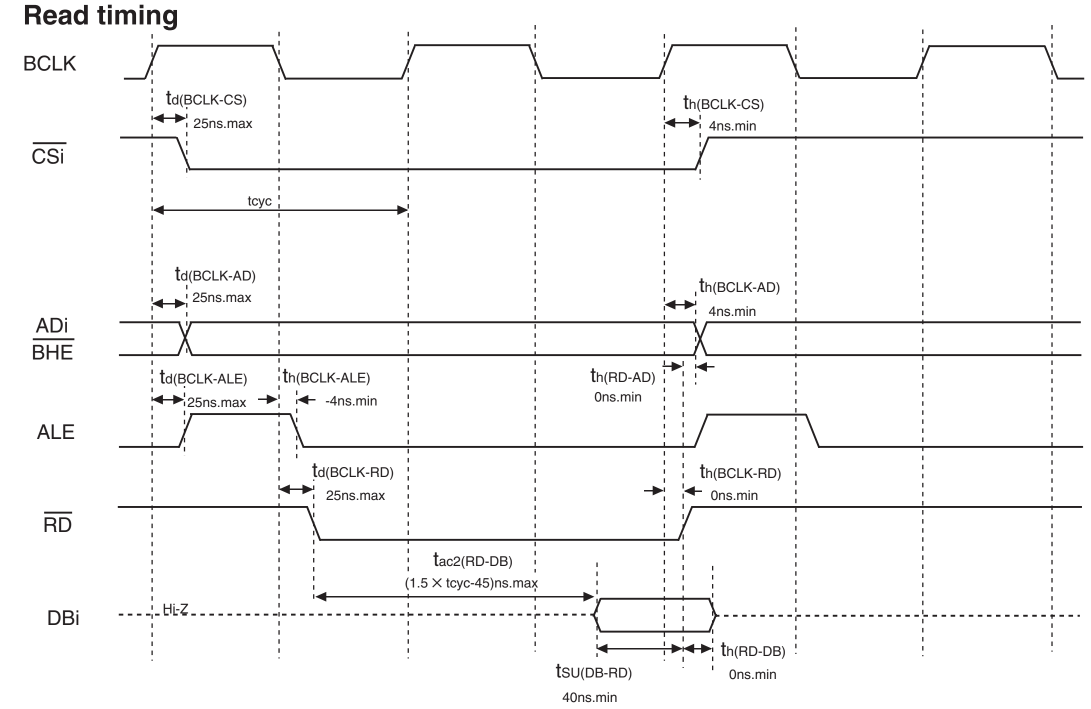
Figure 22.6 Timing Diagram (3)

Memory Expansion Mode and Microprocessor Mode

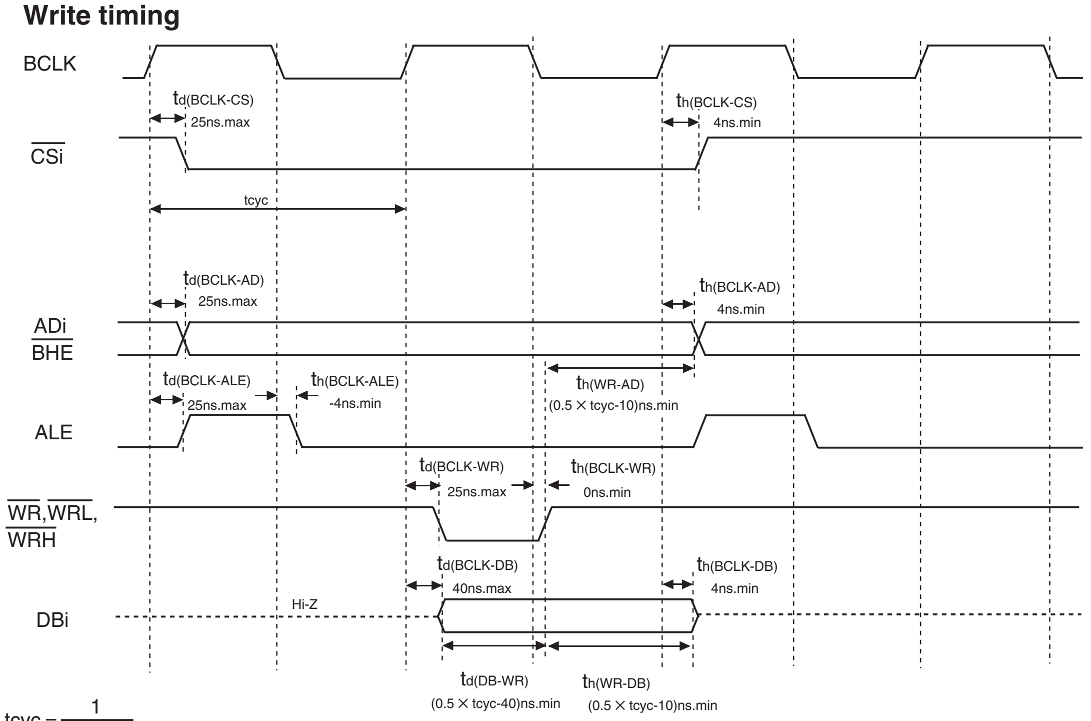
VCC = 5 V

(For 1-wait setting and external area access)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.0 V
- Output timing voltage : V_{OL} = 0.4 V, V_{OH} = 2.4 V

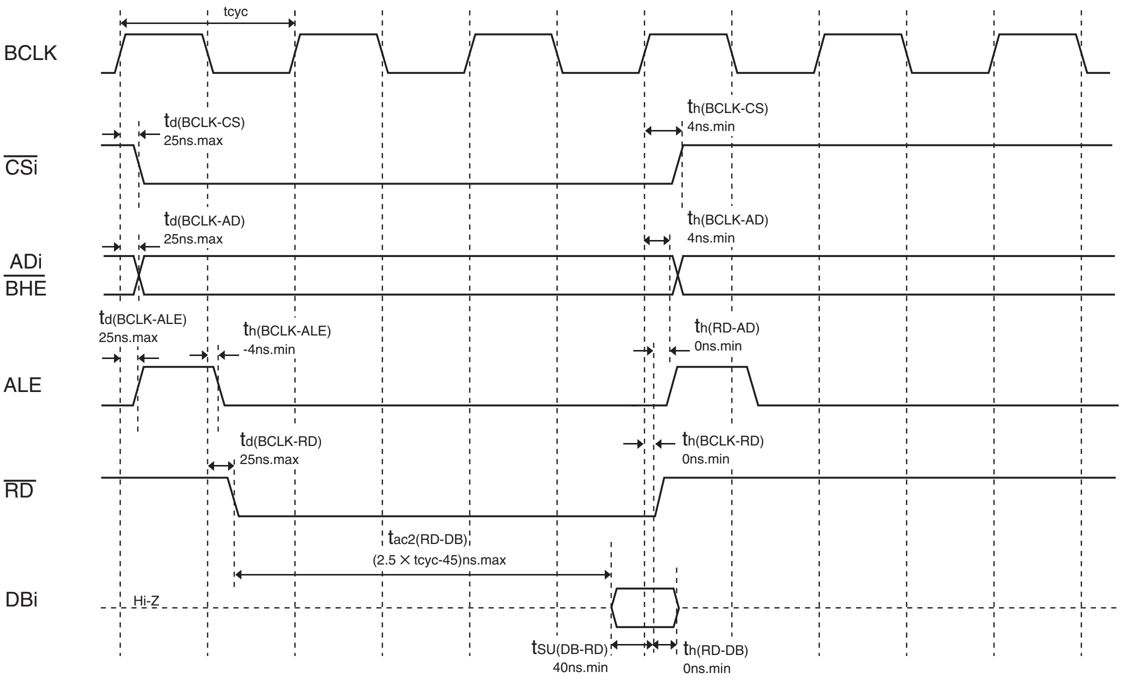
Figure 22.7 Timing Diagram (4)

Memory Expansion Mode and Microprocessor Mode

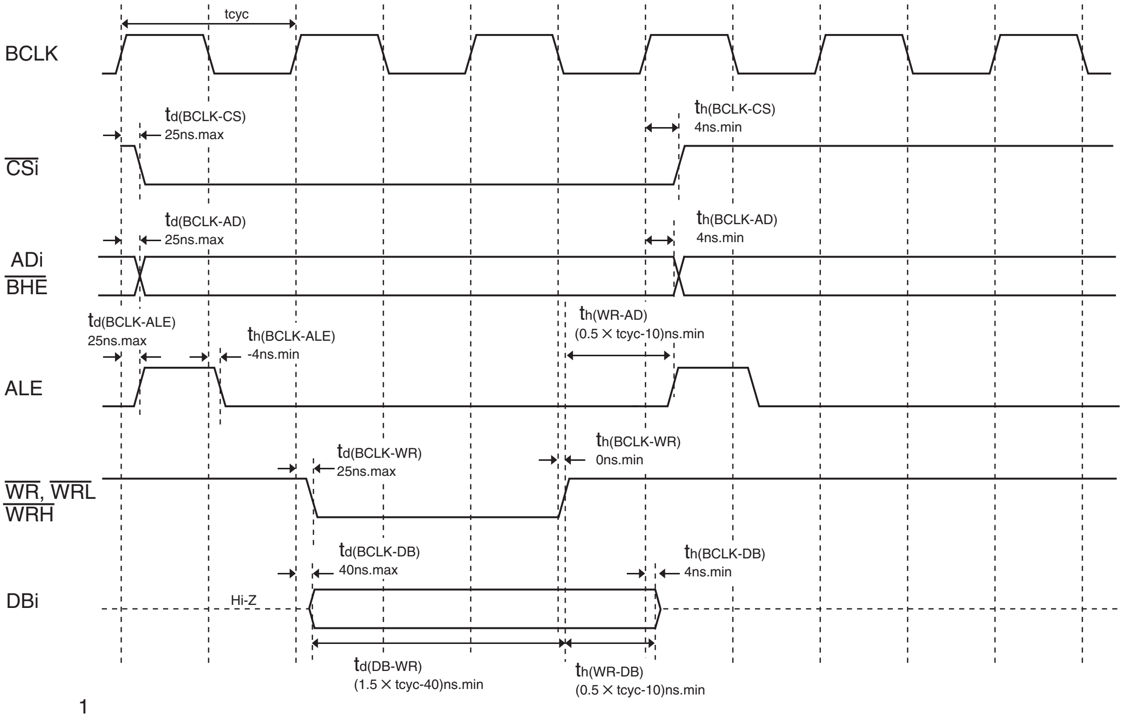
VCC = 5 V

(For 2-wait setting and external area access)

Read timing



Write timing



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

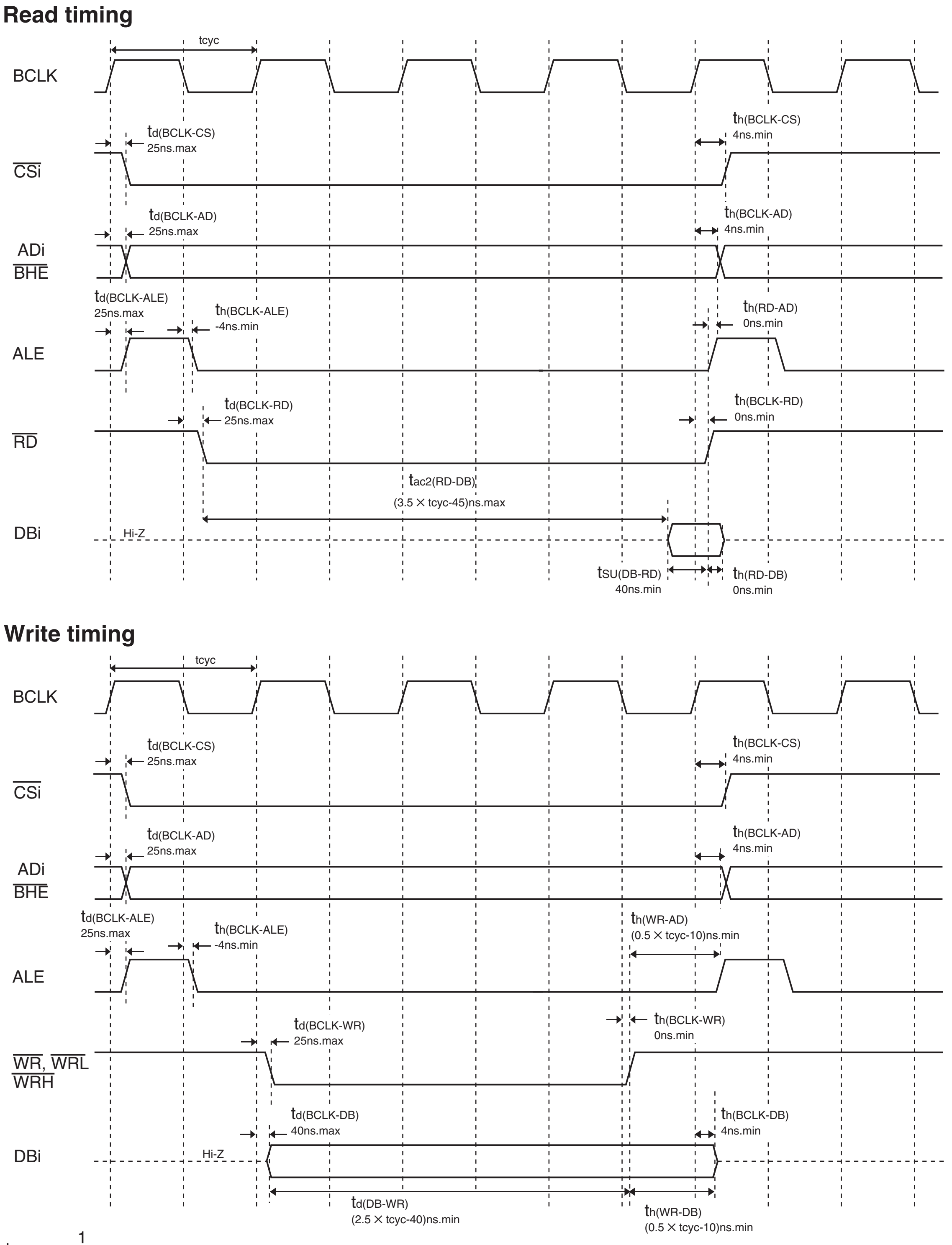
- VCC = 5 V
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.0 V
- Output timing voltage : V_{OL} = 0.4 V, V_{OH} = 2.4 V

Figure 22.8 Timing Diagram (5)

Memory Expansion Mode and Microprocessor Mode

(For 3-wait setting and external area access)

VCC = 5 V



Read timing

Write timing

$$tcyc = \frac{1}{f(BCLK)}$$

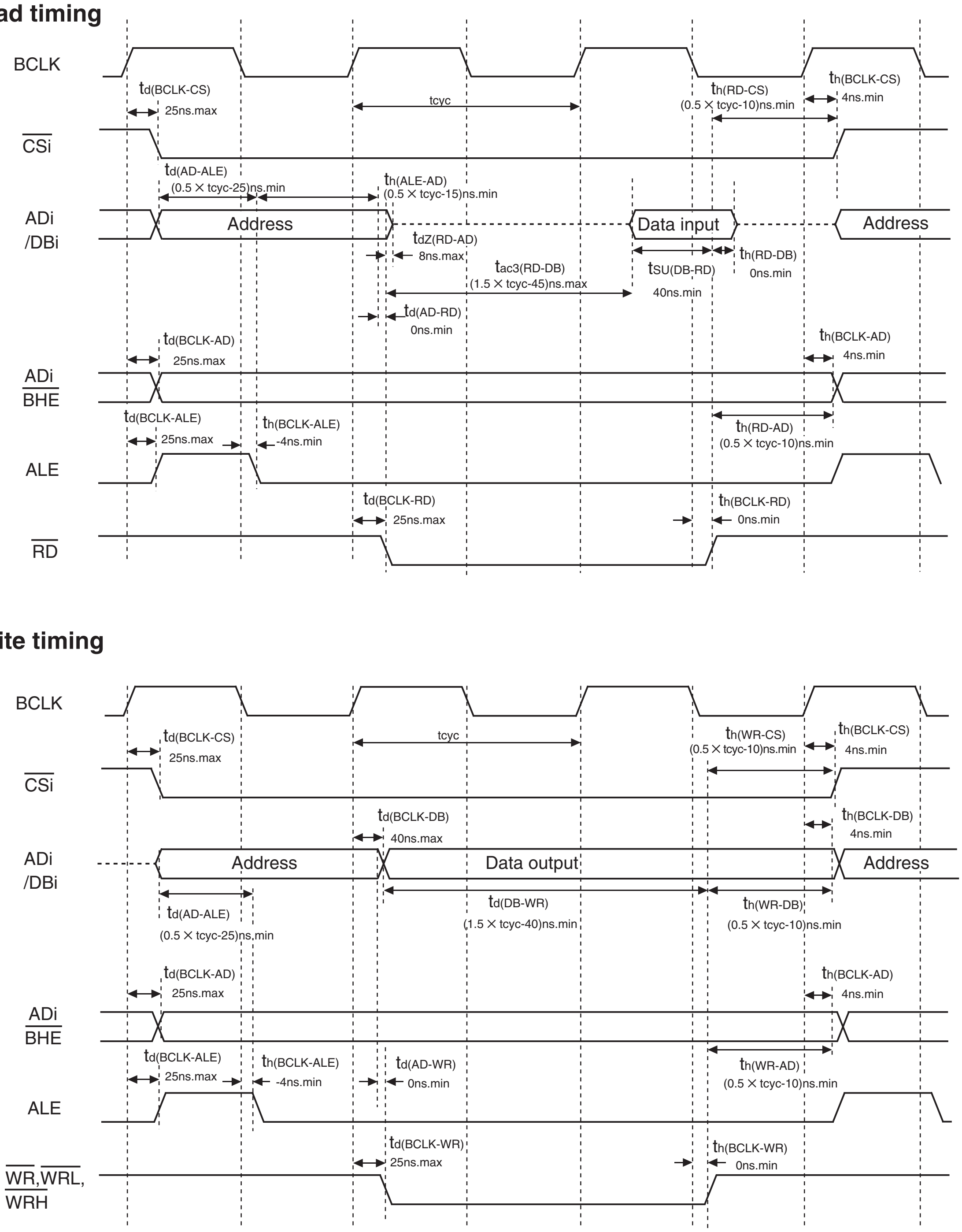
Measuring conditions :
- VCC = 5 V
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.0 V
- Output timing voltage : V_{OL} = 0.4 V, V_{OH} = 2.4 V

Figure 22.9 Timing Diagram (6)

Memory Expansion Mode and Microprocessor Mode VCC = 5 V

(For 1- or 2-wait setting, external area access and multiplexed bus selection)

Read timing



Write timing

$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

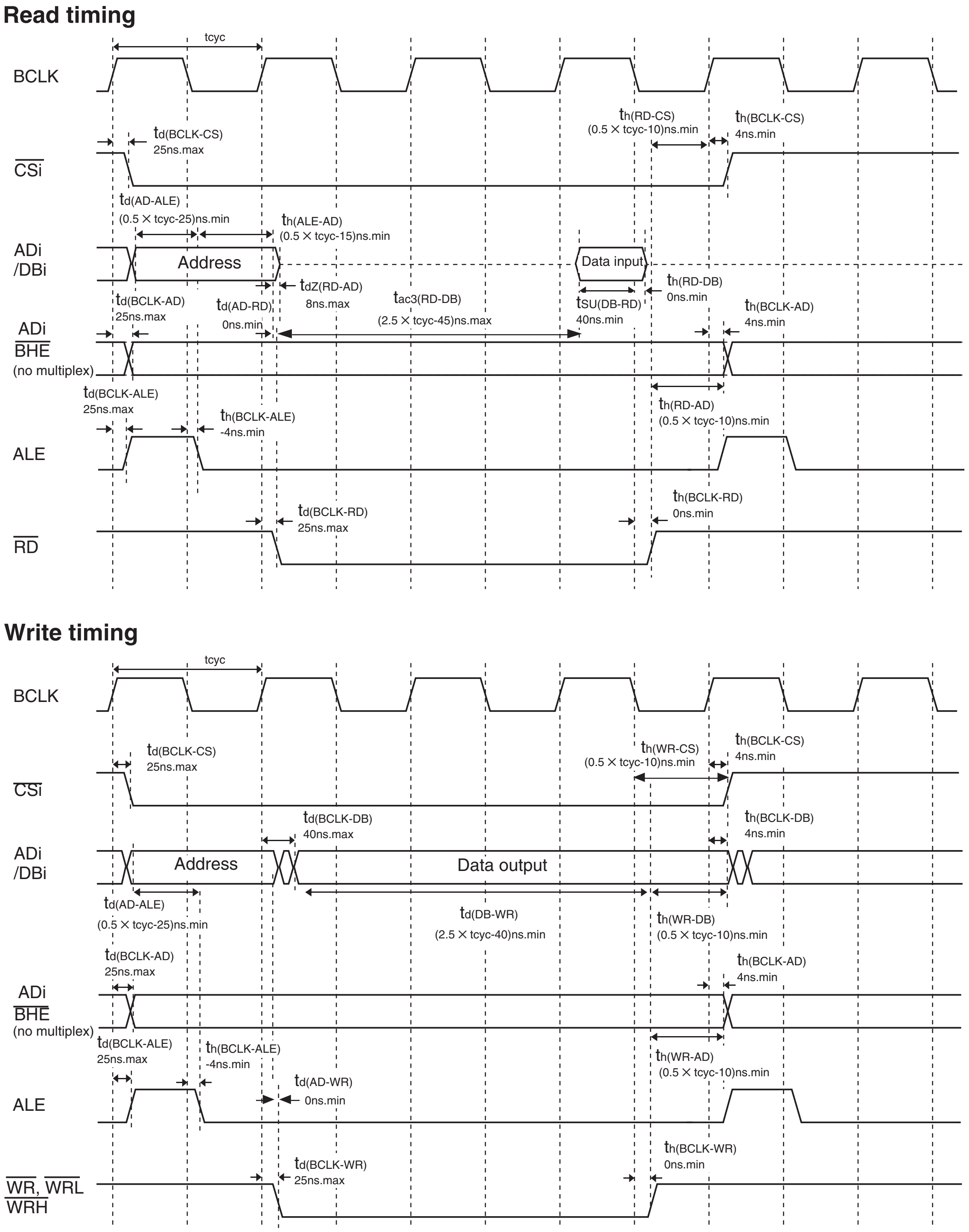
- VCC = 5 V
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.0 V
- Output timing voltage : V_{OL} = 0.4 V, V_{OH} = 2.4 V

Figure 22.10 Timing Diagram (7)

Memory Expansion Mode and Microprocessor Mode

VCC = 5 V

(For 3-wait setting, external area access and multiplexed bus selection)


Read timing

Write timing

$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 5 V
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.0 V
- Output timing voltage : V_{OL} = 0.4 V, V_{OH} = 2.4 V

Figure 22.11 Timing Diagram (8)

Table 22.28 Electrical Characteristics [1] **VCC = 3.3 V**

Symbol	Parameter			Measuring Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7,P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		I_{OH} = –1 mA	V_{CC}-0.5		V_{CC}	V
V_{OH}	HIGH output voltage	XOUT	HIGHPOWER	I_{OH} = –0.1 mA	V_{CC}-0.5		V_{CC}	V
			LOWPOWER	I_{OH} = –50 μA	V_{CC}-0.5		V_{CC}	
	HIGH output voltage	XCOUT	HIGHPOWER	With no load applied		2.5		V
			LOWPOWER	With no load applied		1.6		
V_{OL}	LOW output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7,P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		I_{OL} = 1 mA			0.5	V
V_{OL}	LOW output voltage	XOUT	HIGHPOWER	I_{OL} = 0.1 mA			0.5	V
			LOWPOWER	I_{OL} = 50 μA			0.5	
	LOW output voltage	XCOUT	HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, TA0IN to TA4IN, TB0IN to TB5IN, $\overline{\text{INT0}}$ to $\overline{\text{INT8}}$, $\overline{\text{NMI}}$, $\overline{\text{ADTRG}}$, $\overline{\text{CTS0}}$ to $\overline{\text{CTS2}}$, SCL0 to SCL2, SDA0 to SDA2, CLK0 to CLK6, TA0OUT to TA4OUT, $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$, RXD0 to RXD2, SIN3 to SIN6			0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis	$\overline{\text{RESET}}$			0.2		1.8	V
I_{IH}	HIGH input current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, $\overline{\text{RESET}}$, CNVSS, BYTE		V_I = 3.3 V			4.0	μA
I_{IL}	LOW input current	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1, XIN, $\overline{\text{RESET}}$, CNVSS, BYTE		V_I = 0 V			–4.0	μA
R_{PULLUP}	Pull-up resistance	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0, P9_2 to P9_7, P10_0 to P10_7, P11_0 to P11_7, P12_0 to P12_7, P13_0 to P13_7, P14_0, P14_1		V_I = 0 V	50	100	500	kΩ
R_{fXIN}	Feedback resistance		XIN			3.0		MΩ
R_{fXCIN}	Feedback resistance		XCIN			25		MΩ
V_{RAM}	RAM retention voltage			At stop mode	2.0			V

NOTES:

1. Referenced to VCC = 3.0 to 3.6 V, VSS = 0 V at Topr = –40 to 85°C, f(BCLK) = 24 MHz unless otherwise specified.
2. P11 to P14, $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$, CLK5, CLK6, SIN5, and SIN6 are only in the 128-pin version.

Timing Requirements **VCC = 3.3 V**

(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)

Table 22.29 External Clock Input (XIN Input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_C	External clock input cycle time	62.5		ns
$t_{w(H)}$	External clock input HIGH pulse width	25		ns
$t_{w(L)}$	External clock input LOW pulse width	25		ns
t_r	External clock rise time		15	ns
t_f	External clock fall time		15	ns

Table 22.30 Memory Expansion Mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (for setting with no wait)		(NOTE 1)	ns
$t_{ac2(RD-DB)}$	Data input access time (for setting with wait)		(NOTE 2)	ns
$t_{ac3(RD-DB)}$	Data input access time (when accessing multiplexed bus area)		(NOTE 3)	ns
$t_{su(DB-RD)}$	Data input setup time	50		ns
$t_{su(RDY-BCLK)}$	$\overline{\text{RDY}}$ input setup time	40		ns
$t_{su(HOLD-BCLK)}$	$\overline{\text{HOLD}}$ input setup time	50		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{\text{RDY}}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{\text{HOLD}}$ input hold time	0		ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 60 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 60 \text{ [ns]}$$

n is “2” for 1-wait setting, “3” for 2-wait setting and “4” for 3-wait setting.

3. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 60 \text{ [ns]}$$

n is “2” for 2-wait setting, “3” for 3-wait setting.

Timing Requirements

VCC = 3.3 V

(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)

Table 22.31 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	60		ns

Table 22.32 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	300		ns

Table 22.33 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 22.34 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 22.35 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

Table 22.36 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	2		μs
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	500		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	500		ns

Timing Requirements **VCC = 3.3 V**

(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = –40 to 85°C unless otherwise specified)

Table 22.37 Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	120		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	120		ns

Table 22.38 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table 22.39 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table 22.40 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{C(AD)}$	$\overline{\text{ADTRG}}$ input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ input LOW pulse width	200		ns

Table 22.41 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TXDi output delay time		160	ns
$t_{h(C-Q)}$	TXDi hold time	0		ns
$t_{su(D-C)}$	RXDi input setup time	100		ns
$t_{h(C-D)}$	RXDi input hold time	90		ns

Table 22.42 External Interrupt $\overline{\text{INTi}}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input LOW pulse width	380		ns

Switching Characteristics **VCC = 3.3 V**

(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)

Table 22.43 Memory Expansion Mode and Microprocessor Mode (for setting with no wait)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 22.12		30	ns
$t_{h(BCLK-AD)}$	Address output hold time (in relation to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (in relation to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			30	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (in relation to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			30	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			30	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (in relation to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (in relation to BCLK) [3]		4		ns
$t_{d(DB-WR)}$	Data output delay time (in relation to WR)		(NOTE 2)		ns
$t_{h(WR-DB)}$	Data output hold time (in relation to WR) [3]		(NOTE 1)		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time			40	ns

NOTES:

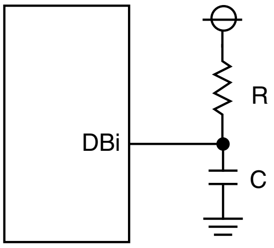
1. Calculated according to the BCLK frequency as follows:

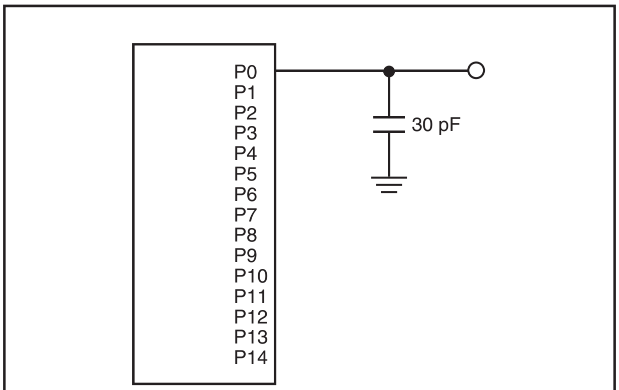
$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \text{ [ns]} \qquad f(BCLK) \text{ is 12.5 MHz or less.}$$

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in
$t = -CR \times \ln(1 - V_{OL} / V_{CC})$
by a circuit of the right figure.
For example, when $V_{OL} = 0.2 V_{CC}$, C = 30 pF, R =1 kΩ, hold time of output "L" level is
$t = -30 \text{ pF} \times 1 \text{ k}\Omega \times \ln(1 - 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns}$.





NOTE:
1. P11 to P14 are only in the 128-pin version.

Figure 22.12 Port P0 to P14 Measurement Circuit

Switching Characteristics **VCC = 3.3 V**

(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)

Table 22.44 Memory Expansion Mode and Microprocessor Mode (for 1- to 3-wait setting and external area access)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 22.12		30	ns
$t_{h(BCLK-AD)}$	Address output hold time (in relation to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (in relation to RD)		0		ns
$t_{h(WR-AD)}$	Address output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			30	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (in relation to BCLK)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		–4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			30	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			30	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (in relation to BCLK)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (in relation to BCLK) (3)		4		ns
$t_{d(DB-WR)}$	Data output delay time (in relation to WR)		(NOTE 2)		ns
$t_{h(WR-DB)}$	Data output hold time (in relation to WR) (3)		(NOTE 1)		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time			40	ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

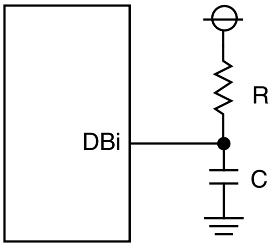
$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 40 \text{ [ns]}$$

n is “1” for 1-wait setting, “2” for 2-wait setting and “3” for 3-wait setting.
When n = 1, f(BCLK) is 12.5 MHz or less.

3. This standard value shows the timing when the output is off, and does not show hold time of data bus.
Hold time of data bus varies with capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in
$t = – CR \times \ln (1 – V_{OL} / V_{CC})$
by a circuit of the right figure.
For example, when $V_{OL} = 0.2 V_{CC}$, C = 30 pF, R =1 kΩ, hold time of output “L” level is
$t = – 30 \text{ pF} \times 1 \text{ k}\Omega \times \ln (1 – 0.2 V_{CC} / V_{CC}) = 6.7 \text{ ns}$.



Switching Characteristics **VCC = 3.3 V**

(Referenced to VCC = 3.3 V, VSS = 0 V, at Topr = –40 to 85 °C unless otherwise specified)

Table 22.45 Memory Expansion Mode and Microprocessor Mode (for 2- to 3-wait setting, external area access and multiplexed bus selection)

Symbol	Parameter	Measuring Condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 22.12		50	ns
$t_{h(BCLK-AD)}$	Address output hold time (in relation to BCLK)		4		ns
$t_{h(RD-AD)}$	Address output hold time (in relation to RD)		(NOTE 1)		ns
$t_{h(WR-AD)}$	Address output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			50	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (in relation to BCLK)		4		ns
$t_{h(RD-CS)}$	Chip select output hold time (in relation to RD)		(NOTE 1)		ns
$t_{h(WR-CS)}$	Chip select output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			40	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			40	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (in relation to BCLK)			50	ns
$t_{h(BCLK-DB)}$	Data output hold time (in relation to BCLK)		4		ns
$t_{d(DB-WR)}$	Data output delay time (in relation to WR)		(NOTE 2)		ns
$t_{h(WR-DB)}$	Data output hold time (in relation to WR)		(NOTE 1)		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time			40	ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (in relation to BCLK)			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (in relation to BCLK)		–4		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (in relation to Address)		(NOTE 3)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (rin relation to Address)		(NOTE 4)		ns
$t_{d(AD-RD)}$	RD signal output delay from the end of Address		0		ns
$t_{d(AD-WR)}$	WR signal output delay from the end of Address		0		ns
$t_{dZ(RD-AD)}$	Address output floating start time			8	ns

NOTES:

1. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 10 \text{ [ns]}$$

2. Calculated according to the BCLK frequency as follows:

$$\frac{(n - 0.5) \times 10^9}{f(BCLK)} - 50 \text{ [ns]}$$ n is “2” for 2-wait setting, “3” for 3-wait setting.

3. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 40 \text{ [ns]}$$

4. Calculated according to the BCLK frequency as follows:

$$\frac{0.5 \times 10^9}{f(BCLK)} - 15 \text{ [ns]}$$

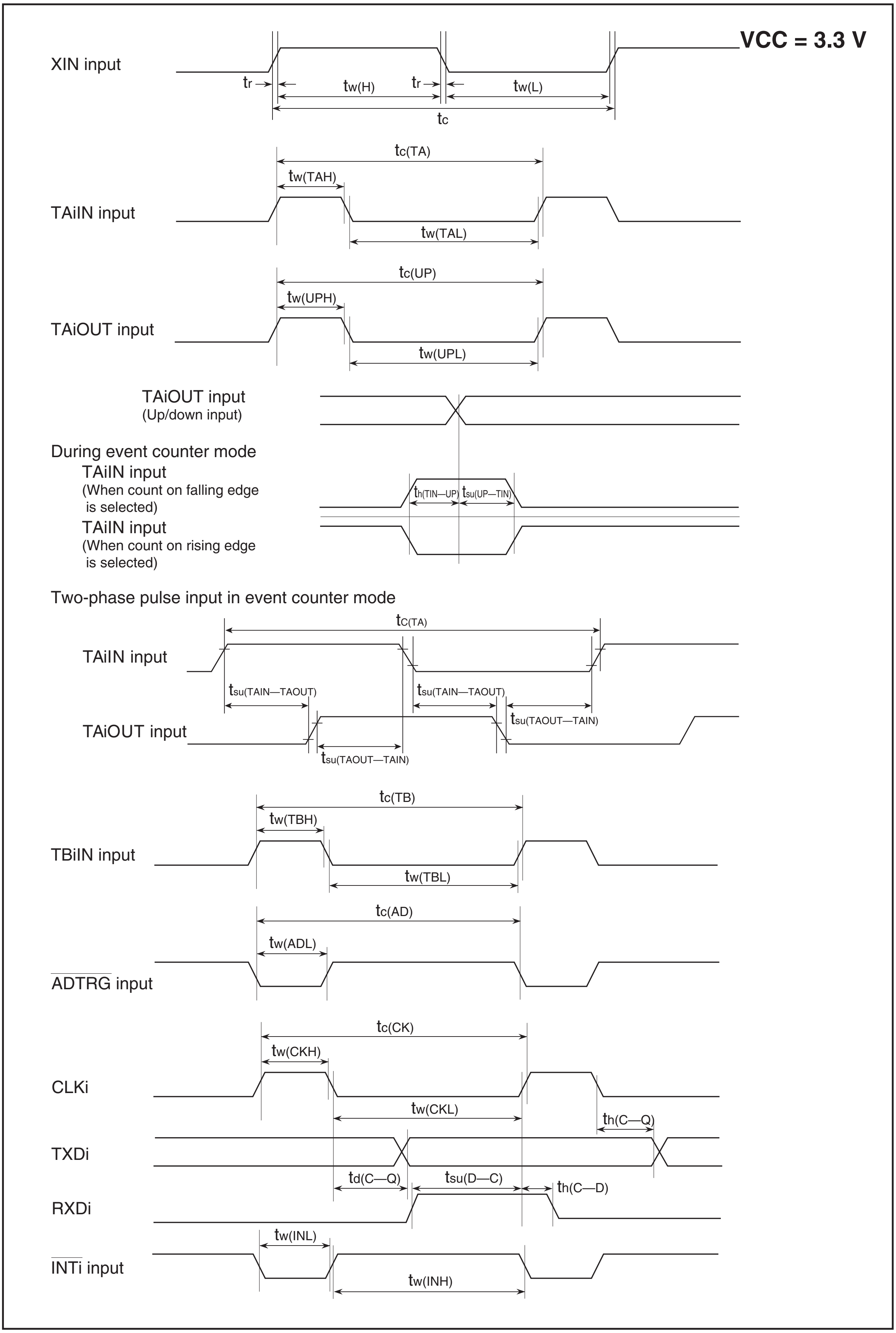
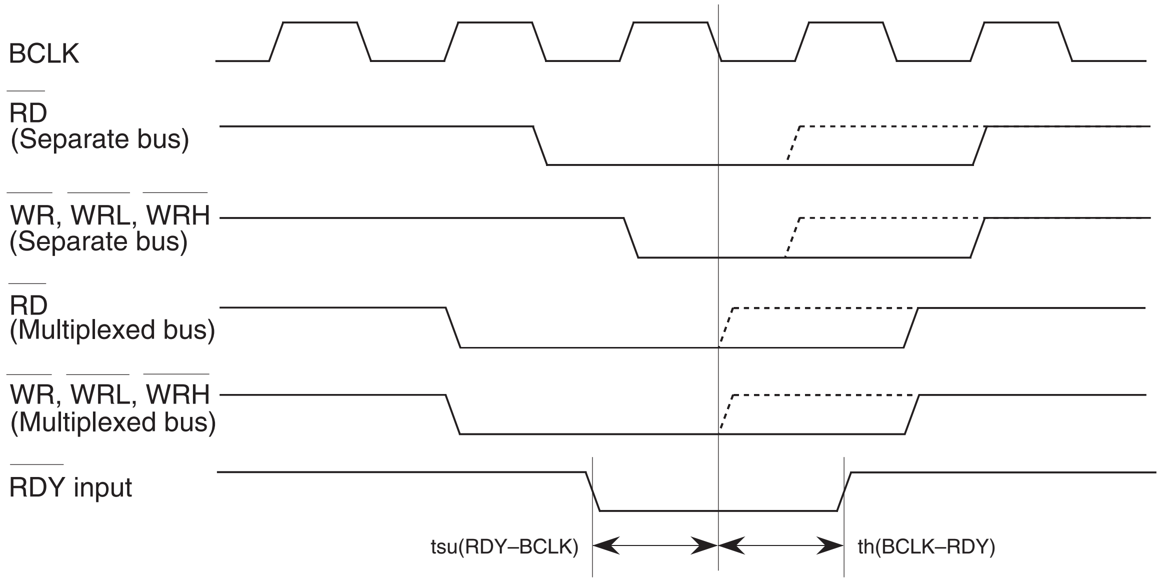


Figure 22.13 Timing Diagram (1)

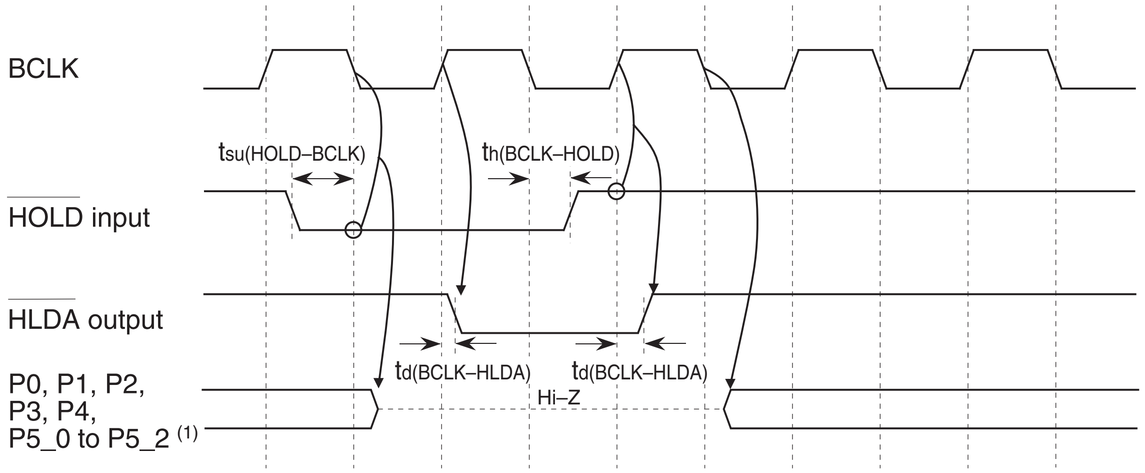
Memory Expansion Mode and Microprocessor Mode

VCC = 3.3 V

(Effective for setting with wait)



(Common to setting with wait and setting without wait)



NOTE:

1. The above pins are set to high-impedance regardless of the input level of the BYTE pin, the PM06 bit in the PM0 register, and the PM11 bit in the PM1 register.

Measuring conditions :

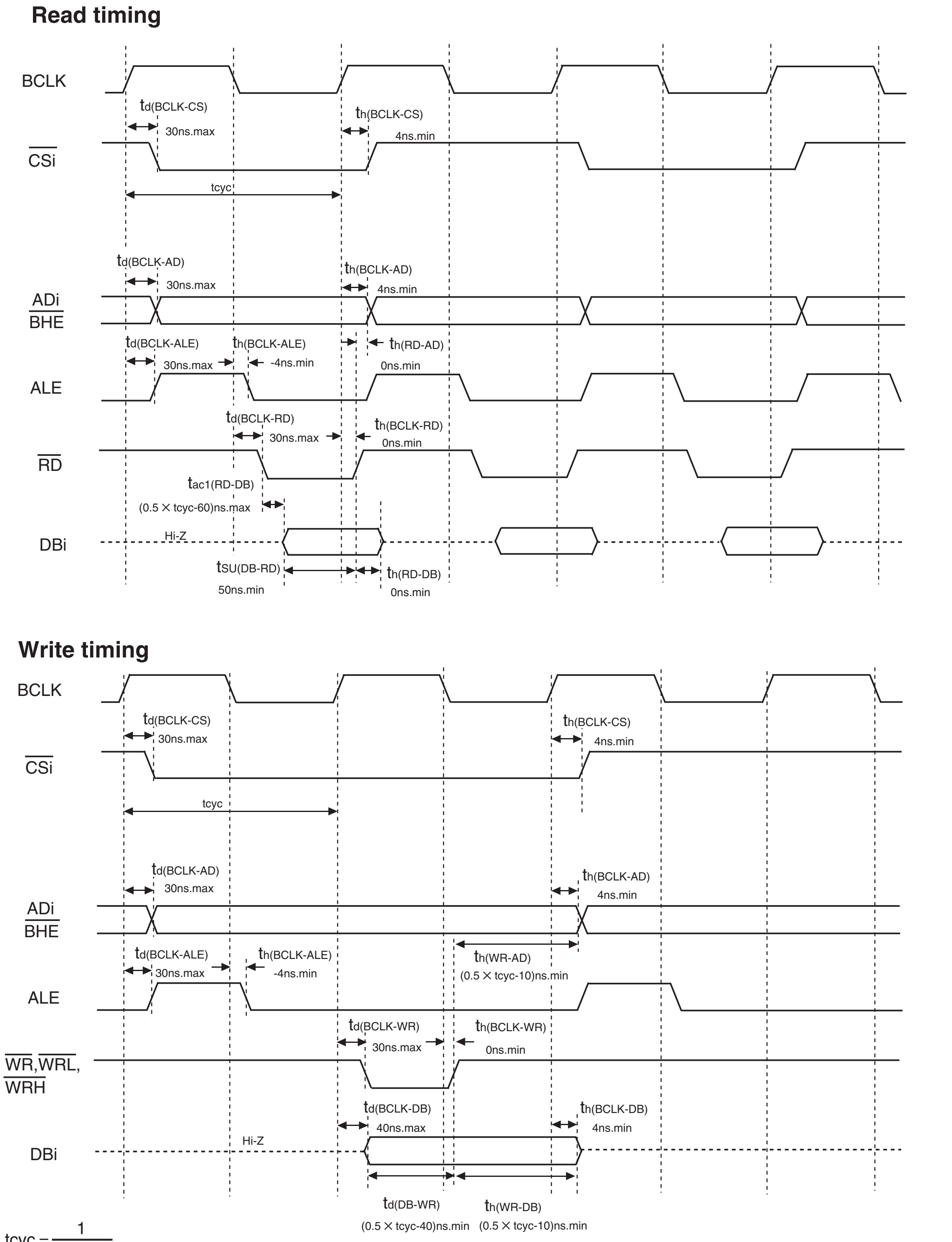
- VCC = 3.3 V
- Input timing voltage : Determined with V_{IL} = 0.6 V, V_{IH} = 2.7 V
- Output timing voltage: Determined with V_{OL} = 1.65 V, V_{OH} = 1.65 V

Figure 22.14 Timing Diagram (2)

Memory Expansion Mode and Microprocessor Mode

(For setting with no wait)

VCC = 3.3 V



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

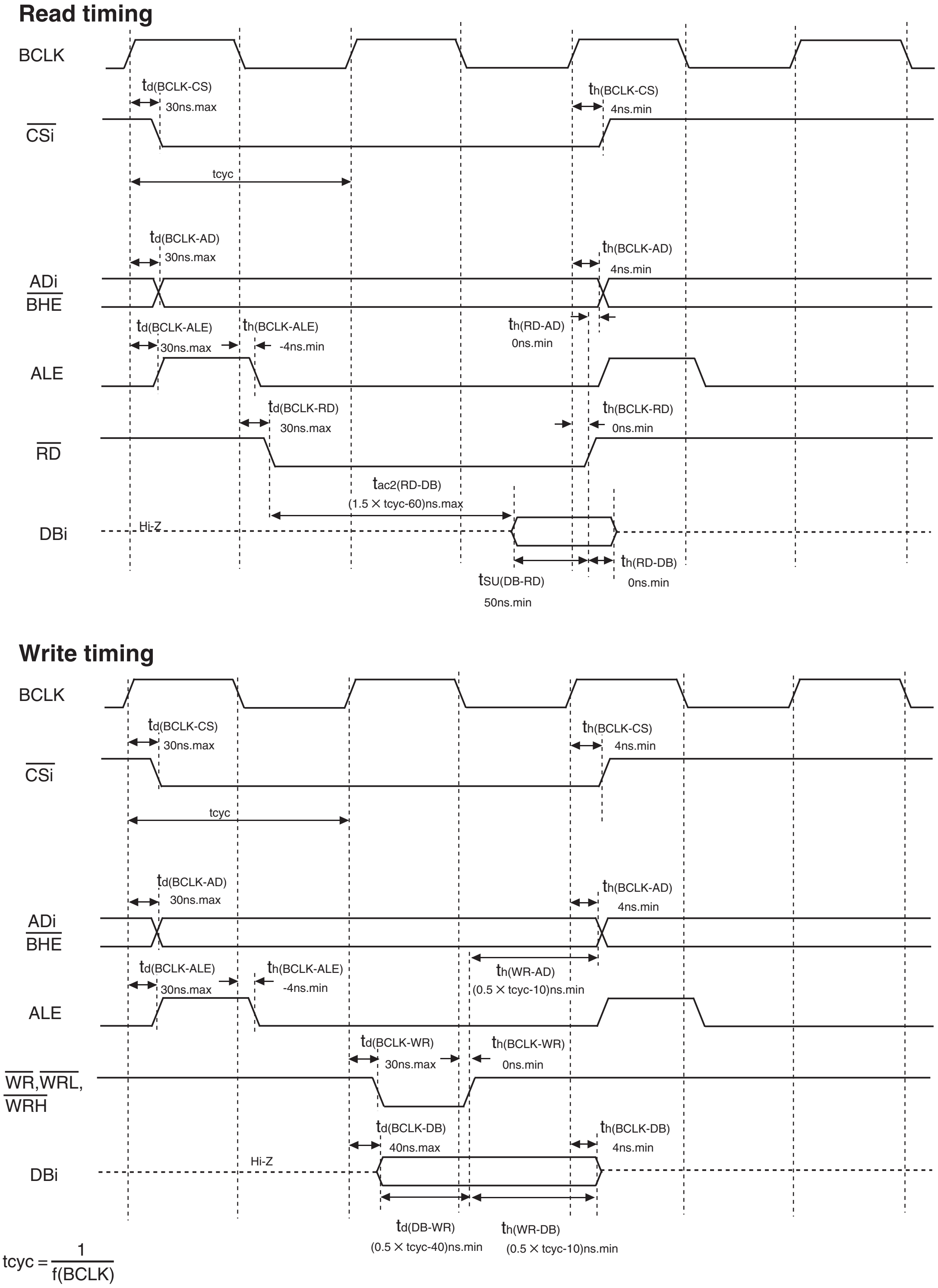
- VCC = 3.3 V
- Input timing voltage : V_{IL} = 0.6 V, V_{IH} = 2.7 V
- Output timing voltage : V_{OL} = 1.65 V, V_{OH} = 1.65 V

Figure 22.15 Timing Diagram (3)

Memory Expansion Mode and Microprocessor Mode

(For 1-wait setting and external area access)

VCC = 3.3 V



$$t_{cyc} = \frac{1}{f(BCLK)}$$

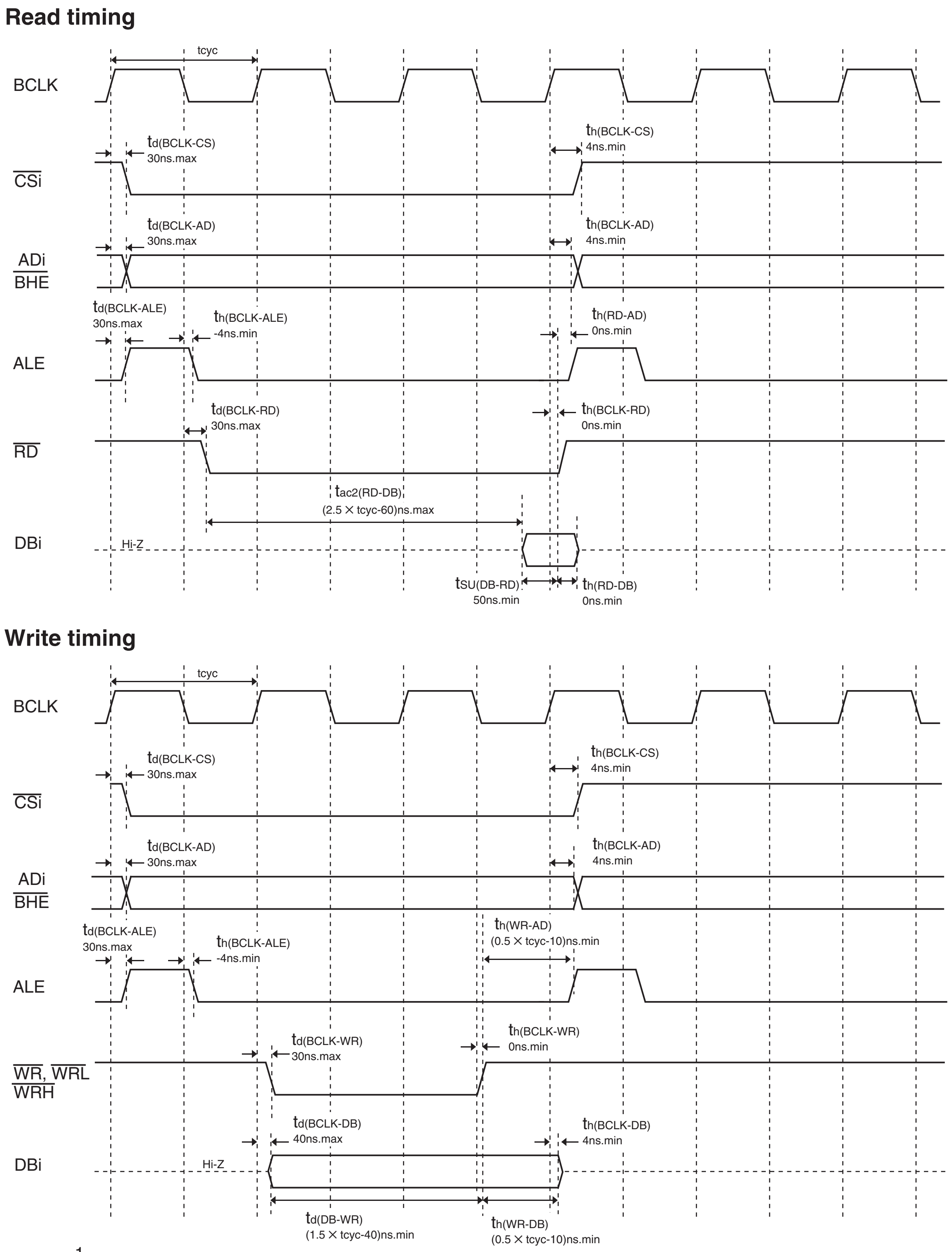
Measuring conditions :
- VCC = 3.3 V
- Input timing voltage : V_{IL} = 0.6 V, V_{IH} = 2.7 V
- Output timing voltage : V_{OL} = 1.65 V, V_{OH} = 1.65 V

Figure 22.16 Timing Diagram (4)

Memory Expansion Mode and Microprocessor Mode

(For 2-wait setting and external area access)

VCC = 3.3 V



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

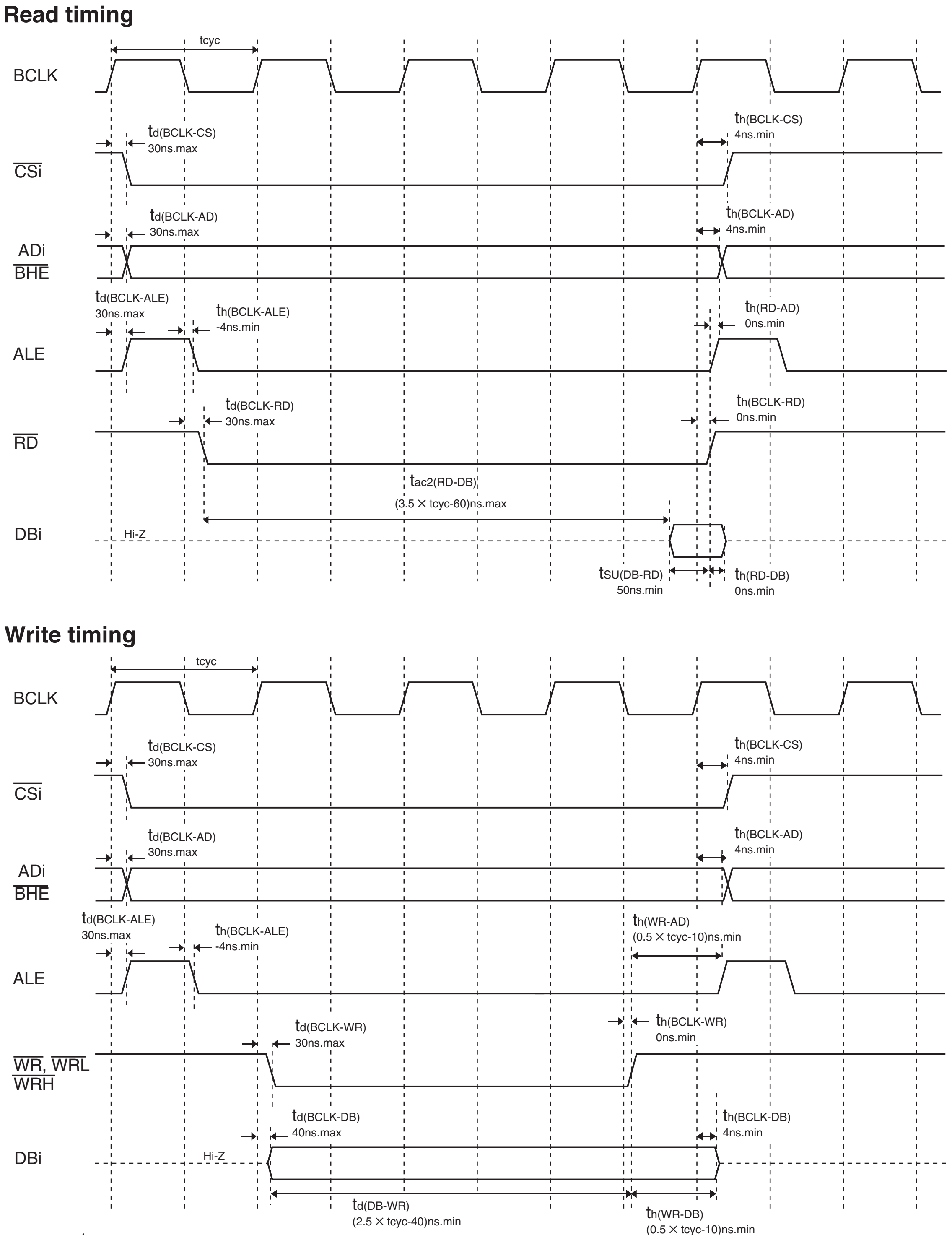
- VCC = 3.3 V
- Input timing voltage : V_{IL} = 0.6 V, V_{IH} = 2.7 V
- Output timing voltage : V_{OL} = 1.65 V, V_{OH} = 1.65 V

Figure 22.17 Timing Diagram (5)

Memory Expansion Mode and Microprocessor Mode

(For 3-wait setting and external area access)

VCC = 3.3 V



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

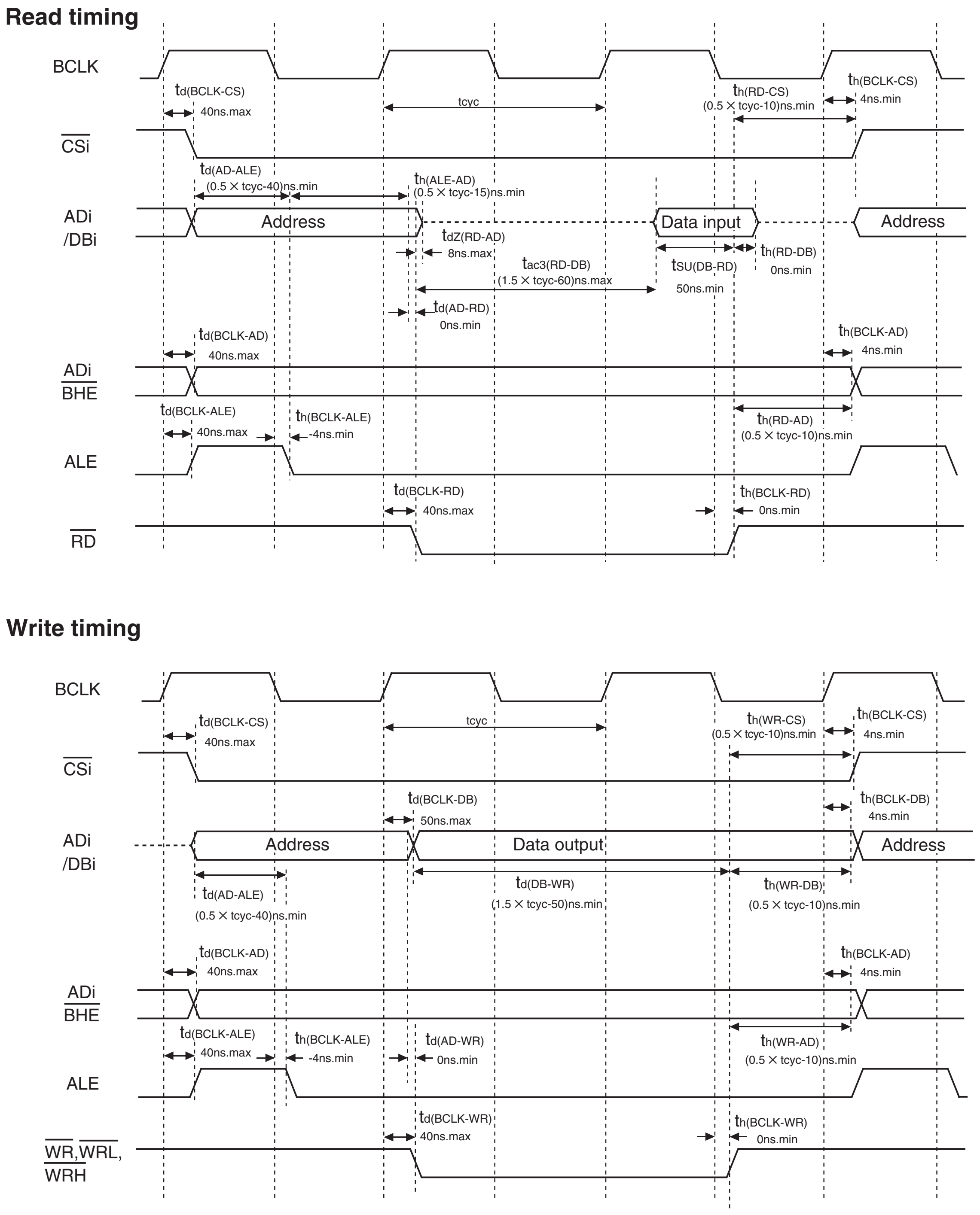
- VCC = 3.3 V
- Input timing voltage : V_{IL} = 0.6 V, V_{IH} = 2.7 V
- Output timing voltage : V_{OL} = 1.65 V, V_{OH} = 1.65 V

Figure 22.18 Timing Diagram (6)

Memory Expansion Mode and Microprocessor Mode

VCC = 3.3 V

(For 2-wait setting, external area access and multiplexed bus selection)



$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :

- VCC = 3.3 V
- Input timing voltage : V_{IL} = 0.6 V, V_{IH} = 2.7 V
- Output timing voltage : V_{OL} = 1.65 V, V_{OH} = 1.65 V

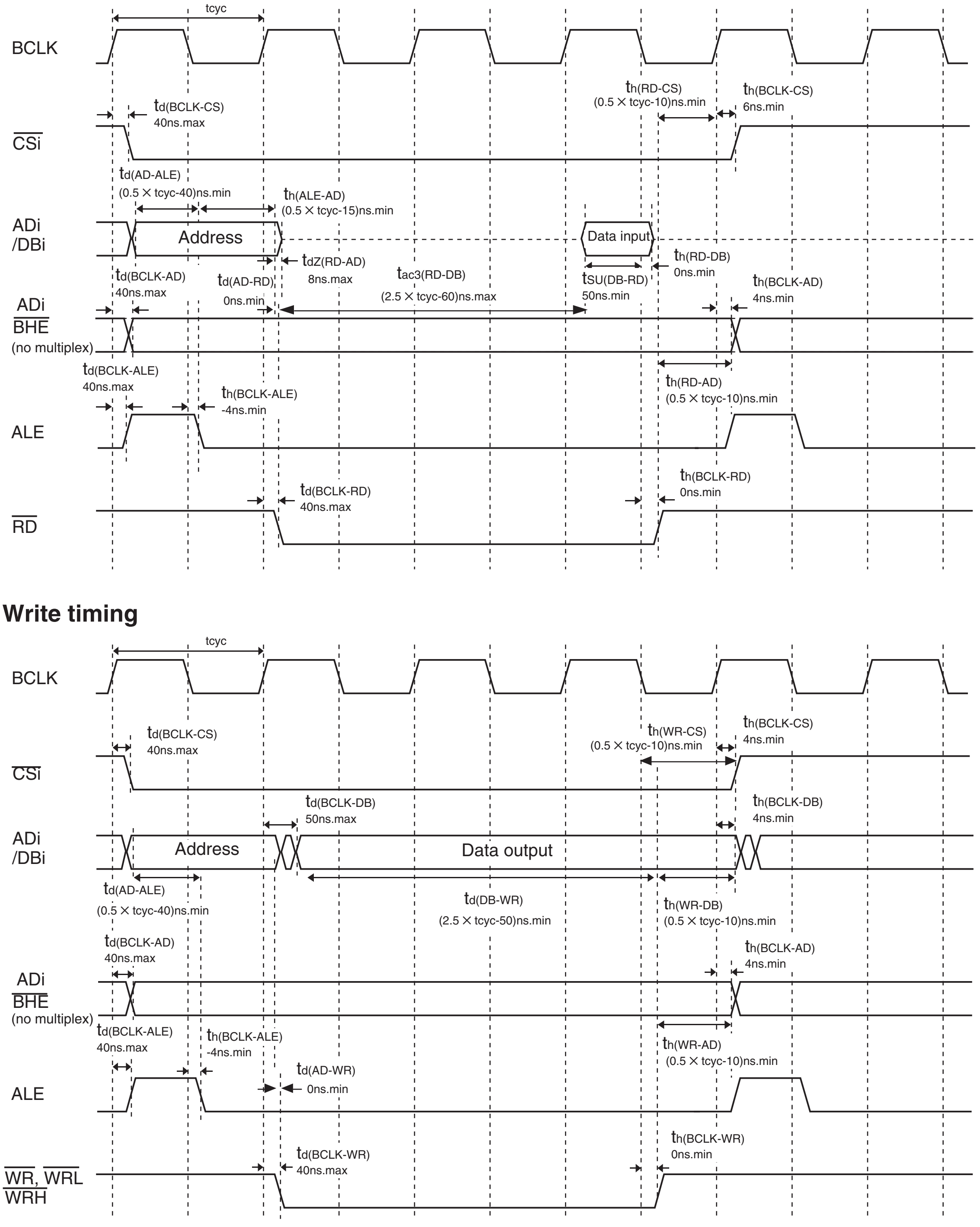
Figure 22.19 Timing Diagram (7)

Memory Expansion Mode and Microprocessor Mode

VCC = 3.3 V

(For 3-wait setting, external area access and multiplexed bus selection)

Read timing



Write timing

$$tcyc = \frac{1}{f(BCLK)}$$

Measuring conditions :
- VCC = 3.3 V
- Input timing voltage : V_{IL} = 0.6 V, V_{IH} = 2.7 V
- Output timing voltage : V_{OL} = 1.65 V, V_{OH} = 1.65 V

Figure 22.20 Timing Diagram (8)

23. Usage Notes

23.1 SFRs

There are the SFRs with write-only bits which can only be written to. Set these registers with undefined values. When establishing the next value by altering the present value, write the present value to the RAM as well as to the register. Transfer the next value to the register after making changes in the RAM.
Table 23.1 lists Registers with Write-only Bits.

Table 23.1 Registers with Write-only Bits

Register Name	Symbol	Address
Watchdog Timer Start Register	WDTS	000Eh
Timer A1-1 Register	TA11	01C3h, 01C2h
Timer A2-1 Register	TA21	01C5h, 01C4h
Timer A4-1 Register	TA41	01C7h, 01C6h
Dead Time Timer	DTT	01CCh
Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	01CDh
SI/O6 Bit Rate Register (1)	S6BRG	01D9h
SI/O3 Bit Rate Register	S3BRG	01E3h
SI/O4 Bit Rate Register	S4BRG	01E7h
SI/O5 Bit Rate Register (1)	S5BRG	01EBh
UART2 Bit Rate Register	U2BRG	01F9h
UART2 Transmit Buffer Register	U2TB	01FBh, 01FAh
Up-Down Flag	UDF	0384h
Timer A0 Register	TA0	0387h, 0386h
Timer A1 Register	TA1	0389h, 0388h
Timer A2 Register	TA2	038Bh, 038Ah
Timer A3 Register	TA3	038Dh, 038Ch
Timer A4 Register	TA4	038Fh, 038Eh
UART0 Bit Rate Register	U0BRG	03A1h
UART0 Transmit Buffer Register	U0TB	03A3h, 03A2h
UART1 Bit Rate Register	U1BRG	03A9h
UART1 Transmit Buffer Register	U1TB	03ABh, 03AAh

NOTE:

1. These registers are only in the 128-pin version.

23.2 External Bus

When resetting CNVSS pin with "H" input, contents of internal ROM cannot be read out.

23.3 External Clock

Do not stop the external clock when it is connected to the XIN pin and the main clock is selected as the CPU clock.

23.4 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met. (Refer to **22. Electrical characteristics**.)

23.5 Power Control

- When exiting stop mode by hardware reset, set $\overline{\text{RESET}}$ pin to “L” until a main clock oscillation is stabilized.

- Set the MR0 bit in the TAiMR register (i = 0 to 4) to 0 (pulse is not output) to use the timer A to exit stop mode.

- In the main clock oscillation or low power dissipation mode, set the CM02 bit in the CM0 register to 0 (do not stop peripheral function clock in wait mode) before shifting to stop mode.

- When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not execute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue roadstead the instructions following WAIT, and depending on timing, some of these may execute before the microcomputer enters wait mode.

  Program example when entering wait mode

```
Program Example:        JMP.B    L1       ; Insert JMP.B instruction before WAIT instruction
                   L1:
                        FSET     I        ;
                        WAIT              ; Enter wait mode
                        NOP               ; More than 4 NOP instructions
                        NOP
                        NOP
                        NOP
```

- When entering stop mode, describe as follows.
  (1) To use the BSET instruction for entering stop mode:
  Write the BSET instruction (BSET bit, base:16) as described below.
  When entering stop mode, DMA transfer must be disabled.

```
                        BSET     0,CM1      ; Stop mode setting [bit, base:16]
                        JMP.B    L1         ;
                   L1:
                        NOP                 ; Countermeasure to avoid the program from
                        NOP                 ; stopping by reading instruction ahead
                        NOP                 ; (insert 4 or more NOPs)
                        NOP                 ;
```

  (2) To use the MOV instruction for entering stop mode:
  Write the MOV instruction (MOV.B #IMM8, abs16) as described below.
  When entering stop mode, DMA transfer must be disabled.
  Change the ***src*** value (marked as “#21”), depending on your usage condition.

```
                        MOV.B    #21H,CM1   ; Stop mode setting [#IMM8, abs16]
                        JMP.B    L1         ;
                   L1:
                        NOP                 ; Countermeasure to avoid the program from
                        NOP                 ; stopping by reading instruction ahead
                        NOP                 ; (insert 4 or more NOPs)
                        NOP                 ;
```

- When entering medium-speed mode after transferring to stop mode from low-speed mode and low power dissipation mode, write the MOV instruction (MOV.W #IMM16, abs16) as described below.
  When entering stop mode and exiting from stop mode, DMA transfer must be disabled.
  Change the ***src*** value (marked as “#2118”) depending on your usage condition.

```
          MOV.W   #2118H,CM0    ; Stop mode setting [#IMM16, abs16]
          JMP.S   L1            ;
L1:
          NOP                   ; Countermeasure to avoid the program from
          NOP                   ; stopping by reading instruction ahead
          NOP                   ; (insert 4 or more NOPs)
          NOP                   ;
```

- Wait until the main clock oscillation stabilizes, before switching the clock source for CPU clock to the main clock.
  Similarly, wait until the sub clock oscillation stabilizes, before switching the clock source for CPU clock to the sub clock.

- Suggestions to reduce power consumption.

Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that high-impedance state. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

A/D converter

When A/D conversion is not performed, set the VCUT bit in the ADCON1 register to 0 (VREF not connection). When A/D conversion is performed, start the A/D conversion at least 1 µs or longer after setting the VCUT bit to 1 (VREF connection).

D/A converter

When not performing D/A conversion, set the DAiE bit (i = 0, 1) in the DACON register to 0 (input disabled) and DAi register to 00h.

Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

23.6 Oscillation Stop, Re-oscillation Detection Function

If the following conditions are all met, the following restriction occur in operation of oscillation stop, re-oscillation stop detection interrupt.

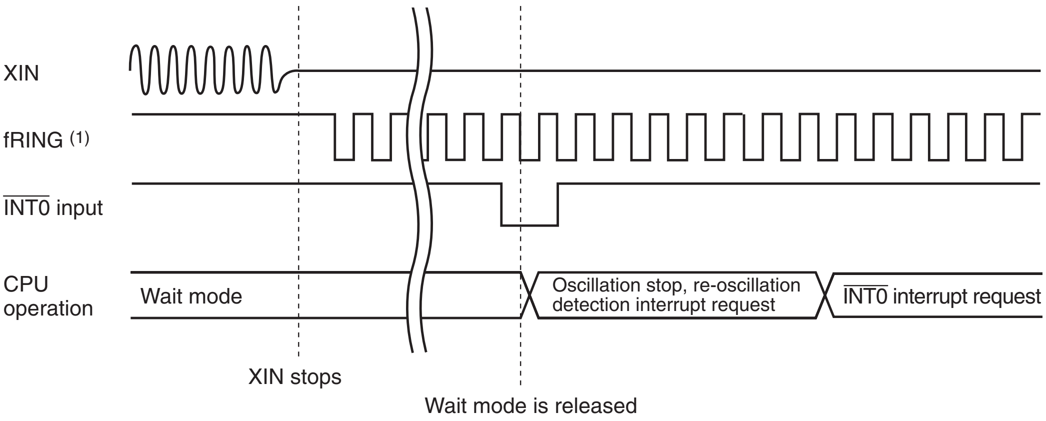
Conditions

- CM20 bit in CM2 register =1 (oscillation stop, re-oscillation stop detection function enabled)
- CM27 bit in CM2 register =1 (oscillation stop, re-oscillation stop detection interrupt)
- CM02 bit in CM0 register =0 (do not stop peripheral function clock in wait mode)
- Enter wait mode from high-speed or middle-speed mode

Restriction

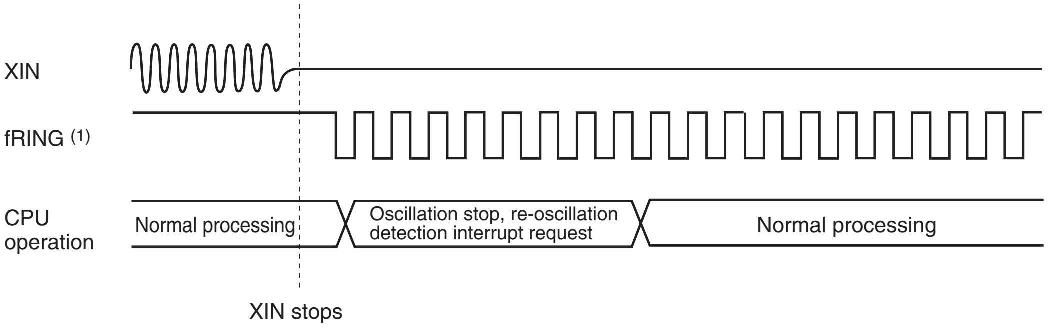
If the oscillation of XIN stops during wait mode, the oscillation stop, re-oscillation stop detection interrupt request is generated after the MCU is exits wait mode, without starting immediately.

Figures 23.1 and 23.2 show the Oscillation Stop, Re-oscillation Stop Detection Operation Timing.



NOTE:
1. This clock is generated by the on-chip oscillator. It is not supplies after reset.
The operating clock can changes from on-chip oscillator clock (on-chip oscillation oscillating) to BCLK by using oscillation stop, re-oscillation detection function or setting the CM21 bit in the CM2 register.

Figure 23.1 Oscillation Stop, Re-oscillation Stop Detection Operation Timing at Wait Mode (when moving out of wait mode by using INT0 interrupt)



NOTE:
1. This clock is generated by the on-chip oscillator. It is not supplies after reset.
The operating clock can changes from on-chip oscillator clock (on-chip oscillation oscillating) to BCLK by using oscillation stop, re-oscillation detection function or setting the CM21 bit in the CM2 register.

Figure 23.2 Oscillation Stop, Re-oscillation Stop Detection Operation Timing at Normal Processing

23.7 Protection

Set the PRC2 bit in the PRCR register to 1 (write enabled) and then write to given address, and the PRC2 bit will be set to 0 (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to 1. Make sure no interrupts or no DMA transfers will occur between the instruction in which the PRC2 bit is set to 1 and the next instruction.

23.8 Interrupts

23.8.1 Reading Address 00000h

Do not read the address 00000h in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000h during the interrupt sequence. At this time, the IR bit for the accepted interrupt is set to 0.

If the address 00000h is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is set to 0. This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

23.8.2 Setting SP

Set any value in the SP (USP, ISP) before accepting an interrupt. The SP (USP, ISP) is set to 0000h after reset. Therefore, if an interrupt is accepted before setting any value in the SP (USP, ISP), the program may go out of control.

Especially when using $\overline{\text{NMI}}$ interrupt, set a value in the ISP at the beginning of the program. For the first and only the first instruction after reset, all interrupts including $\overline{\text{NMI}}$ interrupt are disabled.

23.8.3 $\overline{\text{NMI}}$ Interrupt

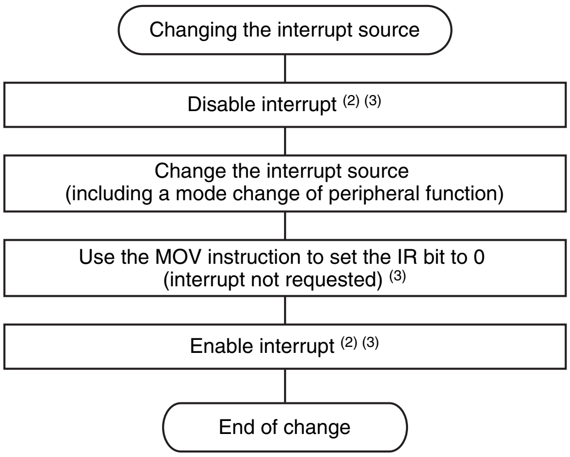
- The $\overline{\text{NMI}}$ interrupt cannot be disabled. If this interrupt is unused, connect the $\overline{\text{NMI}}$ pin to VCC via a resistor (pull-up).
- The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8_5 bit in the P8 register. Note that the P8_5 bit can only be read when determining the pin level in $\overline{\text{NMI}}$ interrupt routine.
- Stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM10 bit in the CM1 register is fixed to 0.
- Do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
- The low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.

23.8.4 Changing Interrupt Source

If the interrupt source is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). If you changed the interrupt source for an interrupt that needs to be used, be sure to set the IR bit for that interrupt to 0 (interrupt not requested).

Changing the interrupt source referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the source, polarity or timing of an interrupt, be sure to set the IR bit for that interrupt to 0 (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.

Figure 23.3 shows the Procedure for Changing Interrupt Source.


Changing the interrupt source
Disable interrupt (2) (3)
Change the interrupt source
(including a mode change of peripheral function)
Use the MOV instruction to set the IR bit to 0
(interrupt not requested) (3)
Enable interrupt (2) (3)
End of change

IR bit: A bit in the interrupt control register for the interrupt whose interrupt source is to be changed

NOTES:

1. The above settings must be executed individually. Do not execute two or more settings simultaneously (using one instruction).
2. Use the I flag for the $\overline{\mathrm{INTi}}$ interrupt (i = 0 to 8; 6 to 8 are only in the 128-pin version). For the interrupts from peripheral functions other than the $\overline{\mathrm{INTi}}$ interrupt, turn off the peripheral function that is the interrupt source in order not to generate an interrupt request before changing the interrupt source. In this case, if the maskable interrupts can all be disabled without causing a problem, use the I flag. Otherwise, if it is not possible to disable all maskable interrupts, use bits ILVL2 to ILVL0 of the interrupt whose source is changed.
3. Refer to **23.8.6 Rewrite Interrupt Control Register** for details about the instructions to use and the notes to be taken for instruction execution.

Figure 23.3 Procedure for Changing Interrupt Generate Factor

23.8.5 $\overline{\mathrm{INT}}$ Interrupt

- Either an “L” level of at least tW(INH) or an “H” level of at least tW(INL) width is necessary for the signal input to pins $\overline{\mathrm{INT0}}$ to $\overline{\mathrm{INT8}}$ [1] regardless of the CPU operation clock.
- If the POL bit in registers INT0IC to INT8IC [2], bits IFSR10 to IFSR15 in the IFSR1 register or bits IFSR23 to IFSR25 [3] in the IFSR2 register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to set the IR bit to 0 (interrupt not requested) after changing any of those register bits.

NOTES:

1. The pins $\overline{\mathrm{INT6}}$ to $\overline{\mathrm{INT8}}$ are only in the 128-pin version.
2. Registers INT6IC to INT8IC are only in the 128-pin version.
3. Bits IFSR23 to IFSR25 are effective only in the 128-pin version. In the 100-pin version, these bits are set to 0 (one edge).

23.8.6 Rewrite Interrupt Control Register

(a) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may be generated. Otherwise, disable the interrupt before rewriting the interrupt control register.

(b) To rewrite the interrupt control register for any interrupt after disabling that interrupt, care must be taken when selecting the instructions.

Changing any bit other than IR bit

If while executing an instruction, an interrupt request controlled by the register being modified is generated, the IR bit of the register may not be set to 1 (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

Changing IR bit

Depending on the instruction used, the IR bit may not always be set to 0 (interrupt not requested). Therefore, be sure to use the MOV instruction to set the IR bit to 0.

(c) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (b) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to 1 (interrupt enabled) before the interrupt control register is rewritten, owing to the effects of the internal bus and the instruction queue buffer.

Example 1: Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Set the TA0IC register to 00h.
    NOP                     ;
    NOP
    FSET    I               ; Enable interrupts.
```

The number of the NOP instruction is as follows.

- The PM20 bit in the PM2 register = 1 (1 wait) : 2
- The PM20 bit = 0 (2 waits) : 3
- When using HOLD function : 4

Example 2: Using the dummy read to the FSET instruction delay

```
INT_SWITCH2:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Set the TA0IC register to 00h.
    MOV.W   MEM, R0         ; Dummy read.
    FSET    I               ; Enable interrupts.
```

Example 3: Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
    PUSHC   FLG
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Set the TA0IC register to 00h.
    POPC    FLG             ; Enable interrupts.
```

23.8.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt request is generated.

23.9 DMAC

23.9.1 Write to DMAE Bit in DMiCON Register (i = 0, 1)

When both of the conditions below are met, follow the steps below.

Conditions

- The DMAE bit is set to 1 again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

Step 1: Write 1 to the DMAE bit and DMAS bit in the DMiCON register simultaneously [1].
Step 2: Make sure that the DMAi is in an initial state [2] in a program.
If the DMAi is not in an initial state, the above steps should be repeated.

NOTES:

1. The DMAS bit remains unchanged even if 1 is written. However, if 0 is written to this bit, it is set to 0 (DMA not requested). In order to prevent the DMAS bit from being modified to 0, 1 should be written to the DMAS bit when 1 is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
   Similarly, when writing to the DMAE bit with a read-modify-write instruction, 1 should be written to the DMAS bit in order to maintain a DMA request which is generated while the instruction is being executing.

2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is 1.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

23.10 Timers

23.10.1 Timer A

23.10.1.1 Timer A (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register and the TAi register before setting the TAiS bit in the TABSR register to 1 (count starts). Always make sure the TAiMR register is modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the counter is read at the same time it is reloaded, the value FFFFh is read. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to a high-impedance state.

23.10.1.2 Timer A (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH in the ONSF register, and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
Always make sure the TAiMR register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH, and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, FFFFh can be read in underflow, while reloading, and 0000h in overflow. When setting the TAi register to a value during a counter stop, the setting value can be read before a counter starts counting. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to a high-impedance state.

23.10.1.3 Timer A (One-shot Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, bits TA0TGL and TA0TGH in the ONSF register, and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
Always make sure the TAiMR register, bits TA0TGL and TA0TGH, and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

When setting the TAiS bit to 0 (count stops), the followings occur:

- A counter stops counting and a content of reload register is reloaded.
- TAiOUT pin outputs “L”.
- After one cycle of the CPU clock, the IR bit in the TAiIC register is set to 1 (interrupt request).

Output in one-shot timer mode synchronizes with a count source internally generated. When an external trigger has been selected, one-cycle delay of a count source as maximum occurs between a trigger input to TAiIN pin and output in one-shot timer mode.

The IR bit is set to 1 when timer operating mode is set with any of the following procedures:

- Select one-shot timer mode after reset.
- Change an operating mode from timer mode to one-shot timer mode.
- Change an operating mode from event counter mode to one-shot timer mode.

To use the timer Ai interrupt (the IR bit), set the IR bit to 0 after the changes listed above have been made.

When a trigger occurs, while counting, a counter reloads the reload register to continue counting after generating a re-trigger and counting down once. To generate a trigger while counting, generate a second trigger between occurring the previous trigger and operating longer than one cycle of a timer count source.

When the external trigger is selected as count start condition, do not input again the external trigger between 300 ns before the counter reaches 0000h.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to a high-impedance state.

23.10.1.4 Timer A (Pulse Width Modulation Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, bits TA0TGL and TA0TGH in the ONSF register, and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
Always make sure the TAiMR register, bits TA0TGL and TA0TGH, and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

The IR bit is set to 1 when setting a timer operating mode with any of the following procedures:

- Select pulse width modulation mode after reset.
- Change an operating mode from timer mode to pulse width modulation mode.
- Change an operating mode from event counter mode to pulse width modulation mode.

To use the timer Ai interrupt (the IR bit), set the IR bit to 0 by program after the above listed changes have been made.

When setting TAiS bit to 0 (count stops) during PWM pulse output, the following action occurs:

- Stop counting.
- When TAiOUT pin is output “H”, output level is set to “L” and the IR bit is set to 1.
- When TAiOUT pin is output “L”, both output level and the IR bit remain unchanged.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to a high-impedance state.

23.10.2 Timer B

23.10.2.1 Timer B (Timer Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit [1] in the TABSR or the TBSR register to 1 (count starts).
Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not.

NOTE:

1. Bits TB0S to TB2S are the bits 5 to 7 in the TABSR register, bits TB3S to TB5S are the bits 5 to 7 in the TBSR register.

A value of a counter, while counting, can be read in the TBi register at any time. FFFFh is read while reloading. Setting value is read between setting values in the TBi register at count stop and starting a counter.

23.10.2.2 Timer B (Event Counter Mode)

The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 5) register and TBi register before setting the TBiS bit in the TABSR or the TBSR register to 1 (count starts).
Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not.

The counter value can be read out on-the-fly at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always FFFFh. If the TBi register is read after setting a value in it while not counting but before the counter starts counting, the read value is the one that has been set in the register.

23.10.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 5) register before setting the TBiS bit in the TABSR or TBSR register to 1 (count starts).
Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not. To set the MR3 bit to 0 by writing to the TBiMR register while the TBiS bit = 1 (count starts), be sure to write the same value as previously written to bits TM0D0, TM0D1, MR0, MR1, TCK0, and TCK1 and, a 0 to the MR2 bit.

The IR bit in the TBiIC register goes to 1 (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The interrupt source can be determined by use of the MR3 bit in the TBiMR register within the interrupt routine.

If the interrupt source cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times timer B has overflowed.

To set the MR3 bit to 0 (no overflow), set the TBiMR register with setting the TBiS bit to 1 and counting the next count source after setting the MR3 bit to 1 (overflow).

Use the IR bit in the TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt source.

When a count is started and the first effective edge is input, an undefined value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

A value of the counter is undefined at the beginning of a count. The MR3 bit may be set to 1 and timer Bi interrupt request may be generated between a count start and an effective edge input.

For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an “H” level width or an “L” level width.

23.11 Thee-Phase Motor Control Timer Function

If there is a possibility that you may write data to TAi-1 register (i = 1, 2, 4) near Timer B2 overflow, read the value of TB2 register, verify that there is sufficient time until Timer B2 overflows, before doing an immediate write to TAi-1 register.

In order to shorten the period from reading TB2 register to writing data to TAi-1 register, ensure that no interrupt will be processed during this period.

If there is not enough time till Timer B2 overflows, only write to TAi-1 register after Timer B2 overflowed.

23.12 Serial Interface

23.12.1 Clock Synchronous Serial I/O Mode

23.12.1.1 Transmission/reception

With an external clock selected, and choosing the $\overline{\text{RTS}}$ function, the output level of the $\overline{\text{RTSi}}$ pin goes to "L" when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{\text{RTSi}}$ pin goes to "H" when reception starts. So if the $\overline{\text{RTSi}}$ pin is connected to the $\overline{\text{CTSi}}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{\text{RTS}}$ function has no effect.

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins $\overline{\text{RTS2}}$ and CLK2 go to a high-impedance state.

23.12.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register = 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit = 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in UiTB register)
- If $\overline{\text{CTS}}$ function is selected, input on the $\overline{\text{CTSi}}$ pin = L

23.12.1.3 Reception

In operating the clock synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TXDi (i = 0 to 2) pin when receiving data.

When an internal clock is selected, set the TE bit in the UiC1 register (i = 0 to 2) to 1 (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated. When an external clock is selected, set the TE bit to 1 and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.

When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RI bit in the UiC1 register = 1 (data present in the UiRB register), an overrun error occurs and the OER bit in the UiRB register is set to 1 (overrun error occurred). In this case, because the content of the UiRB register is undefined, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the IR bit in the SiRIC register does not change state.

To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

When an external clock is selected, the conditions must be met while if the CKPOL bit = 0, the external clock is in the high state; if the CKPOL bit = 1, the external clock is in the low state.

- The RE bit in the UiC1 register = 1 (reception enabled)
- The TE bit in the UiC1 register = 1 (transmission enabled)
- The TI bit in the UiC1 register = 0 (data present in the UiTB register)

23.12.2 Special Modes

23.12.2.1 Special Mode 1 (I^{2}C Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the UiSMR4 register to 0 (start and stop conditions not output) and wait for more than half cycle of the transfer clock before setting each condition generate bit (bits STAREQ, RSTAREQ, and STPREQ) from 0 (clear) to 1 (start).

23.12.2.2 Special Mode 2

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins $\overline{\text{RTS2}}$ and CLK2 go to a high-impedance state.

23.12.2.3 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to 1 (transmission completed) and U2ERE bit in the U2C1 register to 1 (error signal output) after reset. Therefore, when using SIM mode, be sure to set the IR bit to 0 (no interrupt request) after setting these bits.

23.12.3 SI/Oi (i = 3 to 6) [1]

The SOUTi default value which is set to the SOUTi pin by the SMi7 in the SiC register bit approximately 10 ns may be output when changing the SMi3 bit in the SiC register from 0 (I/O port) to 1 (SOUTi output and CLKi function) while the SMi2 bit in the SiC register to 0 (SOUTi output) and the SMi6 bit is set to 1 (internal clock). And then the SOUTi pin is held high-impedance.
If the level which is output from the SOUTi pin is a problem when changing the SMi3 bit from 0 to 1, set the default value of the SOUTi pin by the SMi7 bit.

NOTE:

1. SI/O5 and SI/O6 are only in the 128-pin version.

23.13 A/D Converter

Set the ADCON0 (except bit 6), registers ADCON1 and ADCON2 when A/D conversion is stopped (before a trigger occurs). After stopping A/D conversion, the VCUT bit in the ADCON1 register is changed from 1 (VREF connected) to 0 (VREF not connected),

When the VCUT bit is changed from 0 to 1, start A/D conversion after passing 1 µs or longer.

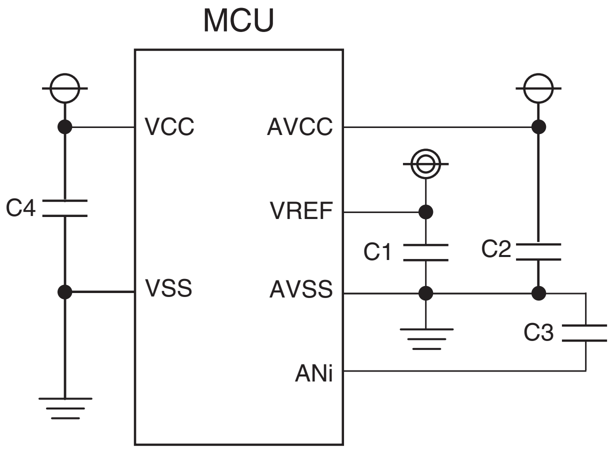
To prevent noise-induced device malfunction or latch-up, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (ANi (i = 0 to 7), AN0_i, and AN2_i) each and the AVSS pin. Similarly, insert a capacitor between the VCC pin and the VSS pin.
Figure 23.4 shows the Use of Capacitors to Reduce Noise.

Make sure the port direction bits for those pins that are used as analog inputs are set to 0 (input mode). Also, if the TGR bit in the ADCON0 register = 1 (external trigger), make sure the port direction bit for the $\overline{\text{ADTRG}}$ pin is set to 0 (input mode).

When using key input interrupt, do not use any of four pins AN4 to AN7 as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)

The φAD frequency must be 10 MHz or less. Without sample and hold, limit the φAD frequency to 250 kHz or more. With the sample and hold, limit the φAD frequency to 1 MHz or more.

When changing an A/D operating mode, select analog input pin again in bits CH2 to CH0 in the ADCON0 register and bits SCAN1 to SCAN0 in the ADCON1 register.



ANi: ANi, AN0_i, and AN2_i (i =0 to 7)

NOTES:
1. C1 ≥ 0.47 µF, C2 ≥ 0.47 µF, C3 ≥ 100 pF, C4 ≥ 0.1 µF (reference).
2. Use thick and shortest possible wiring to connect capacitors.

Figure 23.4 Use of Capacitors to Reduce Noise

If the CPU reads the ADi register (i = 0, 1) at the same time the conversion result is stored in the ADi register after completion of A/D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a sub clock is selected for CPU clock.

- When operating in one-shot or single-sweep mode
  Check to see that A/D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register to see if A/D conversion is completed.)
- When operating in repeat mode or repeat sweep mode 0 or 1
  Use the main clock for CPU clock directly without dividing it.

If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to 0 (A/D conversion halted), the conversion result of the A/D converter is undefined. The contents of ADi register irrelevant to A/D conversion may also become undefined. If while A/D conversion is underway the ADST bit is set to 0 in a program, ignore the values of all ADi registers.

When setting the ADST bit to 0 in single sweep mode during A/D conversion and A/D conversion is aborted, disable the interrupt before setting the ADST bit to 0.

The applied intermediate potential may cause more increase in power consumption than other analog input pins (AN0 to AN3, AN0_0 to AN0_7, and AN2_0 to AN2_7), since the AN4 to AN7 are used with the $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$.

23.14 CAN Module

23.14.1 Reading C0STR Register

The CAN module on the M16C/6N Group (M16C/6NL, M16C/6NN) updates the status of the C0STR register in a certain period. When the CPU and the CAN module access to the C0STR register at the same time, the CPU has the access priority; the access from the CAN module is disabled. Consequently, when the updating period of the CAN module matches the access period from the CPU, the status of the CAN module cannot be updated. (See **Figure 23.5 When Updating Period of CAN Module Matches Access Period from CPU**.)

Accordingly, be careful about the following points so that the access period from the CPU should not match the updating period of the CAN module:

(a) There should be a wait time of 3fCAN or longer (see **Table 23.2 CAN Module Status Updating Period**) before the CPU reads the C0STR register. (See **Figure 23.6 With a Wait Time of 3 fCAN Before CPU Read**.)

(b) When the CPU polls the C0STR register, the polling period must be 3 fCAN or longer. (See **Figure 23.7 When Polling Period of CPU is 3 fCAN or Longer**.)

Table 23.2 CAN Module Status Updating Period

3fCAN Period = 3 × XIN (Original Oscillation Period) × Division Value of CAN Clock (CCLK)	
(Example 1) Condition XIN 16 MHz CCLK: Divide-by-1	3 fCAN period = 3 × 62.5 ns × 1 = 187.5 ns
(Example 2) Condition XIN 16 MHz CCLK: Divide-by-2	3 fCAN period = 3 × 62.5 ns × 2 = 375 ns
(Example 3) Condition XIN 16 MHz CCLK: Divide-by-4	3 fCAN period = 3 × 62.5 ns × 4 = 750 ns
(Example 4) Condition XIN 16 MHz CCLK: Divide-by-8	3 fCAN period = 3 × 62.5 ns × 8 = 1.5 µs
(Example 5) Condition XIN 16 MHz CCLK: Divide-by-16	3 fCAN period = 3 × 62.5 ns × 16 = 3 µs

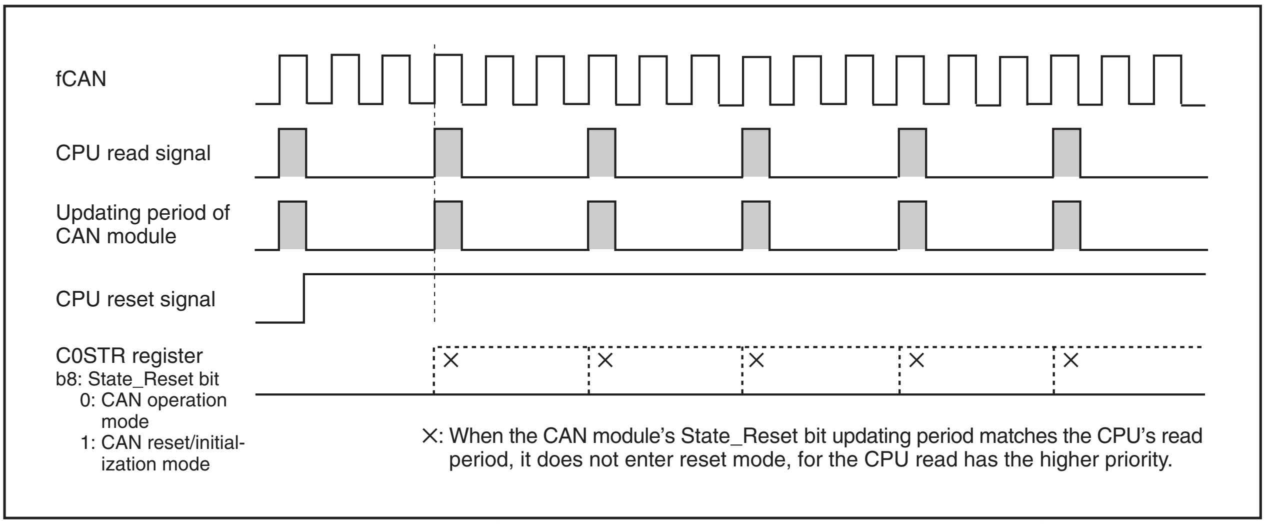

fCAN
CPU read signal
Updating period of CAN module
CPU reset signal
C0STR register
b8: State_Reset bit
0: CAN operation mode
1: CAN reset/initial-ization mode
×: When the CAN module's State_Reset bit updating period matches the CPU's read period, it does not enter reset mode, for the CPU read has the higher priority.

Figure 23.5 When Updating Period of CAN Module Matches Access Period from CPU

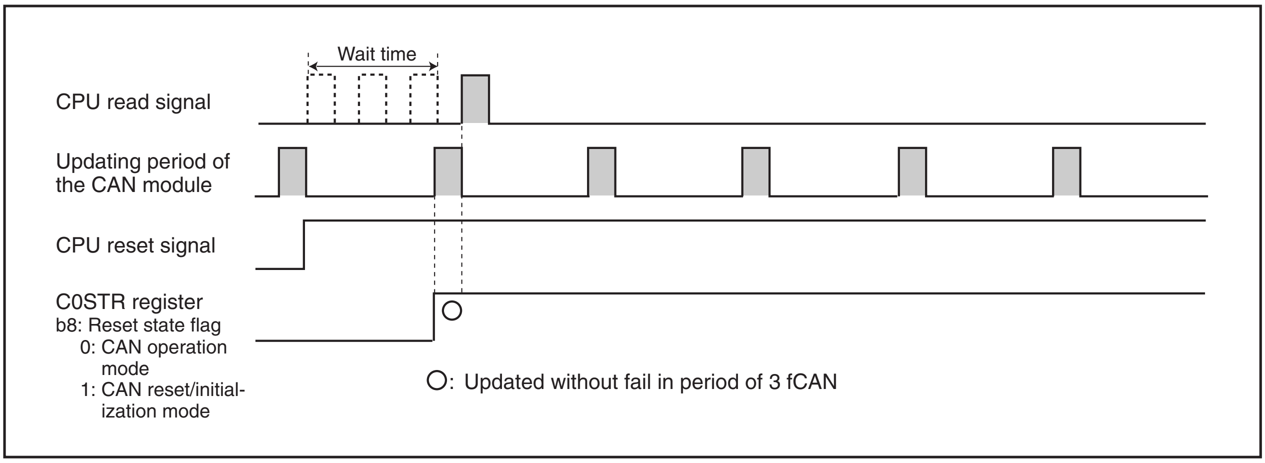

Wait time
CPU read signal
Updating period of the CAN module
CPU reset signal
C0STR register
b8: Reset state flag
0: CAN operation mode
1: CAN reset/initial-ization mode
○: Updated without fail in period of 3 fCAN

Figure 23.6 With Wait Time of 3 fCAN Before CPU Read

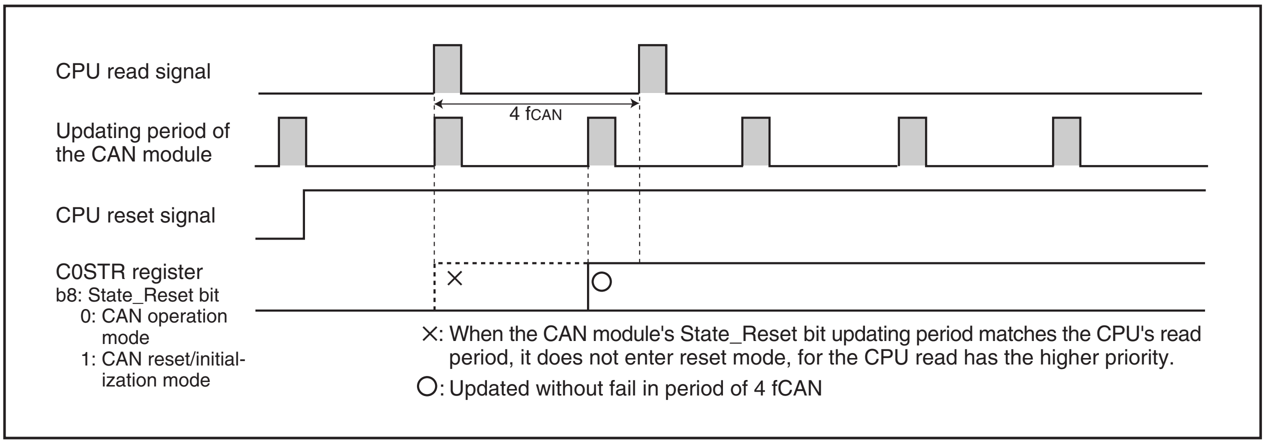

CPU read signal
4 fCAN
Updating period of the CAN module
CPU reset signal
C0STR register
b8: State_Reset bit
0: CAN operation mode
1: CAN reset/initial-ization mode
×: When the CAN module's State_Reset bit updating period matches the CPU's read period, it does not enter reset mode, for the CPU read has the higher priority.
○: Updated without fail in period of 4 fCAN

Figure 23.7 When Polling Period of CPU is 3 fCAN or Longer

23.14.2 Performing CAN Configuration

If the Reset bit in the C0CTLR registe is changed from 0 (operation mode) to 1 (reset/initialization mode) in order to place the CAN module from CAN operation mode into CAN reset/initialization mode, always be sure to check that the State_Reset bit in the C0STR register is set to 1 (reset mode).

Similarly, if the Reset bit is changed from 1 to 0 in order to place the CAN module from CAN reset/ initialization mode into CAN operation mode, always be sure to check that the State_Reset bit is set to 0 (operation mode).

The procedure is described below.

To Place CAN Module from CAN Operation Mode into CAN Reset/Initialization Mode

- Change the Reset bit from 0 to 1
- Check that the State_Reset bit is set to 1

To Place CAN Module from CAN Reset/Initialization Mode into CAN Operation Mode

- Change the Reset bit from 1 to 0
- Check that the State_Reset bit is set to 0

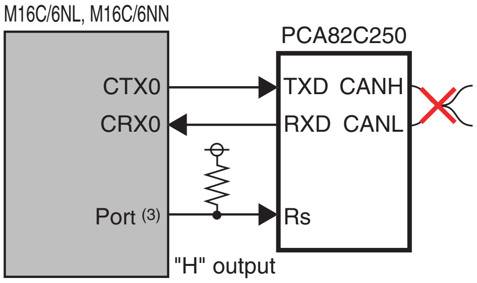
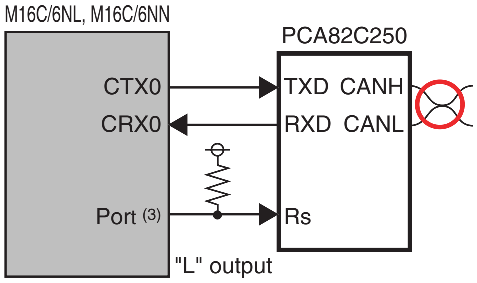
23.14.3 Suggestions to Reduce Power Consumption

When not performing CAN communication, the operation mode of CAN transceiver should be set to “standby mode” or “sleep mode”.

When performing CAN communication, the power consumption in CAN transceiver in not performing CAN communication can be substantially reduced by controlling the operation mode pins of CAN transceiver.

Tables 23.3 and 23.4 show the Recommended Pin Connections.

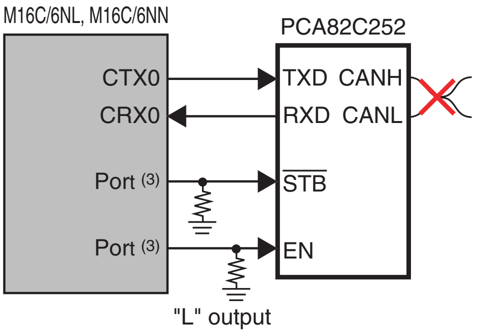
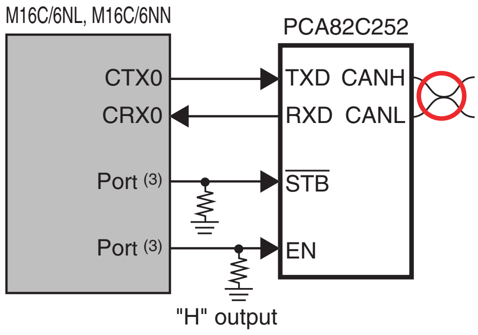
Table 23.3 Recommended Pin Connections (In case of PCA82C250: Philips product)

	Standby Mode	High-speed Mode
Rs pin [(1)]	“H”	“L”
Power consumption in CAN transceiver [(2)]	less than 170 µA	less than 70 mA
CAN communication	impossible	possible
Connection	 M16C/6NL, M16C/6NN PCA82C250 CTX0 CRX0 Port (3) TXD CANH RXD CANL Rs "H" output	 M16C/6NL, M16C/6NN PCA82C250 CTX0 CRX0 Port (3) TXD CANH RXD CANL Rs "L" output

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. In case of Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

Table 23.4 Recommended Pin Connections (In case of PCA82C252: Philips product)

	Sleep Mode	Normal Operation Mode
$\overline{\text{STB}}$ pin [(1)]	“L”	“H”
EN pin [(1)]	“L”	“H”
Power consumption in CAN transceiver [(2)]	less than 50 µA	less than 35 mA
CAN communication	impossible	possible
Connection	 M16C/6NL, M16C/6NN PCA82C252 CTX0 CRX0 Port (3) Port (3) TXD CANH RXD CANL STB EN "L" output	 M16C/6NL, M16C/6NN PCA82C252 CTX0 CRX0 Port (3) Port (3) TXD CANH RXD CANL STB EN "H" output

NOTES:

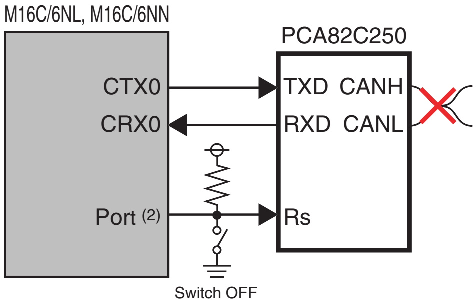
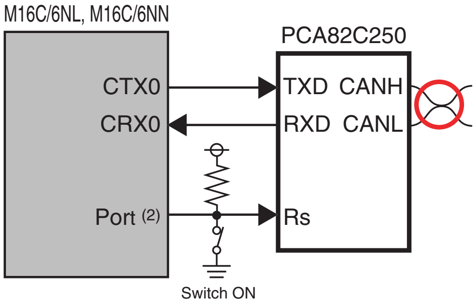
1. The pin which controls the operation mode of CAN transceiver.
2. Ta = 25 °C
3. Connect to enabled port to control CAN transceiver.

23.14.4 CAN Transceiver in Boot Mode

When programming the flash memory in boot mode via CAN bus, the operation mode of CAN transceiver should be set to “high-speed mode” or “normal operation mode”. If the operation mode is controlled by the microcomputer, CAN transceiver must be set the operation mode to “high-speed mode” or “normal operation mode” before programming the flash memory by changing the switch etc.

Tables 23.5 and 23.6 show the Pin Connections of CAN Transceiver.

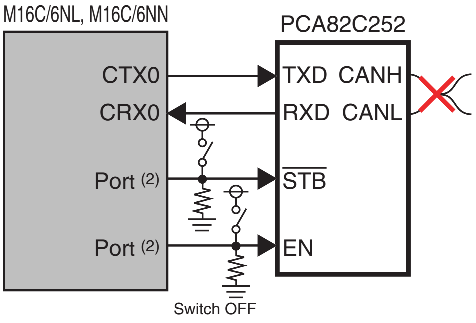
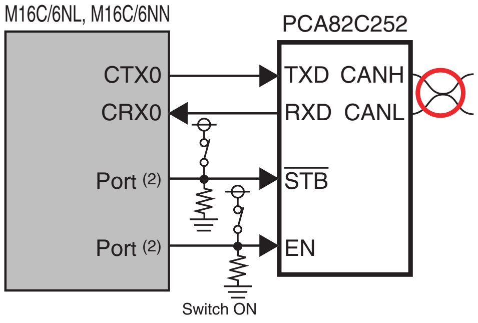
Table 23.5 Pin Connections of CAN Transceiver (In case of PCA82C250: Philips product)

	Standby Mode	High-speed Mode
Rs pin [1]	“H”	“L”
CAN communication	impossible	possible
Connection	 M16C/6NL, M16C/6NN PCA82C250 CTX0 CRX0 Port (2) TXD CANH RXD CANL Rs Switch OFF	 M16C/6NL, M16C/6NN PCA82C250 CTX0 CRX0 Port (2) TXD CANH RXD CANL Rs Switch ON

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

Table 23.6 Pin Connections of CAN Transceiver (In case of PCA82C252: Philips product)

	Sleep Mode	Normal Operation Mode
$\overline{\text{STB}}$ pin [1]	“L”	“H”
EN pin [1]	“L”	“H”
CAN communication	impossible	possible
Connection	 M16C/6NL, M16C/6NN PCA82C252 CTX0 CRX0 Port (2) Port (2) TXD CANH RXD CANL STB EN Switch OFF	 M16C/6NL, M16C/6NN PCA82C252 CTX0 CRX0 Port (2) Port (2) TXD CANH RXD CANL STB EN Switch ON

NOTES:

1. The pin which controls the operation mode of CAN transceiver.
2. Connect to enabled port to control CAN transceiver.

23.15 Programmable I/O Ports

If a low-level signal is applied to the $\overline{\text{NMI}}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on $\overline{\text{NMI}}$ pin enabled), pins P7_2 to P7_5, P8_0 and P8_1 go to a high-impedance state.

Setting the SM32 bit in the S3C register to 1 causes the P9_2 pin to go to a high-impedance state.
Setting the SM42 bit in the S4C register to 1 causes the P9_6 pin to go to a high-impedance state (1).
Setting the SM52 bit in the S5C register to 1 causes the P11_2 pin to go to a high-impedance state (2).
Setting the SM62 bit in the S6C register to 1 causes the P11_6 pin to go to a high-impedance state (2).

NOTES:

1. When using SI/O4, set the SM43 bit in the S4C register to 1 (SOUT4 output, CLK4 function) and the port direction bit corresponding for SOUT4 pin to 0 (input mode).
2. The S5C and S6C registers are only in the 128-pin version. When using these registers, set these registers after setting the PU37 bit in the PUR3 register to 1 (Pins P11 to P14 are usable).

The input threshold voltage of pins differs between programmable I/O ports and peripheral functions. Therefore, if any pin is shared by a programmable I/O port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions VIH and VIL (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable I/O port or the peripheral function—is currently selected.

When changing the PD14_i bit (i = 0, 1) in the PC14 register from 0 (input port) to 1 (output port), follow the procedures below (128-pin version only).

	Setting Procedure	
(1) Set P14_i bit	:MOV.B #00000001b, PC14	; P14_i bit setting
(2) Change PD14_i bit to 1 by MOV instruction	:MOV.B #00110001b, PC14	; Change to output port

Undefined values are read from bits P3_7 to P3_4, PD3_7 to PD3_4 by reading registers P3 and PD3 when bits PM01 to PM00 in the PM0 register are set to 01b (memory expansion mode) or 11b (microprocessor mode) and setting the PM11 bit to 1.
Use the MOV instruction when rewriting registers P3 and PD3 (including the case that the size specifier is “.W” and registers P2 and PD2 are rewritten).

When bits PM01 to PM00 are rewritten, “L” is output from pins P3_7 to P3_4 during 0.5 cycles of the BCLK by setting bits PM01 to PM00 in the PM0 register to 01b (memory expansion mode) or 11b (microprocessor mode) from 00b (single-chip mode) after setting the PM11 bit to 1.

23.16 Dedicated Input Pin

When dedicated input pin voltage is larger than VCC pin voltage, latch up occurs.

When different power supplied to the system, and input voltage of unused dedicated input pin is larger than voltage of VCC pin, connect dedicated input pin to VCC via resistor (approximately 1 kΩ).

Figure 23.8 shows the Circuit Connection.

This note is also applicable when VINPUT exceeds VCC during power-up.

The resistor is not necessary when VCC pin voltage is same or larger than dedicated input pin voltage.

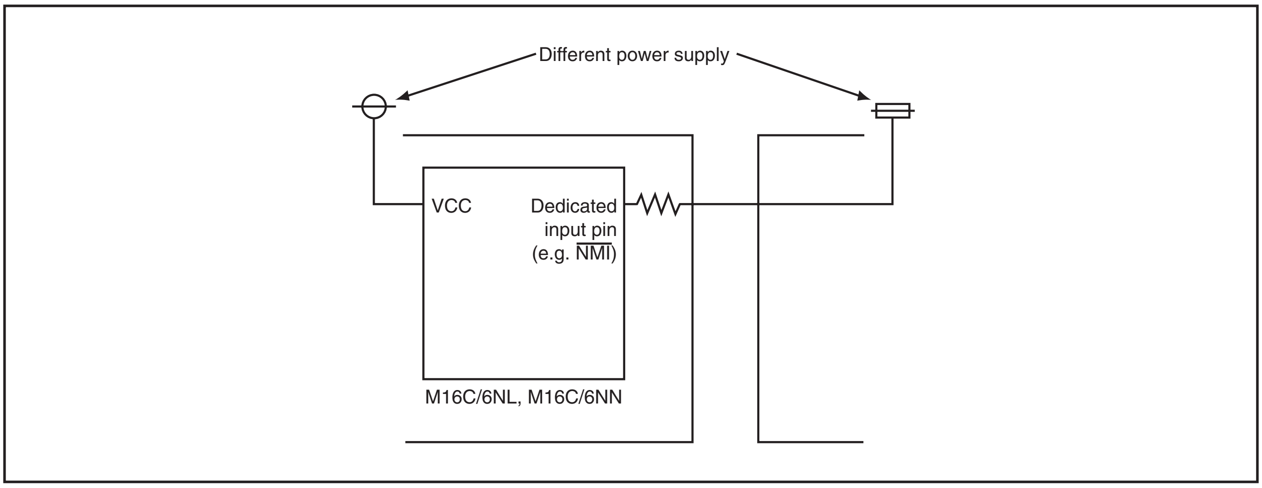


Figure 23.8 Circuit Connection

23.17 Electrical Characteristic Differences between Mask ROM and Flash Memory Version MCUs

Flash memory version and mask ROM version may have different characteristics, operating margin, noise tolerated dose, noise width dose in electrical characteristics due to internal ROM, different layout pattern, etc. When switching to the mask ROM version, conduct equivalent tests as system evaluation tests conducted in the flash memory version.

23.18 Mask ROM Version

When using the masked ROM version, write nothing to internal ROM area.

23.19 Flash Memory Version

23.19.1 Functions to Prevent Flash Memory from Rewriting

ID codes are stored in addresses 0FFFDFh, 0FFFE3h, 0FFFEBh, 0FFFEFh, 0FFFF3h, 0FFFF7h, and 0FFFFBh. If wrong data are written to theses addresses, the flash memory cannot be read or written in standard serial I/O mode and CAN I/O mode.

The ROMCP register is mapped in address 0FFFFFh. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of MCU, these addresses are allocated to the vector addresses (H) of fixed vectors.

23.19.2 Stop Mode

When entering stop mode, execute the instruction which sets the CM10 bit to 1 (stop mode) after setting the FMR01 bit to 0 (CPU rewrite mode disabled) and disabling the DMA transfer.

23.19.3 Wait Mode

When entering wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

23.19.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode

If the CM05 bit is set to 1 (main clock stopped), do not execute the following commands:

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program
- Read lock bit status

23.19.5 Writing Command and Data

Write commands and data to even addresses in the user ROM area.

23.19.6 Program Command

By writing xx40h in the first bus cycle and data to the write address in the second bus cycle, an auto-program operation (data program and verify) will start. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.

23.19.7 Lock Bit Program Command

By writing xx77h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, the lock bit for the specified block is set to 0. The address value specified in the first bus cycle must be the same highest-order even address of a block specified in the second bus cycle.

23.19.8 Operating Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), set the CM11 bit in the CM1 register to 0 (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register. Also, set the PM17 bit in the PM1 register to 1 (with wait state).

23.19.9 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

23.19.10 Interrupts

EW0 Mode

To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.

- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available since registers FMR0 and FMR1 are forcibly reset when either interrupt request is generated. Allocate the jump addresses for each interrupt service routines to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ or watchdog timer interrupt request is generated. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

EW1 Mode

- Do not acknowledge any interrupts with vectors in the relocatable vector table or address match interrupt during auto-programming or auto-erasure.
- Do not use the watchdog timer interrupt.
- The $\overline{\text{NMI}}$ interrupt is available since registers FMR0 and FMR1 are forcibly reset when the interrupt request is generated. Allocate the jump address for the interrupt service routine to the fixed vector table. Flash memory rewrite operation is suspended when the $\overline{\text{NMI}}$ interrupt request is generated. Execute the rewrite program again after exiting the interrupt service routine.

23.19.11 How to Access

To set the FMR01, FMR02, or FMR11 bit to 1, write 1 after first setting the bit to 0. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to 0 and the instruction to set the bit to 1. Set the bit while an “H” signal is applied to the $\overline{\text{NMI}}$ pin.

23.19.12 Rewriting in User ROM Area

EW0 Mode

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory cannot be rewritten because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area while in standard serial I/O mode, parallel I/O mode, or CAN I/O mode.

EW1 Mode

Avoid rewriting any block in which the rewrite control program is stored.

23.19.13 DMA Transfer

In EW1 mode, do not perform a DMA transfer while the FMR00 bit in the FMR0 register is set to 0 (auto-programming or auto-erasure).

23.20 Flash Memory Programming Using Boot Program

When programming the internal flash memory using boot program, be careful about the pins state and connection as follows.

23.20.1 Programming Using Serial I/O Mode

CTX0 pin : This pin automatically outputs “H” level.

CRX0 pin : Connect to CAN transceiver or connect via resister to VCC (pull-up)

Figure 23.9 shows the Pin Connection for Programming Using Serial I/O Mode.

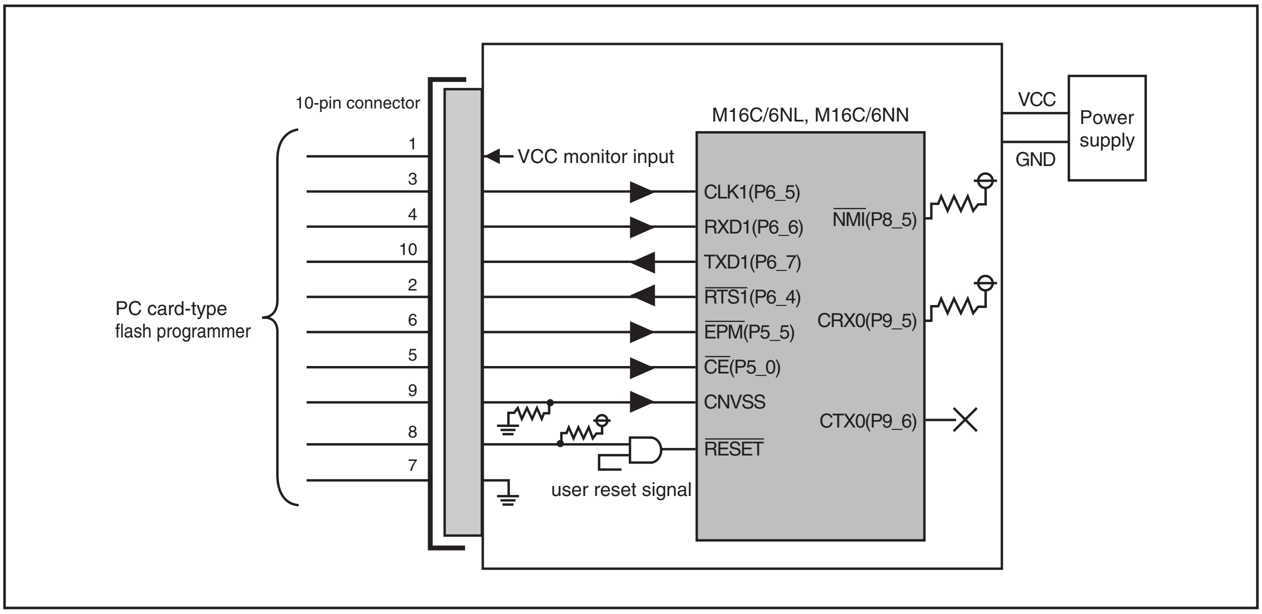


Figure 23.9 Pin Connection for Programming Using Serial I/O Mode

23.20.2 Programming Using CAN I/O Mode

$\overline{\text{RTS1}}$ pin : This pin automatically outputs “H” and “L” level.

Figure 23.10 shows the Pin Connection for Programming Using CAN I/O Mode.

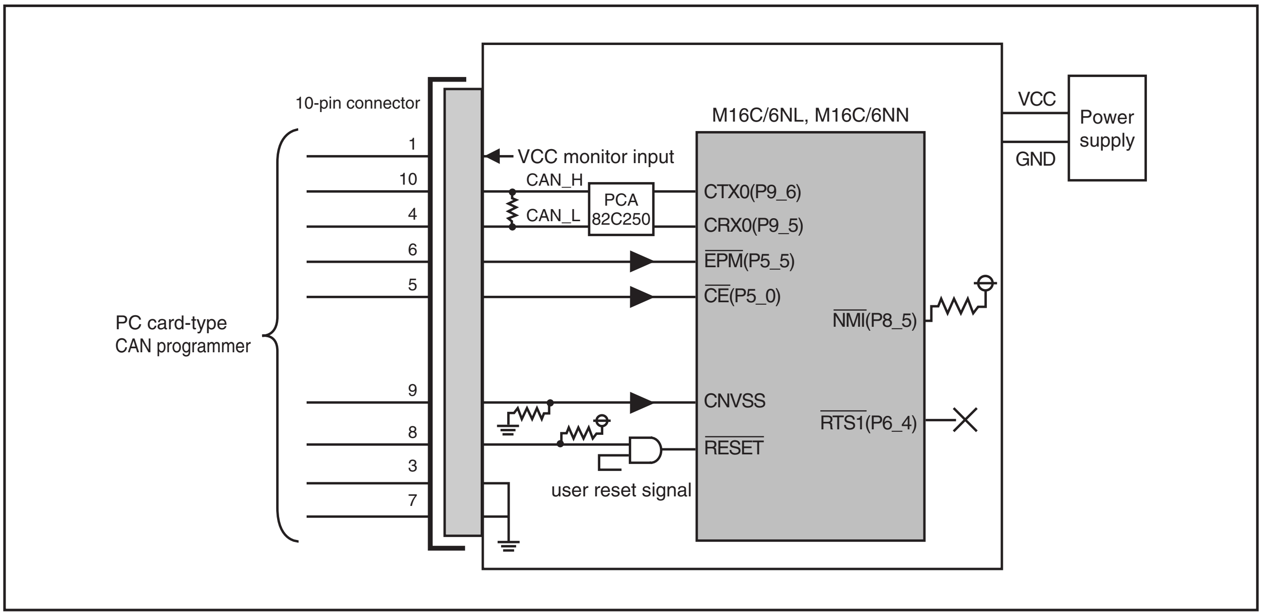


Figure 23.10 Pin Connection for Programming Using CAN I/O Mode

23.21 Noise

Connect a bypass capacitor (approximately 0.1 μF) across pins VCC1 and VSS, and pins VCC2 and VSS using the shortest and thicker possible wiring.

Figure 23.11 shows the Bypass Capacitor Connection.

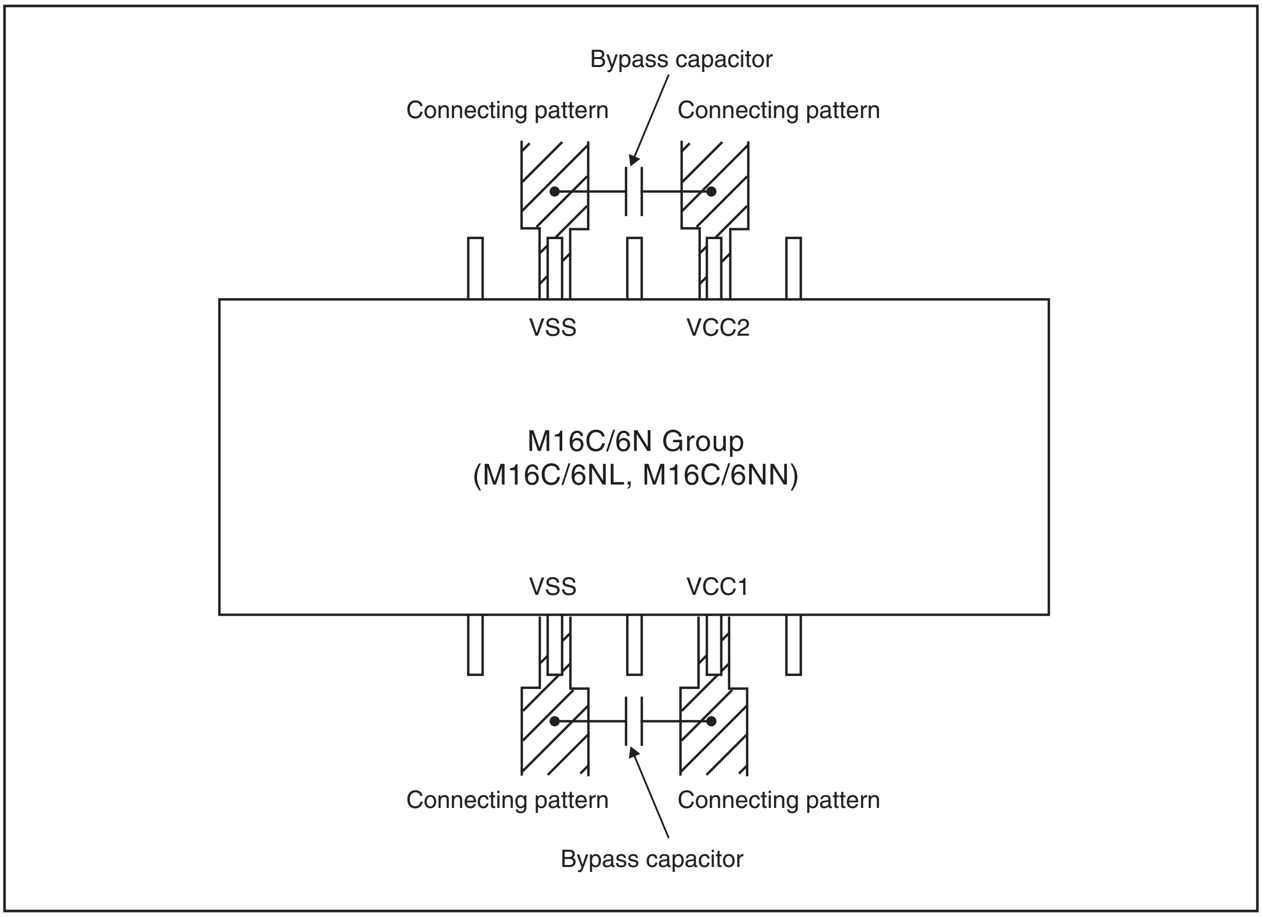
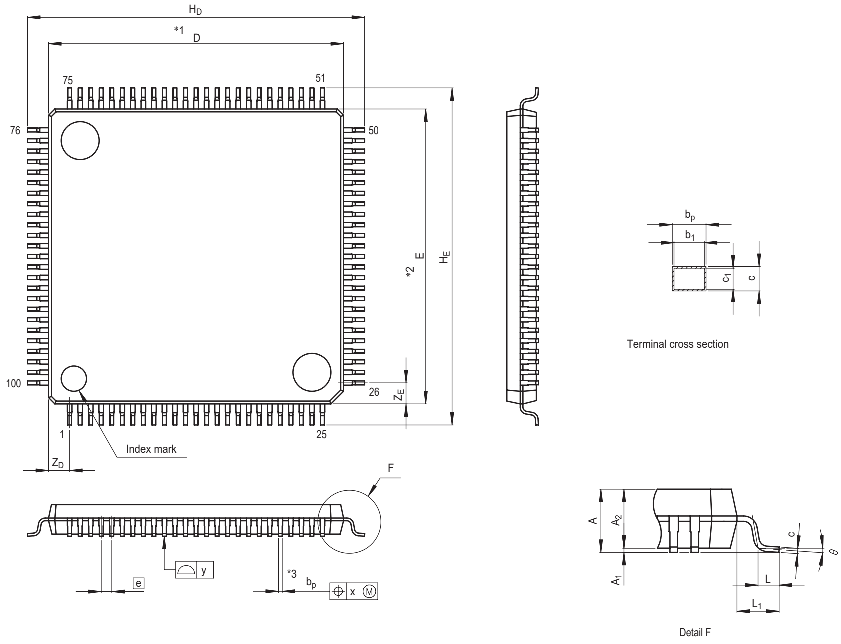


Figure 23.11 Bypass Capacitor Connection

Appendix 1. Package Dimensions

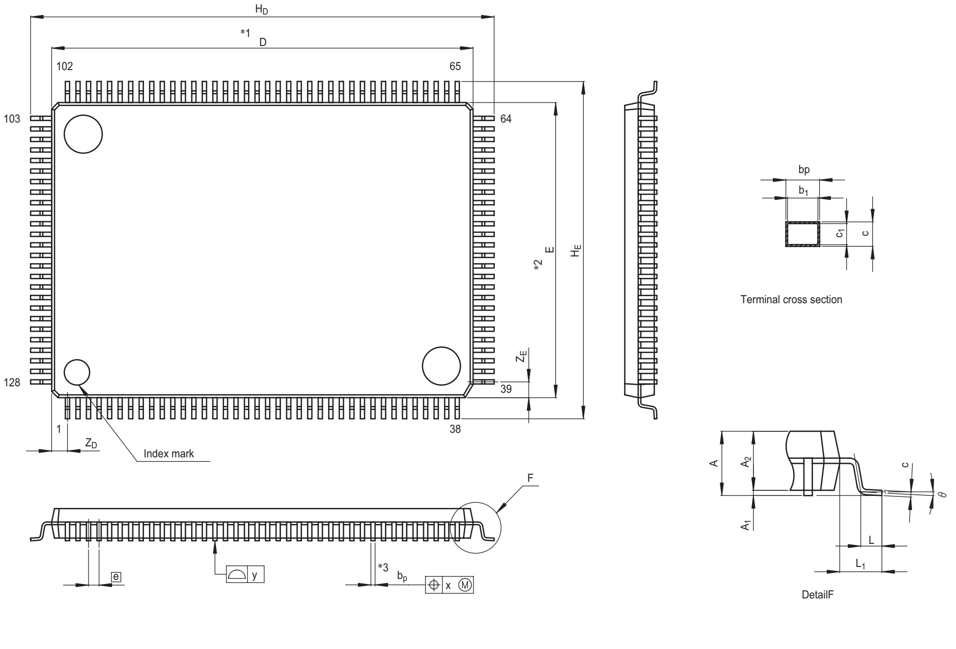
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP128-14x20-0.50	PLQP0128KB-A	128P6Q-A	0.9g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.125	0.2
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.10
y	—	—	0.10
Z_D	—	0.75	—
Z_E	—	0.75	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Memo

Register Index

REVISION HISTORY	M16C/6N Group (M16C/6NL, M16C/6NN) Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 30, 2004	–	First edition issued
1.01	Nov. 01, 2004	–	Revised edition issued * Revised parts and revised contents are as follows (except for expressional change).
		267	Table 21.2 Recommended Operating Conditions (1) • $I_{OH(peak)}$: Unit is revised from “V” to “mA”.
		268	Table 21.3 Recommended Operating Conditions (2) • NOTE 3: “VCC = 3.0 ± 0.3 V” is revised to “VCC = 3.3 ± 0.3 V”.
		288	22.9.1.2 Timer A (Event Counter Mode) is revised.
1.02	Jul. 01, 2005	–	Revised edition issued * Revised parts and revised contents are as follows (except for expressional change).
		5	Table 1.3 Product List is revised.
		13	FIgure 4.1 SFR Information (1): The value of After Reset in CM2 Register is revised.
		19	Figure 4.7 SFR Information (7): NOTE 1 is revised.
		35	Figure 7.4 CM2 Register: The value of After Reset is revised.
		51	Figure 7.13 State Transition in Normal Operation Mode: NOTE 7 is revised.
		74	9.10 Address Match Interrupt: After of 13th line • “Note that when using the external bus in 8-bit width, no address match interrupts can be used for external areas.” is deleted.
		172	Figure 14.37 (upper) SiC Register: NOTE 4 is revised.
		203	Figure 18.6 C0MCTLj Registers • RemActive bit: Function is revised. • RspLock bit: Bit Name is revised. • NOTE 2 is revised.
		204	Figure 18.7 C0CTLR Registers (upper) • LoopBack bit: The expression of Function is revised. • BasicCAN bit: The expression of Function is revised. Figure 18.7 C0CTLR Registers (lower) • TSPreScale bit: Bit Symbol is revised. (“Bit1, Bit0” is deleted.) • TSReset bit: The expression of Function is revised. • RetBusOff bit: The expression of Function is revised. • RXOnly bit: The expression of Function is revised.
		206	Figure 18.9 C0STR Registers (upper): NOTE 1 is deleted. Figure 18.9 C0STR Registers (lower) • State_LoopBack bit: The expression of Function is revised. • State_BasicCAN bit: The expression of Function is revised.
		209	Figure 18.12 C0RECR Register, C0TECR Register, C0TSR Register and C0AFS Register • C0RECR Register: NOTE 2 is deleted. • C0TECR Register: NOTE 1 is deleted. • C0TSR Register: NOTE 1 is deleted.
		220	18.15.1 Reception (1): “(refer to 18.15.2 Transmission)” is deleted.
		225	Figure 19.1 I/O Ports (1): “P7_0” in 4th figure is deleted.
		227	Figure 19.3 I/O Ports (3): “P7_0” is added to middle figure.
		229	Figure 19.6 I/O Pins: NOTE 1 is deleted.

Rev.	Date	Description	
		Page	Summary
1.02	Jul. 01, 2005	269	Table 21.4 Electrical Characteristics (1)
			• Measuring Condition of V_{OL} is revised from “L_{OL} = –200µA” to “L_{OL} = 200µA”.
		270	Table 21.5 Electrical Characteristics (2): Mask ROM (5th item)
			• “f(XCIN)” is changed to “(f(BCLK)).
		271	Table 21.6 A/D Conversion Characteristics: “Tolerance Level Impedance” is deleted.
		304	22.14 Programmable I/O Ports: last 1 to 2 lines
			• (1) Setting Procedure is revised from “#00010000b” to “#00000001b”.
			• (2) Setting Procedure is revised from “#00010011b” to “#00110001b”.
2.00	Nov. 28, 2005	–	Revised edition issued
			* Memory expansion and microprocessor modes are added.
			* Revised parts and revised contents are as follows (except for expressional change).
		2	Table 1.1 Performance Outline (100-pin version): Operation Mode is revised.
		3	Table 1.2 Performance Outline (128-pin version): Operation Mode is revised.
		5	Table 1.3 Product List: NOTE 1 is added.
		6	Figure 1.3 Pin Configuration (1): Bus control pins are added.
		7, 8	Tables 1.4 and 1.5 Pin Characteristics in 100-pin version (1)(2) are added.
		9	Figure 1.4 Pin Configuration (2): Bus control pins are added.
		10 to 12	Tables 1.6 to 1.8 Pin Characteristics in 128-pin version (1)(2)(3) are added.
		13 to 15	Tables 1.8 to 1.10 Pin Description (1)(2)(3) are revised.
		18	3. Memory: Last sentence (In memory expansion ...) is added.
			Figure 3.1 Memory Map: NOTES 1 and 2 are added.
		19	Table 4.1 SFR Information (1)
			• Value of After Reset in PM0 is revised.
			• CSR Register is added to 0008h.
			• CSE Register is added to 001Bh.
			• NOTE 1 is added.
		30	Table 4.12 SFR Information (12)
			• Value of After Reset in PUR1 is revised.
			• NOTE 1 is added.
		31 to 33	5. Reset: Layout is changed.
		32	Figure 5.2 Reset Sequence is revised.
		32	Table 5.1 Pin Status When RESET Pin Level is “L” is revised.
		33	5.2 Software Reset, 5.3 Watchdog Timer Reset, 5.4 Oscillation Stop Detection Reset: Last sentence (Processor mode remains ...) is added to each section.
		33	5.5 Internal Space is added.
		34	6.1 Types Processor Mode and 6.2 Setting Processor Mode are added.
			Table 6.1 Features of Processor Modes, Table 6.2 Processor Mode After Hardware Reset and Table 6.3 PM01 to PM00 Bits Set Values and Processor Modes are added.
		35	Figure 6.1 PM0 Register is revised.
		36	Figure 6.2 PM1 Register is revised.
		38, 39	Figures 6.4 to 6.7 Memory Map and $\overline{CS}$ Area in Memory Expansion Mode and Microprocessor Mode (1) to (4) are added.
		40 to 50	7. Bus is added.

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 28, 2005	59	Figure 8.9 Examples of Main Clock Connection Circuit is revised.
		60	Figure 8.10 Examples of Sub Clock Connection Circuit is revised.
		61	8.1.4 PLL Clock • 9th line: The sentence (When the PLL ... to) is added.
		63	8.2.1 CPU Clock and BCLK • 10th line: The sentence (During memory expansion ...) is added.
		65	8.4.1.6 On-chip Oscillator Mode: Last sentence (When the operation mode is ...) is added.
			8.1.1.7 On-chip Oscillator Low Power Dissipation Mode: Last sentence (When the operation mode is ...) is deleted.
		66	Table 8.4 Pin Status During Wait Mode is revised.
		68	Table 8.6 Interrupts to Stop Mode and Use Conditions is added.
			Table 8.7 Pin Status in Stop Mode is revised.
		71	Figure 8.13 State Transition in Normal Operation Mode: NOTE 7 is deleted.
		82	Figure 10.4 Interrupt Control Registers (2): NOTE 2 is added.
		87	10.5.8 Returning from an Interrupt Routine: Las sentence (Register bank ...) is added.
			10.5.9 Interrupt Priority: First sentence (If two or more...) is revised.
			10.5.10 Interrupt Priority Resolution Circuit: First sentence (The interrupt priority level ...) is revised.
		91	Figure 10.12 IFSR1 Register: NOTES 2 and 4 are revised.
		94	10.10 Address Match Interrupt • Second line from the bottom: sentence (Note that when ...) is added.
		99	Table 12.1 DMAC Specifications: DMA transfer Cycles is added.
		103	12.1 Transfer Cycle: 3rd and 4th sentences (During ... /Furthermore ...) are revised.
			12.1.2 Effect of BYTE Pin Level is added.
			12.1.3 Effect of Software Wait: 3rd to 9th lines is moved from next section of 12.1.2.
			12.1.4 Effect of $\overline{RDY}$ Signal is added.
		105	Table 12.2 DMA Transfer Cycles is revised.
			Table 12.3 Coefficient j, k is revised.
		107	12.5 Channel Priority and DMA Transfer Timing: Last sentence (Refer to ...) is added.
		123	Figure 13.12 TA0MR to TA4MR Registers in PWM Mode: b2 is revised from “1” to “(blank)”.
		134	Figure 14.1 Three-Phase Motor Control Timer Function Block Diagram is revised.
		135	Figure 14.2 UNVC0 Register: NOTES 5 and 6 are revised.
		148	Figure 15.5 U0BRG to U2BRG Registers (lower): NOTE 3 is added.
		149	Figure 15.6 U0C0 to U2C0 Registers (lower): NOTE 5 is added.
		166	Table 15.9 Example of Bit Rates and Settings: 20 MHz is added.
		192	Figure 15.37 SiC Register (upper): NOTE 7 is added.
			Figure 15.37 SiBRG Register (middle): NOTE 4 is added.
		198	Figure 16.1 A/D Converter Block Diagram • ADGSEL1 to ADGSEL0 (right/lower) is revised from “10b” to “11b”. • NOTE 1 is added.
		212	16.2.6 Output Impedance of Sensor under A/D Conversion • 10th line: f(XIN) is revised to f(ϕAD).
		213	Figure 16.10 Analog Input Pin and External Sensor Equivalent Circuit • fAD is revised to ϕAD.

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 28, 2005	214	Figure 17.1 D/A Converter Block Diagram is revised.
		215	Figure 17.2 DA0 and DA1 Registers: Setting Range is added. Figure 17.3 D/A Converter Equivalent Circuit: NOTE 2 is added.
		217	Figure 18.3 CRC Calculation is partly revised.
		229	Figure 19.12 C0TECR Register (2nd register): NOTE 1 is added.
		240	19.15.1 Reception: (5) is partly revised.
		243	20. Programmable I/O Ports • 8th line (Each pin functions ...) is partly revised. • Last sentence (When using ...) is added.
		244	20.1 PDi Register • 4th line: The sentence (During memory expansion ...) is added. 20.2 Pi Register • 9th line: The sentence (During memory expansion ...) is added. 20.3 PURj Register • 5th line: The sentence (However, the pull-up ...) is added.
		250	Figure20.7 PD0 to PD13 Registers: NOTE 2 is added.
		251	Figure20.8 Pi Registers (upper): NOTE 2 is added.
		252	Figure20.9 PUR0 Register (upper): NOTE 1 is added. Figure20.9 PUR1 Register (middle): NOTES 1 to 3 are added.
		254	Table 20.3 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode is added.
		255	Figure 20.12 Unassigned Pins Handling • Figure of memory expansion mode or microprocessor mode is added. • NOTE 1 is added.
		256	Table 21.2 Flash Memory Rewrite Modes Overview • Operation Mode of CPU Rewrite Mode is revised. • NOTE 2 is revised.
		257	21.1 Memory Map: 2nd sentence (The user ROM ...) is revised.
		259	Figure 21.2 ROMCP Register is revised.
		260	Table 21.3 EW0 Mode and EW1 Mode • Flash Memory Status Detection of EW0 Mode is revised. • NOTES 1 and 2 are revised.
		261	21.3.2 EW1 Mode: Last sentence (When an erase/program ...) is added.
		263	21.3.3.4 FMSTP Bit • 8th line: Procedure to change the FMSTP bit setting (1) to (4) are added.
		265	Figure 21.5 Setting and Resetting of EW0 Mode • First frame: “memory expansion mode” is added. • NOTE 5 is revised. Figure 21.6 Setting and Resetting of EW1 Mode: NOTE 1 is revised.
		266	Figure 21.7 Processing Before and After Low Power Dissipation Mode or On-chip Oscillator Low Power Dissipation Mode: • Title, First and second frames (left) and top of right: “on-chip oscillator low power dissipation mode” is added.

REVISION HISTORY	M16C/6N Group (M16C/6NL, M16C/6NN) Hardware Manual

Rev.	Date	Description	
		Page	Summary
2.00	Nov. 28, 2005	272	21.3.4.11 Stop Mode is revised.
			21.3.4.12 Low Power Dissipation Mode and On-chip Oscillator Low Power Dissipation Mode is partly revised.
		271	21.3.5.5 Block Erase Command: Last sentence (Also execute ...) is added.
			Figure 21.9 Block Erase Command: NOTES 2 and 3 are added.
		277	Figure 21.12 Full Status Check and Handling Procedure for Each Error
			• Erase error: (4) is added.
		279	Table 21.7 Pin Functions for Standard Serial I/O Mode
			• Description of VCC1, VCC2, VSS is revised.
			• Description of P8_4 is revised.
			• NOTE 1 is revised.
			• NOTE 2 is added.
		282	Figures 21.15 and 21.16 Circuit Application in Serial I/O Mode 1/2
			• “VCC1” and “VCC2” are added.
		284	Table 21.8 Pin Functions for CAN I/O Mode
			• Description of VCC1, VCC2, VSS is revised.
			• Description of P8_4 is revised.
			• NOTE 1 is added.
		287	Figure 21.19 Circuit Application in CAN I/O Mode: “VCC1” and “VCC2” are added.
		289	Table 22.2 Recommended Operating Conditions (1) is partly revised.
		291	Table 22.4 Electrical Characteristics (1)
			• V_{T+} - V_{T-}: $\overline{\text{HOLD}}$ and $\overline{\text{RDY}}$ are added.
		295	Table 22.12 Memory Expansion Mode and Microprocessor Mode is added.
		298 to 300	Switching Characteristics are added.
		302 to 308	Figures 22.5 to 22.11 Timing Diagram (2) to (8) are added.
		309 to 323	Characteristics of 3.3 V version are added.
		325	23.2 External Bus is added.
		328	23.5 Power Control: 4th and 5th items (When entering wait mode ... / When entering stop mode ...) are revised.
		346	Figure 23.4 Use of Capacitors to Reduce Noise is partly revised.
		347	23.13 A/D Converter: Last item (The applied intermediate ...) is added.
		353	23.15 Programmable I/O Ports: 5th and 6th items (Indeterminate values ... / When the PM01 ...) are added.
		357	23.19.2 Stop Mode is revised.
			23.19.4 Low Power Dissipation Mode and On-Chip Oscillator Low Power Dissipation Mode is partly revised.
			23.19.8 Operation Speed is revised.
2.10	Apr.14, 2006	–	Revised edition issued
			* Revised parts and revised contents are as follows (except for expressional change).
		5	Table 1.3 Product Information: NOTE 2 is added.
		26	Table 4.8 SFR Information (8)
			• The value of After Reset in IDB0 register is revised.
			• The value of After Reset in IDB1 register is revised.

Rev.	Date	Description	
		Page	Summary
2.10	Apr.14, 2006	45	Table 7.5 MCU Status in Hold State • Item: “P10” is revised to “P14 (3)”. • NOTE 3 is added.
		70	Figure 8.12 State Transition to Stop Mode and Wait Mode is revised.
		103	12.1.3 Effect of Software Wait: 3rd to 9th lines (Figure 12.5 shows ... required.) is moved to next section of 12.1.4.
		114	Figure 13.7 Registers TA0MR to TA4MR in Timer Mode: NOTE 2 is added.
		121	Figure 13.11 Registers TA0MR to TA4MR in One-shot Timer Mode: NOTE 3 is added.
		123	Figure 13.12 Registers TA0MR to TA4MR in Pulse Width Modulation Mode: NOTE 4 is added.
		128	Figure 13.18 Registers TB0MR to TB5MR in Timer Mode: NOTE 1 is added.
		131	Figure 13.20 Registers TA0MR to TA4MR in Pulse Period and Pulse Width Measurement Mode: NOTE 2 is added.
		136	Figure 14.3 INVC1 Register: NOTE 6 is added.
		137	Figure 14.4 Registers IDB0 and IDB1 (upper): The value of After Reset is revised.
		141	Figure 14.8 Registers TA1MR, TA2MR, TA4MR (upper): NOTE 1 is added. Figure 14.8 TB2MR Register (lower): NOTE 1 is added.
		145, 146	Figures 15.1 to 15.3 are revised.
		148	Figure 15.5 Registers U0RB to U2RB (middle): NOTE 3 is added.
		149	Figure 15.6 Registers U0C0 to U2C0 (lower): NOTE 6 is added.
		154	Table 15.1 Clock Synchronous Serial I/O Mode Specifications • Transfer clock: “fj/2(n+1)” is revised to “fj/(2(n+1))”. • Note 3 is revised.
		157	Figure 15.11 Transmit and Receive Operation is revised.
		162	Table 15.5 UART Mode Specifications • Transfer clock: “fj/16(n+1)” is revised to “fj/(16(n+1))” and “fEXT/16(n+1)” is revised to “fEXT/(16(n+1))” . • Note 2 is revised.
		165	Figure 15.17 Transmit Operation is revised.
		166	Table 15.9 Example of Bit Rates and Settings: “Actual Time” is revised to “Bit Rate”.
		170	Table 15.10 I^{2}C Mode Specifications • Transfer clock: “fj/2(n+1)” is revised to “fj/(2(n+1))”.
		172	Table 15.11 Registers to Be Used and Settings in I^{2}C Mode: NOTE 3 is added.
		179	Table 15.14 Special Mode 2 Specifications • Transfer clock: “fj/2(n+1)” is revised to “fj/(2(n+1))”.
		186	Table 15.17 SIM Mode Specifications • Transfer clock: “fj/16(n+1)” is revised to “fj/(16(n+1))” and “fEXT/16(n+1)” is revised to “fEXT/(16(n+1))”.
		188	Figure 15.32 Transmit and Receive Timing in SIM Mode is revised.
		190	15.1.6.2 Format is revised.
		192	Figure 15.37 SiC Register (upper): NOTE 8 is added.
		194	Table 15.19 SI/Oi Specifications • Transfer clock: “fj/2(n+1)” is revised to “fj/(2(n+1))”.

Rev.	Date	Description	
		Page	Summary
2.10	Apr.14, 2006	195	Figure 15.39 SI/Oi Operation Timing: Cycle and Note 1 is revised. (1.5 -> 0.5 to 1.0)
		196	15.2.3 Functions for Setting SOUTi Initial Value: 2nd item (However...) is added.
		215	Figure 17.3 D/A Converter Equivalent Circuit is revised.
		224	Figure 19.7 C0CTLR Register (upper): NOTE 4 is added.
		229	Figure 19.11 C0TSR Register (3rd register): NOTE 1 is added.
		230	Figure 19.12 Transition between Operational Modes is revised.
		231	19.5.3 CAN Sleep Mode
			• 1st item: “and Reset bit to 0” is deleted.
		234	Table 19.2 Examples of Bit-rate is revised.
		254	Table 20.3 Unassigned Pin Handling in Memory expansion Mode and Microprocessor Mode
			• Pin Name: “P0 to P7” is revised to “P6, P7”.
		291	Table 22.4 Electrical Characteristics (1): Hysteresis XIN is deleted.
		309	Table 22.28 Electrical Characteristics: Hysteresis XIN is deleted.
		328	23.5 Power Control
			• 5th item: Notes when entering stop mode is revised.
		329	• 6th item: Notes is added.
		346	23.13 A/D Converter
			• 1st item: “After stopping ...” is added.

M16C/6N Group (M16C/6NL, M16C/6NN) Hardware Manual

Publication Data : Rev.1.00 Sep 30, 2004
Rev.2.10 Apr 14, 2006

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

M16C/6N Group (M16C/6NL, M16C/6NN)
Hardware Manual

Renesas Electronics Corporation