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/
32

M32C/8B Group

Hardware Manual

RENESAS MCU
M16C FAMILY / M32C/80 SERIES

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

Renesas Electronics
www.renesas.com

Rev.1.00 2009.11

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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 part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part 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 M32C/8B Group. 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	M32C/8B Group Datasheet	REJ03B0242-0100
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.	M32C/8B Group Hardware Manual	This hardware manual
Software manual	Description of CPU instruction set	M32C/80 Series Software Manual	REJ09B0319-0100
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

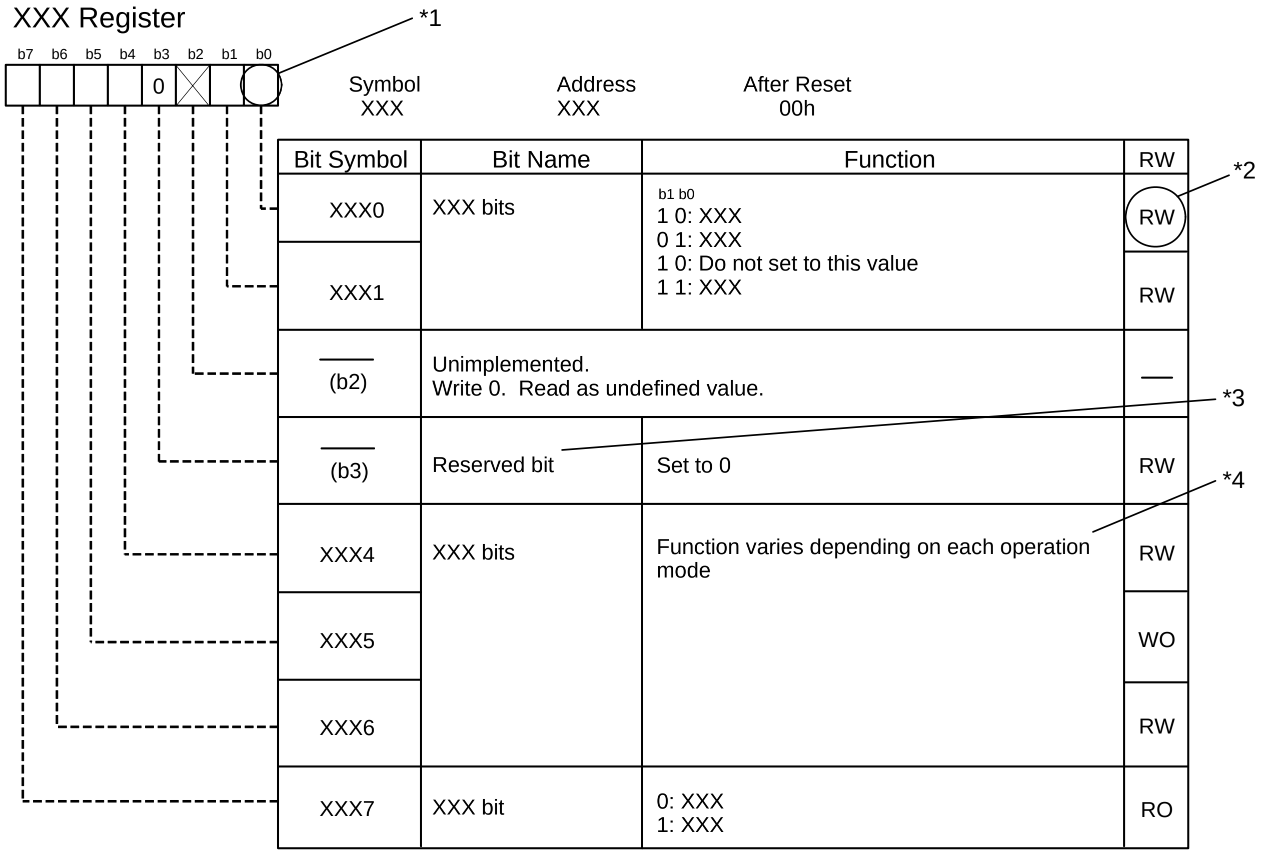
The indication "b" is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication "h" is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b
Hexadecimal: EFA0h
Decimal: 1234

3. Register Notation

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

XXX Register



Symbol	Address	After Reset
XXX	XXX	00h

Bit Symbol	Bit Name	Function	RW
XXX0	XXX bits	b1 b0 1 0: XXX 0 1: XXX 1 0: Do not set to this value 1 1: XXX	RW
XXX1			RW
— (b2)	Unimplemented. Write 0. Read as undefined value.		—
— (b3)	Reserved bit	Set to 0	RW
XXX4	XXX bits	Function varies depending on each operation mode	RW
XXX5			WO
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: Unimplemented.

*2

RW: Read and write.
RO: Read only.
WO: Write only.
–: Unimplemented.

*3

- Reserved bit

  Reserved bit. Set to specified value.

*4

- Unimplemented

  Nothing is implemented to the bit. As the bit may be used for future functions, if necessary, set to 0.
- Do not set to a value

  Operation is not guaranteed when a value is set.
- Function varies according to the 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

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0	PM0	41
0005h	Processor Mode Register 1	PM1	42
0006h	System Clock Control Register 0	CM0	67, 118
0007h	System Clock Control Register 1	CM1	68
0008h			
0009h	Address Match Interrupt Enable Register	AIER	115
000Ah	Protect Register	PRCR	94
000Bh	External Data Bus Width Control Register	DS	44
000Ch	Main Clock Division Register	MCD	69
000Dh	Oscillation Stop Detection Register	CM2	70
000Eh	Watchdog Timer Start Register	WDTS	119
000Fh	Watchdog Timer Control Register	WDC	119
0010h	Address Match Interrupt Register 0	RMAD0	115
0011h			
0012h			
0013h	Processor Mode Register 2	PM2	72
0014h	Address Match Interrupt Register 1	RMAD1	115
0015h			
0016h			
0017h	Reference Voltage Configuration Register	DVCR	38
0018h	Address Match Interrupt Register 2	RMAD2	115
0019h			
001Ah			
001Bh	Voltage Monitor Register	LVDC	37
001Ch	Address Match Interrupt Register 3	RMAD3	115
001Dh			
001Eh			
001Fh	Voltage Regulator Control Register	VRCR	74
0020h			
0021h			
0022h			
0023h			
0024h			
0025h			
0026h	PLL Control Register 0	PLC0	71
0027h			
0028h	Address Match Interrupt Register 4	RMAD4	115
0029h			
002Ah			
002Bh			
002Ch	Address Match Interrupt Register 5	RMAD5	115
002Dh			
002Eh			
002Fh			
0030h			
0031h			
0032h			
0033h			
0034h			
0035h			
0036h			
0037h			
0038h	Address Match Interrupt Register 6	RMAD6	115
0039h			
003Ah			
003Bh			
003Ch	Address Match Interrupt Register 7	RMAD7	115
003Dh			
003Eh			
003Fh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h			
0048h	External Space Wait Control Register 0	EWCR0	50
0049h	External Space Wait Control Register 1	EWCR1	50
004Ah	External Space Wait Control Register 2	EWCR2	50
004Bh	External Space Wait Control Register 3	EWCR3	50
004Ch	Page Mode Wait Control Register 0	PWCR0	62
004Dh	Page Mode Wait Control Register 1	PWCR1	63
004Eh			
004Fh			
0050h	Flash Memory Control Register 3	FMR3	302
0051h			
0052h	Flash Memory Control Register 2	FMR2	302
0053h			
0054h			
0055h	Flash Memory Control Register 1	FMR1	303
0056h			
0057h	Flash Memory Control Register 0	FMR0	303
0058h			
0059h	Flash Memory Control Register 4	FMR4	74
005Ah			
005Bh			
005Ch			
005Dh			
005Eh			
005Fh			
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h	DMA0 Control Register	DM0IC	103
0069h	Timer B5 Interrupt Control Register	TB5IC	
006Ah	DMA2 Control Register	DM2IC	
006Bh	UART2 Receive/ACK Interrupt Control Register	S2RIC	
006Ch	Timer A0 Interrupt Control Register	TA0IC	
006Dh	UART3 Receive/ACK Interrupt Control Register	S3RIC	
006Eh	Timer A2 Interrupt Control Register	TA2IC	
006Fh	UART4 Receive/ACK Interrupt Control Register	S4RIC	
0070h	Timer A4 Interrupt Control Register	TA4IC	
0071h	UART0/UART3 Bus Conflict Detection Interrupt Control Register	BCN0IC/ BCN3IC	
0072h	UART0 Receive/ACK Interrupt Control Register	S0RIC	
0073h	A/D0 Conversion Interrupt Control Register	AD0IC	
0074h	UART1 Receive/ACK Interrupt Control Register	S1RIC	
0075h			
0076h	Timer B1 Interrupt Control Register	TB1IC	103
0077h			
0078h	Timer B3 Interrupt Control Register	TB3IC	103
0079h			
007Ah	$\overline{\text{INT5}}$ Interrupt Control Register	INT5IC	104
007Bh			
007Ch	$\overline{\text{INT3}}$ Interrupt Control Register	INT3IC	104
007Dh			
007Eh	$\overline{\text{INT1}}$ Interrupt Control Register	INT1IC	104
007Fh			

Blank spaces are reserved. No access is allowed.

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0080h			
0081h			
0082h			
0083h			
0084h			
0085h			
0086h			
0087h			
0088h	DMA1 Interrupt Control Register	DM1IC	103
0089h	UART2 Transmit/NACK Interrupt Control Register	S2TIC	
008Ah	DMA3 Interrupt Control Register	DM3IC	
008Bh	UART3 Transmit/NACK Interrupt Control Register	S3TIC	
008Ch	Timer A1 Interrupt Control Register	TA1IC	
008Dh	UART4 Transmit/NACK Interrupt Control Register	S4TIC	
008Eh	Timer A3 Interrupt Control Register	TA3IC	
008Fh	UART2 Bus Conflict Detection Interrupt Control Register	BCN2IC	
0090h	UART0 Transmit/NACK Interrupt Control Register	S0TIC	
0091h	UART1/UART4 Bus Conflict Detection Interrupt Control Register	BCN1IC/BCN4IC	
0092h	UART1 Transmit Complete Interrupt Control Register	S1TIC	
0093h	Key Input Interrupt Control Register	KUPIC	
0094h	Timer B0 Interrupt Control Register	TB0IC	
0095h			
0096h	Timer B2 Interrupt Control Register	TB2IC	103
0097h			
0098h	Timer B4 Interrupt Control Register	TB4IC	103
0099h			
009Ah	$\overline{\text{INT4}}$ Interrupt Control Register	INT4IC	104
009Bh			
009Ch	$\overline{\text{INT2}}$ Interrupt Control Register	INT2IC	104
009Dh			
009Eh	$\overline{\text{INT0}}$ Interrupt Control Register	INT0IC	104
009Fh	Exit Priority Register	RLVL	105, 134
00A0h			
00A1h			
00A2h			
00A3h			
00A4h			
00A5h			
00A6h			
00A7h			
00A8h			
00A9h			
00AAh			
00ABh			
00ACh			
00ADh			
00AEh			
00AFh			
00B0h			
00B1h			
00B2h			
00B3h			
00B4h			
00B5h			
00B6h			
00B7h			
00B8h			
00B9h			
00BAh			
00BBh			
00BCh			
00BDh			
00BEh			
00BFh to 02BFh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
02C0h	X0 Register, Y0 Register	X0R, Y0R	276
02C1h			
02C2h	X1 Register, Y1 Register	X1R, Y1R	
02C3h			
02C4h	X2 Register, Y2 Register	X2R, Y2R	
02C5h			
02C6h	X3 Register, Y3 Register	X3R, Y3R	
02C7h			
02C8h	X4 Register, Y4 Register	X4R, Y4R	
02C9h			
02CAh	X5 Register, Y5 Register	X5R, Y5R	
02CBh			
02CCh	X6 Register, Y6 Register	X6R, Y6R	
02CDh			
02CEh	X7 Register, Y7 Register	X7R, Y7R	
02CFh			
02D0h	X8 Register, Y8 Register	X8R, Y8R	
02D1h			
02D2h	X9 Register, Y9 Register	X9R, Y9R	
02D3h			
02D4h	X10 Register, Y10 Register	X10R, Y10R	
02D5h			
02D6h	X11 Register, Y11 Register	X11R, Y11R	
02D7h			
02D8h	X12 Register, Y12 Register	X12R, Y12R	
02D9h			
02DAh	X13 Register, Y13 Register	X13R, Y13R	
02DBh			
02DCh	X14 Register, Y14 Register	X14R, Y14R	
02DDh			
02DEh	X15 Register, Y15 Register	X15R, Y15R	
02DFh			
02E0h	X/Y Control Register	XYC	276
02E1h			
02E2h			
02E3h			
02E4h	UART1 Special Mode Register 4	U1SMR4	202
02E5h	UART1 Special Mode Register 3	U1SMR3	201
02E6h	UART1 Special Mode Register 2	U1SMR2	200
02E7h	UART1 Special Mode Register	U1SMR	199
02E8h	UART1 Transmit/Receive Mode Register	U1MR	198
02E9h	UART1 Baud Rate Register	U1BRG	204
02EAh	UART1 Transmit Buffer Register	U1TB	206
02EBh			
02ECh	UART1 Transmit/Receive Control Register 0	U1C0	203
02EDh	UART1 Transmit/Receive Control Register 1	U1C1	204
02EEh	UART1 Receive Buffer Register	U1RB	206
02EFh			
02F0h			
02F1h			
02F2h			
02F3h			
02F4h	UART4 Special Mode Register 4	U4SMR4	202
02F5h	UART4 Special Mode Register 3	U4SMR3	201
02F6h	UART4 Special Mode Register 2	U4SMR2	200
02F7h	UART4 Special Mode Register	U4SMR	199
02F8h	UART4 Transmit/Receive Mode Register	U4MR	198
02F9h	UART4 Baud Rate Register	U4BRG	204
02FAh	UART4 Transmit Buffer Register	U4TB	206
02FBh			
02FCh	UART4 Transmit/Receive Control Register 0	U4C0	203
02FDh	UART4 Transmit/Receive Control Register 1	U4C1	204
02FEh	UART4 Receive Buffer Register	U4RB	206
02FFh			
0300h	Timer B3, B4, B5 Count Start Flag	TBSR	171
0301h			
0302h	Timer A11 Register	TA11	187
0303h			
0304h	Timer A21 Register	TA21	
0305h			
0306h	Timer A41 Register	TA41	
0307h			
0308h	Three-Phase PWM Control Register 0	INVC0	180
0309h	Three-Phase PWM Control Register 1	INVC1	181
030Ah	Three-Phase Output Buffer Register 0	IDB0	187
030Bh	Three-Phase Output Buffer Register 1	IDB1	187
030Ch	Dead Time Timer	DTT	186
030Dh	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	185
030Eh			
030Fh			

Blank spaces are reserved. No access is allowed.

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0310h	Timer B3 Register	TB3	170
0311h			
0312h	Timer B4 Register	TB4	
0313h			
0314h	Timer B5 Register	TB5	
0315h			
0316h			
0317h			
0318h			
0319h			
031Ah			
031Bh	Timer B3 Mode Register	TB3MR	167, 168, 169
031Ch	Timer B4 Mode Register	TB4MR	
031Dh	Timer B5 Mode Register	TB5MR	
031Eh			
031Fh	External Interrupt Source Select Register	IFSR	113, 205
0320h			
0321h			
0322h			
0323h			
0324h	UART3 Special Mode Register 4	U3SMR4	202
0325h	UART3 Special Mode Register 3	U3SMR3	201
0326h	UART3 Special Mode Register 2	U3SMR2	200
0327h	UART3 Special Mode Register	U3SMR	199
0328h	UART3 Transmit/Receive Mode Register	U3MR	198
0329h	UART3 Baud Rate Register	U3BRG	204
032Ah	UART3 Transmit Buffer Register	U3TB	206
032Bh			
032Ch	UART3 Transmit/Receive Control Register 0	U3C0	203
032Dh	UART3 Transmit/Receive Control Register 1	U3C1	204
032Eh	UART3 Receive Buffer Register	U3RB	206
032Fh			
0330h			
0331h			
0332h			
0333h			
0334h	UART2 Special Mode Register 4	U2SMR4	202
0335h	UART2 Special Mode Register 3	U2SMR3	201
0336h	UART2 Special Mode Register 2	U2SMR2	200
0337h	UART2 Special Mode Register	U2SMR	199
0338h	UART2 Transmit/Receive Mode Register	U2MR	198
0339h	UART2 Baud Rate Register	U2BRG	204
033Ah	UART2 Transmit Buffer Register	U2TB	206
033Bh			
033Ch	UART2 Transmit/Receive Control Register 0	U2C0	203
033Dh	UART2 Transmit/Receive Control Register 1	U2C1	204
033Eh	UART2 Receive Buffer Register	U2RB	206
033Fh			
0340h	Count Start Register	TABSR	152, 171, 188
0341h	Clock Prescaler Reset Register	CPSRF	73
0342h	One-Shot Start Register	ONSF	153
0343h	Trigger Select Register	TRGSR	151, 184
0344h	Up/Down Flag	UDF	150
0345h			
0346h	Timer A0 Register	TA0	149
0347h			
0348h	Timer A1 Register	TA1	149, 187
0349h			
034Ah	Timer A2 Register	TA2	149, 187
034Bh			
044Ch	Timer A3 Register	TA3	149
034Dh			
034Eh	Timer A4 Register	TA4	149, 187
034Fh			
0350h	Timer B0 Register	TB0	170
0351h			
0352h	Timer B1 Register	TB1	170
0353h			
0354h	Timer B2 Register	TB2	170, 186
0355h			
0356h	Timer A0 Mode Register	TA0MR	145, 146, 147, 148
0357h	Timer A1 Mode Register	TA1MR	
0358h	Timer A2 Mode Register	TA2MR	
0359h	Timer A3 Mode Register	TA3MR	
035Ah	Timer A4 Mode Register	TA4MR	
035Bh	Timer B0 Mode Register	TB0MR	167, 168, 169
035Ch	Timer B1 Mode Register	TB1MR	
035Dh	Timer B2 Mode Register	TB2MR	
035Eh	Timer B2 Special Mode Register	TB2SC	185
035Fh	Count Source Prescaler Register	TCSPR	73, 144

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0360h			
0361h			
0362h			
0363h			
0364h	UART0 Special Mode Register 4	U0SMR4	202
0365h	UART0 Special Mode Register 3	U0SMR3	201
0366h	UART0 Special Mode Register 2	U0SMR2	200
0367h	UART0 Special Mode Register	U0SMR	199
0368h	UART0 Transmit/Receive Mode Register	U0MR	198
0369h	UART0 Baud Rate Register	U0BRG	204
036Ah	UART0 Transmit Buffer Register	U0TB	206
036Bh			
036Ch	UART0 Transmit/Receive Control Register 0	U0C0	203
036Dh	UART0 Transmit/Receive Control Register 1	U0C1	204
036Eh	UART0 Receive Buffer Register	U0RB	206
036Fh			
0370h			
0371h			
0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h	DMA0 Request Source Select Register	DM0SL	122
0379h	DMA1 Request Source Select Register	DM1SL	
037Ah	DMA2 Request Source Select Register	DM2SL	
037Bh	DMA3 Request Source Select Register	DM3SL	
037Ch	CRC Data Register	CRCD	274
037Dh			
037Eh	CRC Input Register	CRCIN	274
037Fh			
0380h	A/D0 Register 0	AD00	257
0381h			
0382h	A/D0 Register 1	AD01	
0383h			
0384h	A/D0 Register 2	AD02	
0385h			
0386h	A/D0 Register 3	AD03	
0387h			
0388h	A/D0 Register 4	AD04	
0389h			
038Ah	A/D0 Register 5	AD05	
038Bh			
038Ch	A/D0 Register 6	AD06	
038Dh			
038Eh	A/D0 Register 7	AD07	
038Fh			
0390h			
0391h			
0392h	A/D0 Control Register 4	AD0CON4	257
0393h			
0394h	A/D0 Control Register 2	AD0CON2	255
0395h	A/D0 Control Register 3	AD0CON3	256
0396h	A/D0 Control Register 0	AD0CON0	253
0397h	A/D0 Control Register 1	AD0CON1	254
0398h	D/A Register 0	DA0	272
0399h			
039Ah	D/A Register 1	DA1	272
039Bh			
039Ch	D/A Control Register	DACON	272
039Dh			
039Eh			
039Fh			
03A0h			
03A1h			
03A2h			
03A3h			
03A4h			
03A5h			
03A6h			
03A7h			
03A8h			
03A9h			
03AAh			
03ABh			
03ACh			
03ADh			
03AEh			
03AFh	Function Select Register C	PSC	290

Blank spaces are reserved. No access is allowed.

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
03B0h	Function Select Register A0	PS0	286
03B1h	Function Select Register A1	PS1	286
03B2h	Function Select Register B0	PSL0	288
03B3h	Function Select Register B1	PSL1	288
03B4h	Function Select Register A2	PS2	287
03B5h	Function Select Register A3	PS3	287
03B6h	Function Select Register B2	PSL2	289
03B7h	Function Select Register B3	PSL3	289
03B8h			
03B9h			
03BAh			
03BBh			
03BCh			
03BDh			
03BEh			
03BFh			
03C0h	Port P6 Register	P6	285
03C1h	Port P7 Register	P7	285
03C2h	Port P6 Direction Register	PD6	284
03C3h	Port P7 Direction Register	PD7	284
03C4h	Port P8 Register	P8	285
03C5h	Port P9 Register	P9	285
03C6h	Port P8 Direction Register	PD8	284
03C7h	Port P9 Direction Register	PD9	284
03C8h	Port P10 Register	P10	285
03C9h	Port P11 Register	P11	285
03CAh	Port P10 Direction Register	PD10	284
03CBh	Port P11 Direction Register	PD11	284
03CCh	Port P12 Register	P12	285
03CDh	Port P13 Register	P13	285
03CEh	Port P12 Direction Register	PD12	284
03CFh	Port P13 Direction Register	PD13	284
03D0h	Port P14 Register	P14	285
03D1h	Port P15 Register	P15	285
03D2h	Port P14 Direction Register	PD14	284
03D3h	Port P15 Direction Register	PD15	284
03D4h			
03D5h			
03D6h			
03D7h			
03D8h			
03D9h			
03DAh	Pull-Up Control Register 2	PUR2	292
03DBh	Pull-Up Control Register 3	PUR3	293
03DCh	Pull-Up Control Register 4	PUR4	294
03DDh			
03DEh			
03DFh			
03E0h	Port P0 Register	P0	285
03E1h	Port P1 Register	P1	285
03E2h	Port P0 Direction Register	PD0	284
03E3h	Port P1 Direction Register	PD1	284
03E4h	Port P2 Register	P2	285
03E5h	Port P3 Register	P3	285
03E6h	Port P2 Direction Register	PD2	284
03E7h	Port P3 Direction Register	PD3	284
03E8h	Port P4 Register	P4	285
03E9h	Port P5 Register	P5	285
03EAh	Port P4 Direction Register	PD4	284
03EBh	Port P5 Direction Register	PD5	284
03ECh			
03EDh			
03EEh			
03EFh			
03F0h	Pull-Up Control Register 0	PUR0	291
03F1h	Pull-Up Control Register 1	PUR1	291
03F2h			
03F3h			
03F4h			
03F5h			
03F6h			
03F7h			
03F8h			
03F9h			
03FAh			
03FBh			
03FCh			
03FDh			
03FEh			
03FFh	Port Control Register	PCR	295

Blank spaces are reserved. No access is allowed.

1. Overview

1.1 Features

The M32C/8B Group is a single-chip control MCU, fabricated using high-performance silicon gate CMOS technology, embedding the M32C/80 Series CPU core. The M32C/8B Group is housed in 144-pin and 100-pin plastic molded LQFP packages.
With a 16-Mbyte address space, this MCU combines advanced instruction manipulation capabilities to process complex instructions by less bytes and execute instructions at higher speed.
The M32C/8B Group has a multiplier and DMAC adequate for office automation, communication devices and industrial equipment, and other high-speed processing applications.

1.1.1 Applications

- Audio-Visual equipment (e.g. televisions, audio components)
- Home Appliances (e.g. air conditioners, washing machines, sewing machines)
- Industrial equipment (e.g. programmable logic controllers)
- Computers and peripherals, cameras, etc.

1.1.2 Specifications

Tables 1.1 to 1.4 list the specifications of the M32C/8B Group.

Table 1.1 Specifications (144-Pin Package) (1/2)

Item	Function	Specification
CPU	Central processing unit	M32C/80 core (multiplier: 16 bits × 16 bits → 32 bits, multiply-addition operation instructions: 16 × 16 + 48 → 48 bits) • Basic instructions: 108 • Minimum instruction execution time: 31.3 ns (f(CPU) = 32 MHz / VCC1 = 3.0 to 5.5 V) • Operating modes: Single-chip mode, memory expansion mode, and microprocessor mode
Memory	ROM / RAM	Flash memory version: 256KB + 8KB/32 KB, 128KB + 8KB/32 KB ROMless version : – / 32KB
Power Supply Voltage Detection		Voltage monitor interrupt (optional)(1)
External Bus Expansion	Bus / memory expansion function	• Address space: 16 Mbytes • External bus interface: 1 to 7 wait states can be inserted, 4 chip select outputs, 3 V and 5 V interfaces • Bus format: Switchable between separate bus and multiplexed bus formats, switchable data bus width (8-bit or 16-bit)
Clock	Clock generation circuits	• 4 circuits: Main clock, sub clock, on-chip oscillator, PLL frequency synthesizer • Oscillation stop detection: Main clock oscillation stop detect function • Frequency divider circuit: Dividing ratio selectable among 1, 2, 3, 4, 6, 8, 10, 12, 14, 16 • Low power consumption features: Wait mode, stop mode
Interrupts		• Interrupt vectors: 70 • External interrupt inputs: 11 ($\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 6, Key input × 4) Single-chip mode Memory expansion and microprocessor mode with 8-bit external bus 8 ($\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 3, Key input × 4) Memory expansion and microprocessor mode with 16-bit external bus • Interrupt priority levels: 7
Watchdog Timer		15-bit × 1 channel (with prescaler)
DMA	DMAC	• 4 channels, cycle steal method • Trigger sources: 31 • Transfer modes: 2 (single transfer and repeat transfer)
	DMACII	• Can be activated by all peripheral function interrupt sources • Transfer modes: 2 (single transfer and burst transfer) • Immediate transfer, calculation transfer, and chain transfer functions
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one-shot timer mode, pulse width modulation (PWM) mode Event counter 2-phase pulse signal processing (2-phase encoder input) × 3
	Timer B	16-bit timer × 6 Timer mode, event counter mode, pulse period measurement mode, pulse width measurement mode
	Timer function for 3-phase motor control	3-phase inverter control × 1 (using timer A1, timer A2, timer A4, and timer B2) On-chip dead time timer

NOTE:

1. Please contact a Renesas sales office for optional features.

Table 1.2 Specifications (144-Pin Package) (2/2)

Item	Function	Specification
Serial Interface	UART0 to UART4	Clock synchronous / asynchronous × 5 I^2C bus, special mode 2, GCI mode, SIM mode IEBus (optional)[1][2]
A/D Converter		10-bit resolution x 34 channels (in single-chip mode) 10-bit resolution x 18 channels (in memory expansion mode and microprocessor mode) including sample and hold function
D/A Converter		8-bit resolution × 2 channels
CRC Calculation Circuit		CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) compliant
X/Y Converter		16 bits x 16 bits
I/O Ports	Programmable I/O ports	• Input only: 1 • CMOS I/O: 121 (in single-chip mode) 81 (in memory expansion and microprocessor mode with 8-bit external bus) 73 (in memory expansion and microprocessor mode with 16-bit external bus) with selectable pull-up resistor • N channel open drain ports: 2
Flash Memory		• Erase and program voltage: VCC1 = VCC2 = 3.0 V to 5.5 V • Erase and program endurance: 100 times (all areas) • Program security: ROM code protect and ID code check • Debug functions: On-chip debug and on-board flash reprogram
Operating Frequency / Supply Voltage		32 MHz / VCC1 = 3.0 V to 5.5 V, VCC2 = 3.0 V to VCC1
Current Consumption		26 mA (32 MHz, VCC1 = VCC2 = 5 V) 23 mA (32 MHz, VCC1 = VCC2 = 3.3 V) 110 µA (approx. 1 MHz, VCC1 = VCC2 = 3.3 V, on-chip oscillator low-power consumption mode → wait mode) 8 µA (approx. 32 kHz / VCC1 = VCC2 = 3.3 V, low-power consumption mode → wait mode) 4 µA (VCC1 = VCC2 = 3.3 V, stop mode)
Operating Ambient Temperature (°C)		-20 to 85°C, -40 to 85°C (optional)[2]
Package		144-pin LQFP (PLQP0144KA-A)

NOTES:

1. IEBus is a registered trademark of NEC Electronics Corporation.
2. Please contact a Renesas sales office for optional features.

Table 1.3 Specifications (100-Pin Package) (1/2)

Item	Function	Specification
CPU	Central processing unit	M32C/80 core (multiplier: 16 bits × 16 bits → 32 bits, multiply-addition operation instructions: 16 × 16 + 48 → 48 bits) • Basic instructions: 108 • Minimum instruction execution time: 31.3 ns (f(CPU) = 32 MHz / VCC1 = 3.0 to 5.5 V) • Operating modes: Single-chip mode, memory expansion mode, and microprocessor mode
Memory	ROM / RAM	Flash memory version: 256KB + 8KB/32 KB, 128KB + 8KB/32 KB ROMless version : – / 32KB
Power Supply Voltage Detection		Voltage monitor interrupt (optional)(1)
External Bus Expansion	Bus / memory expansion function	• Address space: 16 Mbytes • External bus interface: 1 to 7 wait states can be inserted, 4 chip select outputs, 3 V and 5 V interfaces • Bus format: Switchable between separate bus and multiplexed bus formats, switchable data bus width (8-bit or 16-bit)
Clock	Clock generation circuits	• 4 circuits: Main clock, sub clock, on-chip oscillator, PLL frequency synthesizer • Oscillation stop detection: Main clock oscillation stop detect function • Frequency divider circuit: Dividing ratio selectable among 1, 2, 3, 4, 6, 8, 10, 12, 14, 16 • Low power consumption features: Wait mode, stop mode
Interrupts		• Interrupt vectors: 70 • External interrupt inputs: 11 ($\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 6, Key input × 4) Single-chip mode Memory expansion and microprocessor mode with 8-bit external bus 8 ($\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 3, Key input × 4) Memory expansion and microprocessor mode with 16-bit external bus • Interrupt priority levels: 7
Watchdog Timer		15-bit × 1 channel (with prescaler)
DMA	DMAC	• 4 channels, cycle steal method • Trigger sources: 31 • Transfer modes: 2 (single transfer and repeat transfer)
	DMACII	• Can be activated by all peripheral function interrupt sources • Transfer modes: 2 (single transfer and burst transfer) • Immediate transfer, calculation transfer, and chain transfer functions
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one-shot timer mode, pulse width modulation (PWM) mode Event counter 2-phase pulse signal processing (2-phase encoder input) × 3
	Timer B	16-bit timer × 6 Timer mode, event counter mode, pulse period measurement mode, pulse width measurement mode
	Timer function for 3-phase motor control	3-phase inverter control × 1 (using timer A1, timer A2, timer A4, and timer B2) On-chip dead time timer

NOTE:

1. Please contact a Renesas sales office for optional features.

Table 1.4 Specifications (100-Pin Package) (2/2)

Item	Function	Specification
Serial Interface	UART0 to UART4	Clock synchronous / asynchronous × 5 I^2C bus, special mode 2, GCI mode, SIM mode IEBus (optional)[1][2]
A/D Converter		10-bit resolution x 26 channels (in single-chip mode) 10-bit resolution x 10 channels (in memory expansion mode and microprocessor mode) including sample and hold function
D/A Converter		8-bit resolution × 2 channels
CRC Calculation Circuit		CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) compliant
X/Y Converter		16 bits x 16 bits
I/O Ports	Programmable I/O ports	• Input only: 1 • CMOS I/O: 85 (in single-chip mode) 45 (in memory expansion and microprocessor mode with 8-bit external bus) 37 (in memory expansion and microprocessor mode with 16-bit external bus) with selectable pull-up resistor • N channel open drain ports: 2
Flash Memory Version		• Erase and program voltage: VCC1 = VCC2 = 3.0 V to 5.5 V • Erase and program endurance: 100 times (all areas) • Program security: ROM code protect and ID code check • Debug functions: On-chip debug and on-board flash reprogram
Operating Frequency / Supply Voltage		32 MHz: VCC1 = 3.0 V to 5.5 V, VCC2 = 3.0 to VCC1
Current Consumption		26 mA (32 MHz, VCC1 = VCC2 = 5 V) 23 mA (32 MHz, VCC1 = VCC2 = 3.3 V) 110 μA (approx. 1 MHz, VCC1 = VCC2 = 3.3 V, on-chip oscillator low-power consumption mode → wait mode) 8 μA (approx. 32 kHz / VCC1 = VCC2 = 3.3 V, low-power consumption mode → wait mode) 4 μA (VCC1 = VCC2 = 3.3 V, stop mode)
Operating Ambient Temperature (°C)		-20 to 85°C, -40 to 85°C (optional)[2]
Package		100-pin LQFP (PLQP0100KB-A)

NOTES:

1. IEBus is a registered trademark of NEC Electronics Corporation.
2. Please contact a Renesas sales office for optional features.

1.2 Product List

Table 1.5 lists product information. Figure 1.1 shows product numbering system.

Table 1.5 Product List **Current as of Oct. 2009**

Part Number	Package Code	ROM Capacity	RAM Capacity	Remarks
M308B8FGGP	PLQP0144KA-A (144P6Q-A)	256 KB + 8KB(1)	32 KB	Flash memory
M308B6FGGP	PLQP0100KB-A (100P6Q-A)			
M308B8FCGP (P)	PLQP0144KA-A (144P6Q-A)	128 KB + 8KB(1)		
M308B6FCGP (P)	PLQP0100KB-A (100P6Q-A)			
M308B8SGP	PLQP0144KA-A (144P6Q-A)	–		ROMless
M308B6SGP	PLQP0100KB-A (100P6Q-A)			

(P): Under planning

NOTE:

1. Additional 8-Kbyte space is available for data flash memory.

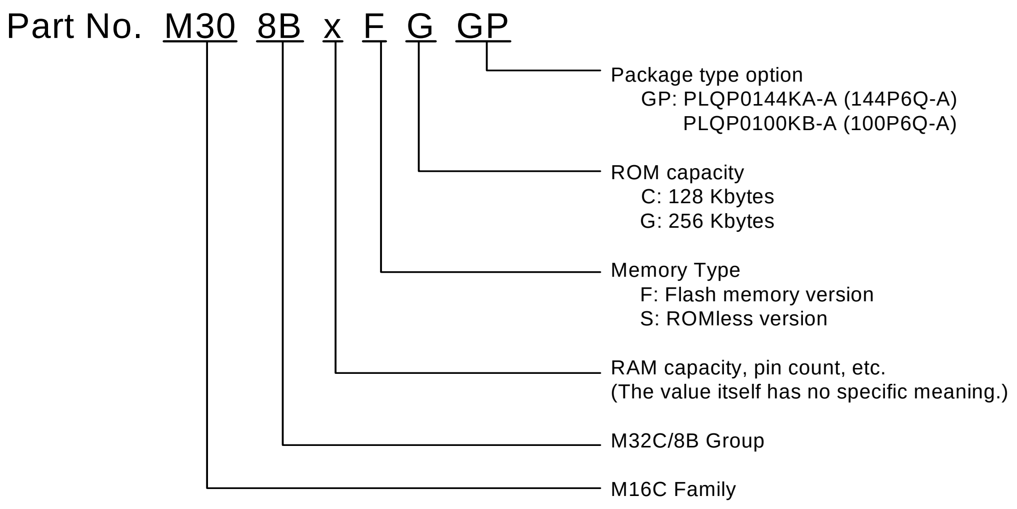
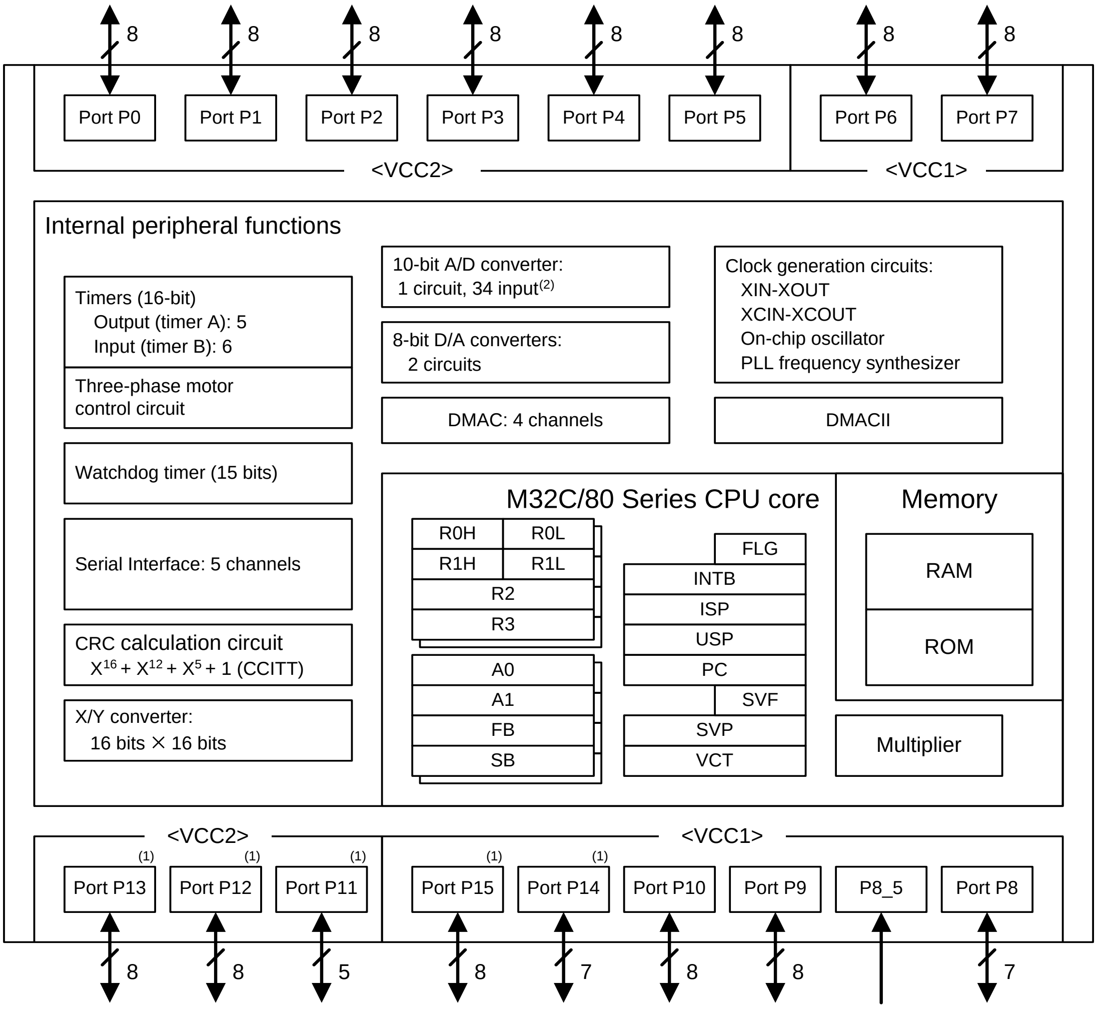


Figure 1.1 Product Numbering System

1.3 Block Diagram

Figure 1.2 shows a block diagram of the M32C/8B Group.

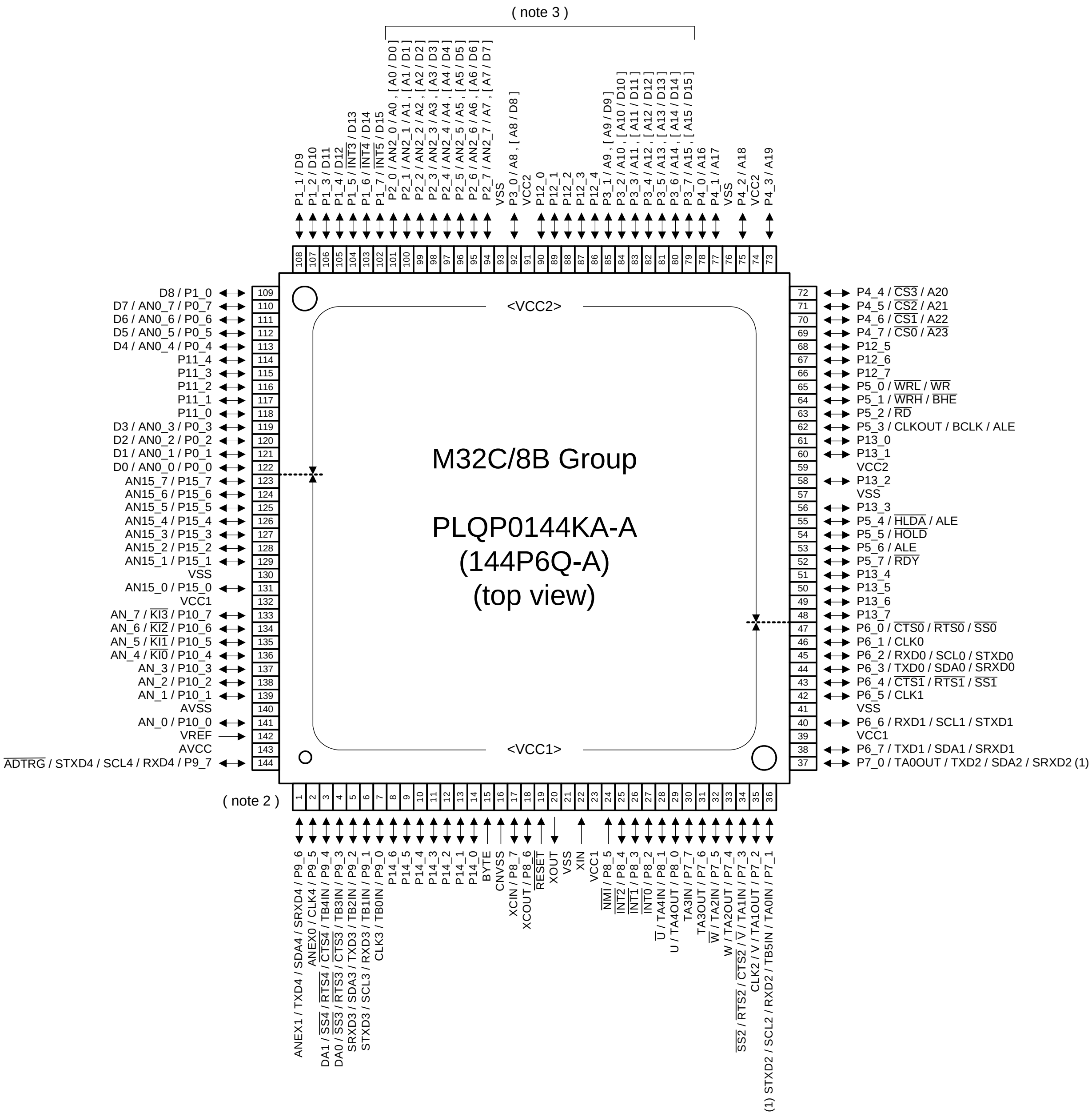


NOTES:
1. Ports P11 to P15 are provided in the 144-pin package only.
2. 34 channels are available in the 144-pin package. 26 channels are available in the 100-pin package.

Figure 1.2 M32C/8B Group Block Diagram

1.4 Pin Assignments

Figures 1.3 and 1.4 show pin assignments (top view).



NOTES:

1. P7_0 and P7_1 are N-channel open drain output ports .
2. Refer to **Package Dimensions** for the pin1 position on the package.
3. Pin names in brackets [ ] represent a single functional signal. They should not be considered as two separate functional signals.

Figure 1.3 Pin Assignment for 144-pin Package

Table 1.6 144-Pin Package List of Pin Names (1/3)

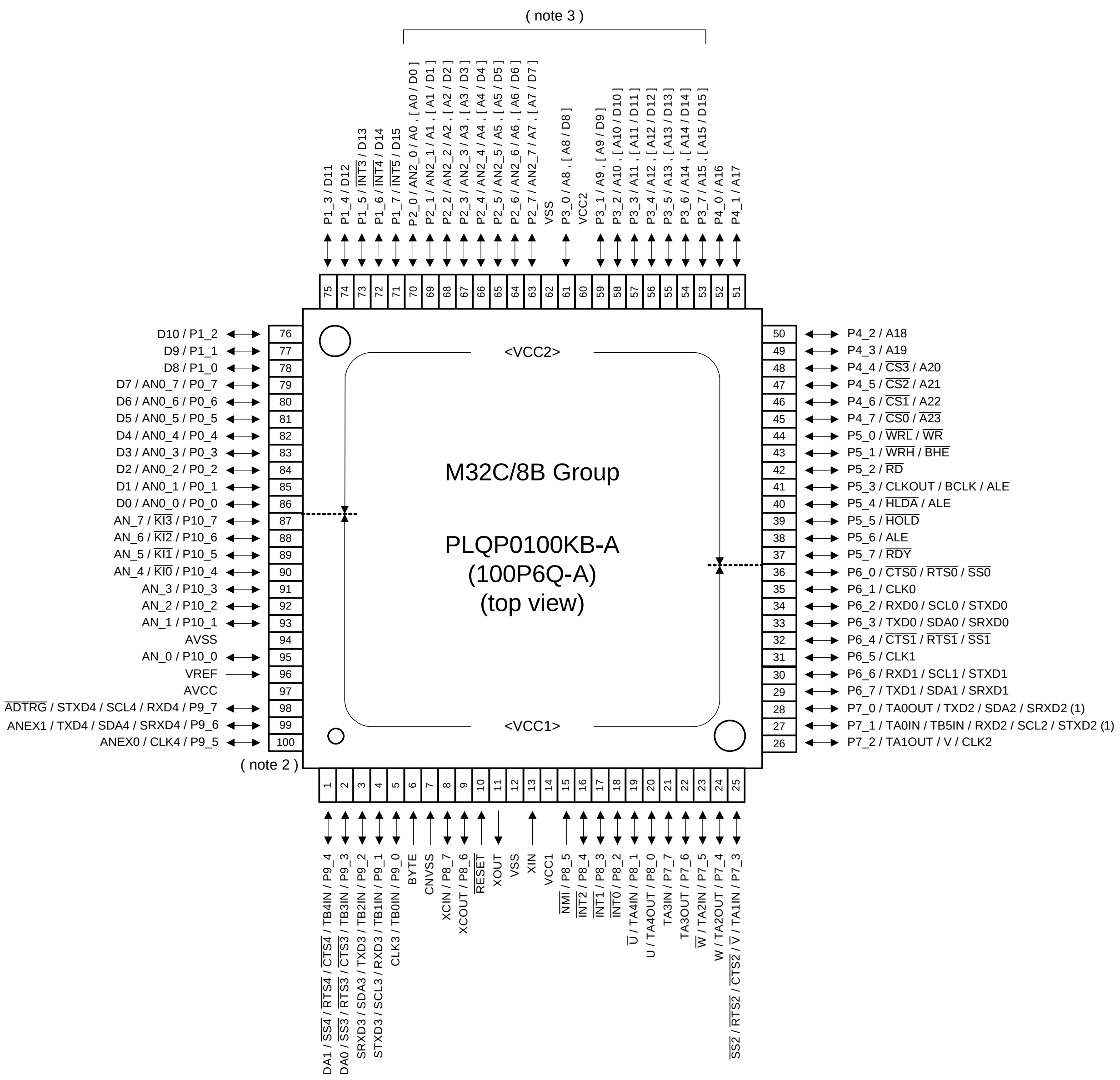
Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	Bus Control Pin
1		P9_6			TXD4/SDA4/SRXD4	ANEX1	
2		P9_5			CLK4	ANEX0	
3		P9_4		TB4IN	$\overline{\mathrm{CTS4}}$/$\overline{\mathrm{RTS4}}$/$\overline{\mathrm{SS4}}$	DA1	
4		P9_3		TB3IN	$\overline{\mathrm{CTS3}}$/$\overline{\mathrm{RTS3}}$/$\overline{\mathrm{SS3}}$	DA0	
5		P9_2		TB2IN	TXD3/SDA3/SRXD3		
6		P9_1		TB1IN	RXD3/SCL3/STXD3		
7		P9_0		TB0IN	CLK3		
8		P14_6					
9		P14_5					
10		P14_4					
11		P14_3					
12		P14_2					
13		P14_1					
14		P14_0					
15	BYTE						
16	CNVSS						
17	XCIN	P8_7					
18	XCOUT	P8_6					
19	$\overline{\mathrm{RESET}}$						
20	XOUT						
21	VSS						
22	XIN						
23	VCC1						
24		P8_5	$\overline{\mathrm{NMI}}$				
25		P8_4	$\overline{\mathrm{INT2}}$				
26		P8_3	$\overline{\mathrm{INT1}}$				
27		P8_2	$\overline{\mathrm{INT0}}$				
28		P8_1		TA4IN/$\overline{\mathrm{U}}$			
29		P8_0		TA4OUT/U			
30		P7_7		TA3IN			
31		P7_6		TA3OUT			
32		P7_5		TA2IN/$\overline{\mathrm{W}}$			
33		P7_4		TA2OUT/W			
34		P7_3		TA1IN/$\overline{\mathrm{V}}$	$\overline{\mathrm{CTS2}}$/$\overline{\mathrm{RTS2}}$/$\overline{\mathrm{SS2}}$		
35		P7_2		TA1OUT/V	CLK2		
36		P7_1		TA0IN/TB5IN	RXD2/SCL2/STXD2		
37		P7_0		TA0OUT	TXD2/SDA2/SRXD2		
38		P6_7			TXD1/SDA1/SRXD1		
39	VCC1						
40		P6_6			RXD1/SCL1/STXD1		
41	VSS						
42		P6_5			CLK1		
43		P6_4			$\overline{\mathrm{CTS1}}$/$\overline{\mathrm{RTS1}}$/$\overline{\mathrm{SS1}}$		
44		P6_3			TXD0/SDA0/SRXD0		
45		P6_2			RXD0/SCL0/STXD0		
46		P6_1			CLK0		
47		P6_0			$\overline{\mathrm{CTS0}}$/$\overline{\mathrm{RTS0}}$/$\overline{\mathrm{SS0}}$		
48		P13_7					
49		P13_6					
50		P13_5					

Table 1.7 144-Pin Package List of Pin Names (2/3)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	Bus Control Pin
51		P13_4					
52		P5_7					$\overline{\text{RDY}}$
53		P5_6					ALE
54		P5_5					$\overline{\text{HOLD}}$
55		P5_4					$\overline{\text{HLDA}}$/ALE
56		P13_3					
57	VSS						
58		P13_2					
59	VCC2						
60		P13_1					
61		P13_0					
62	CLKOUT	P5_3					BCLK/ALE
63		P5_2					$\overline{\text{RD}}$
64		P5_1					$\overline{\text{WRH}}$/$\overline{\text{BHE}}$
65		P5_0					$\overline{\text{WRL}}$/$\overline{\text{WR}}$
66		P12_7					
67		P12_6					
68		P12_5					
69		P4_7					$\overline{\text{CS0}}$/$\overline{\text{A23}}$
70		P4_6					$\overline{\text{CS1}}$/A22
71		P4_5					$\overline{\text{CS2}}$/A21
72		P4_4					$\overline{\text{CS3}}$/A20
73		P4_3					A19
74	VCC2						
75		P4_2					A18
76	VSS						
77		P4_1					A17
78		P4_0					A16
79		P3_7					A15,[A15/D15]
80		P3_6					A14,[A14/D14]
81		P3_5					A13,[A13/D13]
82		P3_4					A12,[A12/D12]
83		P3_3					A11,[A11/D11]
84		P3_2					A10,[A10/D10]
85		P3_1					A9,[A9/D9]
86		P12_4					
87		P12_3					
88		P12_2					
89		P12_1					
90		P12_0					
91	VCC2						
92		P3_0					A8,[A8/D8]
93	VSS						
94		P2_7				AN2_7	A7,[A7/D7]
95		P2_6				AN2_6	A6,[A6/D6]
96		P2_5				AN2_5	A5,[A5/D5]
97		P2_4				AN2_4	A4,[A4/D4]
98		P2_3				AN2_3	A3,[A3/D3]
99		P2_2				AN2_2	A2,[A2/D2]
100		P2_1				AN2_1	A1,[A1/D1]

Table 1.8 144-Pin Package List of Pin Names (3/3)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	Bus Control Pin
101		P2_0				AN2_0	A0,[A0/D0]
102		P1_7	$\overline{\text{INT5}}$				D15
103		P1_6	$\overline{\text{INT4}}$				D14
104		P1_5	$\overline{\text{INT3}}$				D13
105		P1_4					D12
106		P1_3					D11
107		P1_2					D10
108		P1_1					D9
109		P1_0					D8
110		P0_7				AN0_7	D7
111		P0_6				AN0_6	D6
112		P0_5				AN0_5	D5
113		P0_4				AN0_4	D4
114		P11_4					
115		P11_3					
116		P11_2					
117		P11_1					
118		P11_0					
119		P0_3				AN0_3	D3
120		P0_2				AN0_2	D2
121		P0_1				AN0_1	D1
122		P0_0				AN0_0	D0
123		P15_7				AN15_7	
124		P15_6				AN15_6	
125		P15_5				AN15_5	
126		P15_4				AN15_4	
127		P15_3				AN15_3	
128		P15_2				AN15_2	
129		P15_1				AN15_1	
130	VSS						
131		P15_0				AN15_0	
132	VCC1						
133		P10_7	$\overline{\text{KI3}}$			AN_7	
134		P10_6	$\overline{\text{KI2}}$			AN_6	
135		P10_5	$\overline{\text{KI1}}$			AN_5	
136		P10_4	$\overline{\text{KI0}}$			AN_4	
137		P10_3				AN_3	
138		P10_2				AN_2	
139		P10_1				AN_1	
140	AVSS						
141		P10_0				AN_0	
142	VREF						
143	AVCC						
144		P9_7			RXD4/SCL4/STXD4	$\overline{\text{ADTRG}}$	



NOTES:

1. P7_0 and P7_1 are N-channel open drain output ports.
2. Refer to **Package Dimensions** for the pin1 position on the package.
3. Pin names in brackets [ ] represent a single functional signal. They should not be considered as two separate functional signals.

Figure 1.4 Pin Assignment for 100-pin Package

Table 1.9 100-Pin Package List of Pin Names (1/2)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	Bus Control Pin
1		P9_4		TB4IN	$\overline{\text{CTS4}}$/$\overline{\text{RTS4}}$/$\overline{\text{SS4}}$	DA1	
2		P9_3		TB3IN	$\overline{\text{CTS3}}$/$\overline{\text{RTS3}}$/$\overline{\text{SS3}}$	DA0	
3		P9_2		TB2IN	TXD3/SDA3/SRXD3		
4		P9_1		TB1IN	RXD3/SCL3/STXD3		
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	$\overline{\text{INT2}}$				
17		P8_3	$\overline{\text{INT1}}$				
18		P8_2	$\overline{\text{INT0}}$				
19		P8_1		TA4IN/$\overline{\text{U}}$			
20		P8_0		TA4OUT/U			
21		P7_7		TA3IN			
22		P7_6		TA3OUT			
23		P7_5		TA2IN/$\overline{\text{W}}$			
24		P7_4		TA2OUT/W			
25		P7_3		TA1IN/$\overline{\text{V}}$	$\overline{\text{CTS2}}$/$\overline{\text{RTS2}}$/$\overline{\text{SS2}}$		
26		P7_2		TA1OUT/V	CLK2		
27		P7_1		TA0IN/TB5IN	RXD2/SCL2/STXD2		
28		P7_0		TA0OUT	TXD2/SDA2/SRXD2		
29		P6_7			TXD1/SDA1/SRXD1		
30		P6_6			RXD1/SCL1/STXD1		
31		P6_5			CLK1		
32		P6_4			$\overline{\text{CTS1}}$/$\overline{\text{RTS1}}$/$\overline{\text{SS1}}$		
33		P6_3			TXD0/SDA0/SRXD0		
34		P6_2			RXD0/SCL0/STXD0		
35		P6_1			CLK0		
36		P6_0			$\overline{\text{CTS0}}$/$\overline{\text{RTS0}}$/$\overline{\text{SS0}}$		
37		P5_7					$\overline{\text{RDY}}$
38		P5_6					ALE
39		P5_5					$\overline{\text{HOLD}}$
40		P5_4					$\overline{\text{HLDA}}$/ALE
41	CLKOUT	P5_3					BCLK/ALE
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					$\overline{\text{CS0}}$/A23
46		P4_6					$\overline{\text{CS1}}$/A22
47		P4_5					$\overline{\text{CS2}}$/A21
48		P4_4					$\overline{\text{CS3}}$/A20
49		P4_3					A19
50		P4_2					A18

Table 1.10 100-Pin Package List of Pin Names (2/2)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART Pin	Analog Pin	Bus Control Pin
51		P4_1					A17
52		P4_0					A16
53		P3_7					A15,[A15/D15]
54		P3_6					A14,[A14/D14]
55		P3_5					A13,[A13/D13]
56		P3_4					A12,[A12/D12]
57		P3_3					A11,[A11/D11]
58		P3_2					A10,[A10/D10]
59		P3_1					A9,[A9/D9]
60	VCC2						
61		P3_0					A8,[A8/D8]
62	VSS						
63		P2_7				AN2_7	A7,[A7/D7]
64		P2_6				AN2_6	A6,[A6/D6]
65		P2_5				AN2_5	A5,[A5/D5]
66		P2_4				AN2_4	A4,[A4/D4]
67		P2_3				AN2_3	A3,[A3/D3]
68		P2_2				AN2_2	A2,[A2/D2]
69		P2_1				AN2_1	A1,[A1/D1]
70		P2_0				AN2_0	A0,[A0/D0]
71		P1_7	$\overline{\text{INT5}}$				D15
72		P1_6	$\overline{\text{INT4}}$				D14
73		P1_5	$\overline{\text{INT3}}$				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	$\overline{\text{KI3}}$			AN_7	
88		P10_6	$\overline{\text{KI2}}$			AN_6	
89		P10_5	$\overline{\text{KI1}}$			AN_5	
90		P10_4	$\overline{\text{KI0}}$			AN_4	
91		P10_3				AN_3	
92		P10_2				AN_2	
93		P10_1				AN_1	
94	AVSS						
95		P10_0				AN_0	
96	VREF						
97	AVCC						
98		P9_7			RXD4/SCL4/STXD4	$\overline{\text{ADTRG}}$	
99		P9_6			TXD4/SDA4/SRXD4	ANEX1	
100		P9_5			CLK4	ANEX0	

1.5 Pin Functions

Table 1.11 Pin Functions (100-Pin and 144-Pin Packages) (1/3)

Item	Pin Name	I/O Type	Supply Voltage	Description
Power supply	VCC1,VCC2 VSS	–	–	Apply 3.0 to 5.5 V to pins VCC1 and VCC2, and 0 V to the VSS pin. Meet the input condition of VCC1 ≥ VCC2.
Analog power supply input	AVCC AVSS	–	VCC1	Power supply input pins to the A/D converter and D/A converter. Connect the AVCC pin to VCC1, and the AVSS pin to VSS.
Reset input	$\overline{\text{RESET}}$	I	VCC1	The MCU is placed in the reset state while applying an “L” signal to the $\overline{\text{RESET}}$ pin.
CNVSS	CNVSS	I	VCC1	This pin switches processor mode. Apply an “L” to the CNVSS pin to start up in single-chip mode, or an “H” to start up in microprocessor mode and boot mode.
External data bus width select input	BYTE	I	VCC1	This pin switches data bus width in external memory space 3. A data bus is 16 bits wide when the BYTE pin is held “L” and 8 bits wide when it is held “H”. Fix to either “L” or “H”. Apply an “L” to the BYTE pin in single-chip mode.
Bus control Pins	D0 to D7	I/O	VCC2	Data (D0 to D7) input/output pins while accessing an external memory space with separate bus.
	D8 to D15	I/O	VCC2	Data (D8 to D15) input/output pins while accessing an external memory space with 16-bit separate bus.
	A0 to A22	O	VCC2	Address bits (A0 to A22) output pins.
	$\overline{\text{A23}}$	O	VCC2	Inverted address bit (A23) output pin.
	A0/D0 to A7/D7	I/O	VCC2	Data (D0 to D7) input/output and 8 low-order address bits (A0 to A7) output are performed by time-sharing these pins while accessing an external memory space with multiplexed bus.
	A8/D8 to A15/D15	I/O	VCC2	Data (D8 to D15) input/output and 8 middle-order address bits (A8 to A15) output are performed by time-sharing these pins while accessing an external memory space with 16-bit multiplexed bus.
	$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	O	VCC2	Chip-select signal output pins used to specify external devices.
	$\overline{\text{WRL}}$/$\overline{\text{WR}}$ $\overline{\text{WRH}}$/$\overline{\text{BHE}}$ $\overline{\text{RD}}$	O	VCC2	$\overline{\text{WRL}}$, $\overline{\text{WRH}}$, ($\overline{\text{WR}}$, $\overline{\text{BHE}}$) and $\overline{\text{RD}}$ signal output pins. $\overline{\text{WRL}}$ and $\overline{\text{WRH}}$ can be switched with $\overline{\text{WR}}$ and $\overline{\text{BHE}}$ by a program. • $\overline{\text{WRL}}$, $\overline{\text{WRH}}$ and $\overline{\text{RD}}$ are selected: If external data bus is 16 bits wide, data is written to an even address in external memory space while an “L” is output from the $\overline{\text{WRL}}$ pin. Data is written to an odd address while an “L” is output from the $\overline{\text{WRH}}$ pin. Data is read while an “L” is output from the $\overline{\text{RD}}$ pin. • $\overline{\text{WR}}$, $\overline{\text{BHE}}$ and $\overline{\text{RD}}$ are selected: Data is written while an “L” is output from the $\overline{\text{WR}}$ pin. Data is read while an “L” is output from the $\overline{\text{RD}}$ pin. Data in odd address is accessed while an “L” is output from the $\overline{\text{BHE}}$ pin. Select $\overline{\text{WR}}$, $\overline{\text{BHE}}$ and $\overline{\text{RD}}$ when an external data bus is 8 bits wide.
	ALE	O	VCC2	ALE signal is used for the external devices to latch address signals when the multiplexed bus is selected.
	$\overline{\text{HOLD}}$	I	VCC2	The MCU is placed in the hold state while an “L” signal is applied to the $\overline{\text{HOLD}}$ pin.
	$\overline{\text{HLDA}}$	O	VCC2	The $\overline{\text{HLDA}}$ pin outputs an “L” while the MCU is placed in the hold state.
	$\overline{\text{RDY}}$	I	VCC2	Bus is placed in the wait state while an “L” signal is applied to the $\overline{\text{RDY}}$ pin.

Table 1.12 Pin Functions (100-Pin and 144-Pin Packages) (2/3)

Item	Pin Name	I/O Type	Supply Voltage	Description
Main clock input	XIN	I	VCC1	Input/output pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT. To apply an external clock, apply it to XIN and leave XOUT open.
Main clock output	XOUT	O	VCC1	
Sub clock input	XCIN	I	VCC1	Input/output pins for the sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUT. To apply an external clock, apply it to XCIN and leave XCOUT open.
Sub clock output	XCOUT	O	VCC1	
BCLK output	BCLK	O	VCC2	Bus clock output pin
Clock output	CLKOUT	O	VCC2	The CLKOUT pin outputs the clock having the same frequency as fC, f8, or f32
$\overline{\text{INT}}$ interrupt input	$\overline{\text{INT0}}$ to $\overline{\text{INT2}}$	I	VCC1	$\overline{\text{INT}}$ interrupt input pins
	$\overline{\text{NT3}}$ to $\overline{\text{INT5}}$	I	VCC2	
$\overline{\text{NMI}}$ interrupt input	$\overline{\text{NMI}}$	I	VCC1	$\overline{\text{NMI}}$ interrupt input pin. Connect the $\overline{\text{NMI}}$ pin to VCC1 via a resistor when the $\overline{\text{NMI}}$ interrupt is not used.
Timer A	TA0OUT to TA4OUT	I/O	VCC1	Timer A0 to A4 input/output pins (TA0OUT is N-channel open drain output)
	TA0IN to TA4IN	I	VCC1	Timer A0 to A4 input pins
Timer B	TB0IN to TB5IN	I	VCC1	Timer B0 to B5 input pins
Three-phase motor control timer output	U, $\overline{\text{U}}$, V, $\overline{\text{V}}$, W, $\overline{\text{W}}$	O	VCC1	Three-phase motor control timer output pins
Serial interface	$\overline{\text{CTS0}}$ to $\overline{\text{CTS4}}$	I	VCC1	Input pins to control data transmission
	$\overline{\text{RTS0}}$ to $\overline{\text{RTS4}}$	O	VCC1	Output pins to control data reception
	CLK0 to CLK4	I/O	VCC1	Serial clock input/output pins
	RXD0 to RXD4	I	VCC1	Serial data input pins
	TXD0 to TXD4	O	VCC1	Serial data output pins (TXD2 is N-channel open drain output)
I^2C mode	SDA0 to SDA4	I/O	VCC1	Serial data input/output pins (SDA2 is N-channel open drain output)
	SCL0 to SCL4	I/O	VCC1	Serial clock input/output pins (SCL2 is N-channel open drain output)
Serial interface special function	STXD0 to STXD4	O	VCC1	Serial data output pins when slave mode is selected (STXD2 is N-channel open drain output)
	SRXD0 to SRXD4	I	VCC1	Serial data input pins when slave mode is selected
	$\overline{\text{SS0}}$ to $\overline{\text{SS4}}$	I	VCC1	Control input pins used in the serial interface special mode.

Table 1.13 Pin Functions (100-Pin and 144-Pin Packages) (3/3)

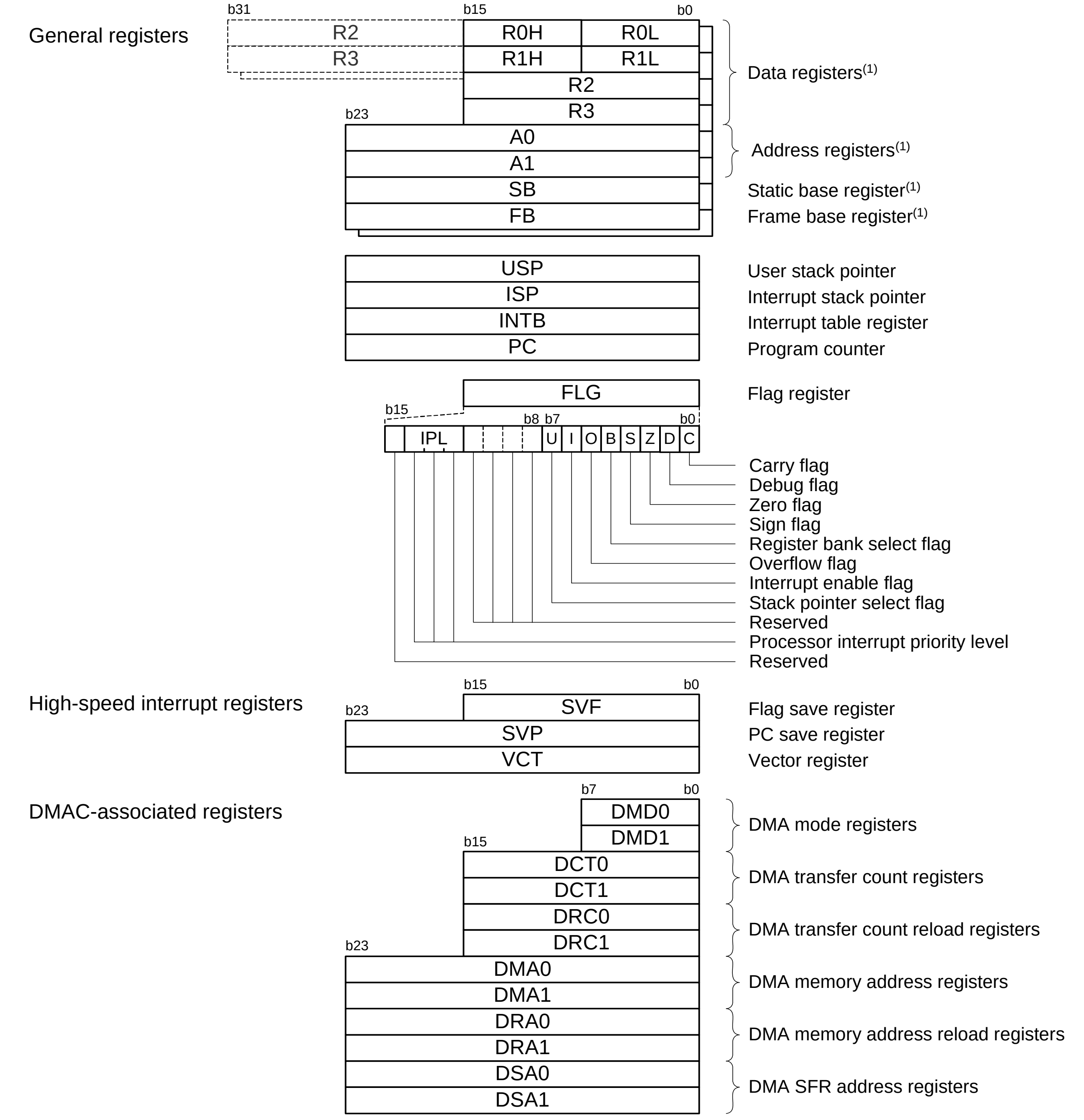
Item	Pin Name	I/O Type	Supply Voltage	Description
Reference voltage input	VREF	I	–	The VREF pin supplies the reference voltage to the A/D converter and D/A converter.
A/D converter	AN_0 to AN_7	I	VCC1	Analog input pins for the A/D converter.
	AN0_0 to AN0_7, AN2_0 to AN2_7	I	VCC2	Analog input pins for the A/D converter.
	$\overline{\text{ADTRG}}$	I	VCC1	External trigger input pin for the A/D converter.
	ANEX0	I/O	VCC1	Extended analog input pin for the A/D converter or output pin in external op-amp connection mode.
	ANEX1	I	VCC1	Extended analog input pin for the A/D converter.
D/A converter	DA0, DA1	O	VCC1	Output pins for the D/A converter.
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	I/O	VCC2	8-bit CMOS I/O ports. The Port Pi Direction Register (i = 0 to 15) determines if each pin is used as an input port or an output port. The Pull-up Control Registers determine if the input ports, divided into groups of four, are pulled up or not.
	P6_0 to P6_7, P7_0 to P7_7, P9_0 to P9_7, P10_0 to P10_7	I/O	VCC1	These 8-bit I/O ports are functionally equivalent to P0. (P7_0 and P7_1 are N-channel open drain output.)
	P8_0 to P8_4 P8_6, P8_7			These I/O ports are functionally equivalent to P0.
Input port	P8_5	I	VCC1	Shares the pin with $\overline{\text{NMI}}$. Input port to read $\overline{\text{NMI}}$ pin level.
Key input interrupt input	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	VCC1	Key input interrupt input pins

Table 1.14 Pin Functions (144-Pin Package Only)

Item	Pin Name	I/O Type	Supply Voltage	Description
A/D converter	AN15_0 to AN15_7	I	VCC1	Analog input pins for the A/D converter
I/O port	P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7	I/O	VCC2	These I/O ports are functionally equivalent to P0.
	P14_0 to P14_6, P15_0 to P15_7	I/O	VCC1	

2. Central Processing Unit (CPU)

Figure 2.1 shows the CPU registers.
The register bank is comprised of eight registers (R0, R1, R2, R3, A0, A1, SB, and FB) out of 28 CPU registers. There are two sets of register banks.



NOTE:
1. These registers comprise a register bank.
There are two sets of register banks (register bank 0 and register bank 1).

Figure 2.1 CPU Register

2.1 General Registers

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

R0, R1, R2, and R3 are 16-bit registers for transfer, arithmetic and logic operations. R0 and R1 can be split into high-order (R0H/R1H) and low-order bits (R0L/R1L) to be used separately as 8-bit data registers.
R0 can be combined with R2 and used as a 32-bit data register (R2R0). The same applies to R3R1.

2.1.2 Address Registers (A0 and A1)

A0 and A1 are 24-bit registers used for A0-/A1-indirect addressing, A0-/A1-relative addressing, transfer, arithmetic and logic operations.

2.1.3 Static Base Register (SB)

SB is a 24-bit register used for SB-relative addressing.

2.1.4 Frame Base Register (FB)

FB is a 24-bit register used for FB-relative addressing.

2.1.5 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)

The stack pointers (SP), USP and ISP, are 24 bits wide each. The U flag is used to switch between USP and ISP. Refer to **2.1.8 Flag Register (FLG)** for details on the U flag. Set USP and ISP to even addresses to execute an interrupt sequence efficiently.

2.1.6 Interrupt Table Register (INTB)

INTB is a 24-bit register indicating the starting address of a relocatable interrupt vector table.

2.1.7 Program Counter (PC)

PC is 24 bits wide and indicates the address of the next instruction to be executed.

2.1.8 Flag Register (FLG)

FLG is a 16-bit register indicating the CPU state.

2.1.8.1 Carry Flag (C)

The C flag indicates whether or not carry or borrow has been generated after executing an instruction.

2.1.8.2 Debug Flag (D)

The D flag is for debugging only. Set it to 0.

2.1.8.3 Zero Flag (Z)

The Z flag becomes 1 when an arithmetic operation results in 0; otherwise becomes 0.

2.1.8.4 Sign Flag (S)

The S flag becomes 1 when an arithmetic operation results in a negative value; otherwise becomes 0.

2.1.8.5 Register Bank Select Flag (B)

Register bank 0 is selected when the B flag is set to 0. Register bank 1 is selected when this flag is set to 1.

2.1.8.6 Overflow Flag (O)

The O flag becomes 1 when an arithmetic operation results in an overflow; otherwise becomes 0.

2.1.8.7 Interrupt Enable Flag (I)

The I flag enables maskable interrupts.
Interrupts are disabled when the I flag is set to 0 and enabled when it is set to 1. The I flag becomes 0 when an interrupt request is acknowledged.

2.1.8.8 Stack Pointer Select Flag (U)

ISP is selected when the U flag is set to 0. USP is selected when the U flag is set to 1.
The U flag becomes 0 when a hardware interrupt request is acknowledged or the INT instruction specifying software interrupt numbers 0 to 31 is executed.

2.1.8.9 Processor Interrupt Priority Level (IPL)

IPL is 3 bits wide and assigns processor interrupt priority levels from level 0 to level 7.
If a requested interrupt has higher priority level than IPL, the interrupt is enabled.

2.1.8.10 Reserved Space

Only write 0 to bits assigned to the reserved space. When read, the bits return undefined values.

2.2 High-Speed Interrupt Registers

Registers associated with the high-speed interrupt are as follows:

- Flag save register (SVF)
- PC save register (SVP)
- Vector register (VCT)

Refer to **11.4 High-Speed Interrupt** for details.

2.3 DMAC-Associated Registers

Registers associated with the DMAC are as follows:

- DMA mode register (DMD0, DMD1)
- DMA transfer count register (DCT0, DCT1)
- DMA transfer count reload register (DRC0, DRC1)
- DMA memory address register (DMA0, DMA1)
- DMA memory address reload register (DRA0, DRA1)
- DMA SFR address register (DSA0, DSA1)

Refer to **13. DMAC** for details.

3. Memory

Figure 3.1 shows a memory map of the M32C/8B Group.

The M32C/8B Group has 16-Mbyte address space from addresses 000000h to FFFFFFh.

The internal ROM is allocated in lower addresses, beginning with address FFFFFFh. For example, a 256-Kbyte internal ROM area is allocated in addresses FC0000h to FFFFFFh.

The fixed interrupt vectors are allocated in addresses FFFFDCh to FFFFFFh. They store the starting address of each interrupt routine. Refer to **11. Interrupts** for details.

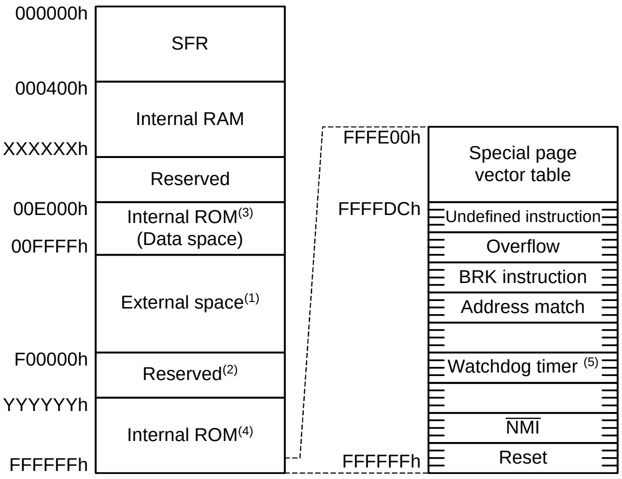
The internal RAM is allocated higher addresses, beginning with address 000400h. For example, a 32-Kbyte internal RAM area is allocated in addresses 000400h to 0083FFh. The internal RAM is used not only for storing data but for the stacks when subroutines are called or when interrupt requests are acknowledged.

SFRs are allocated in addresses 000000h to 0003FFh. The peripheral function control registers such as for I/O ports, A/D converters, serial interfaces, timers are allocated here. All blank spaces within SFRs are reserved and cannot be accessed by users.

The special page vectors are allocated addresses FFFE00h to FFFFDBh. They are used for the JMPS instruction and JSRS instruction. Refer to the Renesas publication **M32C/80 Series Software Manual** for details.

Internal RAM	
Capacity	XXXXXXh
32 Kbytes	0083FFh

Internal ROM	
Capacity	YYYYYYh
128 Kbytes	FE0000h
256 Kbytes	FC0000h



NOTES:
1. The space is used as the external space in memory expansion mode and microprocessor mode. It is reserved in single-chip mode.
2. The space is reserved in single-chip mode and memory expansion mode. It is used as the external space in microprocessor mode.
3. Additional two blocks of 4-Kbyte space are provided in the flash memory version to store data. This space is used in single-chip mode and memory expansion mode. It is reserved in microprocessor mode.
4. This space is used as the internal ROM in single-chip mode and memory expansion mode. It is used as the external space in microprocessor mode.
5. The watchdog timer interrupt, oscillation stop detection interrupt, and voltage monitor interrupt share the same vector.

Figure 3.1 Memory Map

4. Special Function Registers (SFRs)

Special Function Registers (SFRs) are the control registers of peripheral functions. Tables 4.1 to 4.11 list SFR address maps.

Table 4.1 SFR Address Map (1/11)

Address	Register	Symbol	After Reset
0000h			
0001h			
0002h			
0003h			
0004h	Processor Mode Register 0[1]	PM0	1000 0000b(CNVSS="L") 0000 0011b(CNVSS="H")
0005h	Processor Mode Register 1	PM1	00h
0006h	System Clock Control Register 0	CM0	0000 1000b
0007h	System Clock Control Register 1	CM1	0010 0000b
0008h			
0009h	Address Match Interrupt Enable Register	AIER	00h
000Ah	Protect Register	PRCR	XXXX 0000b
000Bh	External Data Bus Width Control Register	DS	XXXX 1000b(BYTE="L") XXXX 0000b(BYTE="H")
000Ch	Main Clock Division Register	MCD	XXX0 1000b
000Dh	Oscillation Stop Detection Register	CM2	00h
000Eh	Watchdog Timer Start Register	WDTS	XXh
000Fh	Watchdog Timer Control Register	WDC	000X XXXXb
0010h	Address Match Interrupt Register 0	RMAD0	000000h
0011h			
0012h			
0013h	Processor Mode Register 2	PM2	00h
0014h	Address Match Interrupt Register 1	RMAD1	000000h
0015h			
0016h			
0017h	Reference Voltage Configuration Register	DVCR	1000 1111b
0018h	Address Match Interrupt Register 2	RMAD2	000000h
0019h			
001Ah			
001Bh	Voltage Monitor Register	LVDC	0000 1000h
001Ch	Address Match Interrupt Register 3	RMAD3	000000h
001Dh			
001Eh			
001Fh	Voltage Regulator Control Register	VRCR	00h
0020h			
0021h			
0022h			
0023h			
0024h			
0025h			
0026h	PLL Control Register 0	PLC0	0001 X010b
0027h			
0028h	Address Match Interrupt Register 4	RMAD4	000000h
0029h			
002Ah			
002Bh			
002Ch	Address Match Interrupt Register 5	RMAD5	000000h
002Dh			
002Eh			
002Fh			

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTE:

1. Bits PM01 and PM00 in the PM0 register maintain values set before reset, even after software reset or watchdog timer reset has been performed.

Table 4.2 SFR Address Map (2/11)

Address	Register	Symbol	After Reset
0030h			
0031h			
0032h			
0033h			
0034h			
0035h			
0036h			
0037h			
0038h	Address Match Interrupt Register 6	RMAD6	000000h
0039h			
003Ah			
003Bh			
003Ch	Address Match Interrupt Register 7	RMAD7	000000h
003Dh			
003Eh			
003Fh			
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h			
0048h	External Space Wait Control Register 0	EWCR0	X0X0 0011b
0049h	External Space Wait Control Register 1	EWCR1	X0X0 0011b
004Ah	External Space Wait Control Register 2	EWCR2	X0X0 0011b
004Bh	External Space Wait Control Register 3	EWCR3	X0X0 0011b
004Ch	Page Mode Wait Control Register 0(1)	PWCR0	0001 0001b
004Dh	Page Mode Wait Control Register 1(1)	PWCR1	0001 0001b
004Eh			
004Fh			
0050h	Flash Memory Control Register 3(2)	FMR 3	XX0X XX00b
0051h			
0052h	Flash Memory Control Register 2(2)	FMR 2	XXXX XXX0b
0053h			
0054h			
0055h	Flash Memory Control Register 1(2)	FMR1	0000 XX0Xb
0056h			
0057h	Flash Memory Control Register 0(2)	FMR0	0000 0001b
0058h			
0059h	Flash Memory Control Register 4(2)	FMR4	00h
005Ah			
005Bh			
005Ch			
005Dh			
005Eh			
005Fh			

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:

1. These registers are available only in ROMless version.
2. These registers are not available in ROMless version.

Table 4.3 SFR Address Map (3/11)

Address	Register	Symbol	After Reset
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h	DMA0 Interrupt Control Register	DM0IC	XXXX X000b
0069h	Timer B5 Interrupt Control Register	TB5IC	XXXX X000b
006Ah	DMA2 Interrupt Control Register	DM2IC	XXXX X000b
006Bh	UART2 Receive/ACK Interrupt Control Register	S2RIC	XXXX X000b
006Ch	Timer A0 Interrupt Control Register	TA0IC	XXXX X000b
006Dh	UART3 Receive/ACK Interrupt Control Register	S3RIC	XXXX X000b
006Eh	Timer A2 Interrupt Control Register	TA2IC	XXXX X000b
006Fh	UART4 Receive/ACK Interrupt Control Register	S4RIC	XXXX X000b
0070h	Timer A4 Interrupt Control Register	TA4IC	XXXX X000b
0071h	UART0/UART3 Bus Conflict Detection Interrupt Control Register	BCN0IC/BCN3IC	XXXX X000b
0072h	UART0 Receive/ACK Interrupt Control Register	S0RIC	XXXX X000b
0073h	A/D0 Conversion Interrupt Control Register	AD0IC	XXXX X000b
0074h	UART1 Receive/ACK Interrupt Control Register	S1RIC	XXXX X000b
0075h			
0076h	Timer B1 Interrupt Control Register	TB1IC	XXXX X000b
0077h			
0078h	Timer B3 Interrupt Control Register	TB3IC	XXXX X000b
0079h			
007Ah	$\overline{\mathrm{INT5}}$ Interrupt Control Register	INT5IC	XX00 X000b
007Bh			
007Ch	$\overline{\mathrm{INT3}}$ Interrupt Control Register	INT3IC	XX00 X000b
007Dh			
007Eh	$\overline{\mathrm{INT1}}$ Interrupt Control Register	INT1IC	XX00 X000b
007Fh			
0080h			
0081h			
0082h			
0083h			
0084h			
0085h			
0086h			
0087h			
0088h	DMA1 Interrupt Control Register	DM1IC	XXXX X000b
0089h	UART2 Transmit/NACK Interrupt Control Register	S2TIC	XXXX X000b
008Ah	DMA3 Interrupt Control Register	DM3IC	XXXX X000b
008Bh	UART3 Transmit/NACK Interrupt Control Register	S3TIC	XXXX X000b
008Ch	Timer A1 Interrupt Control Register	TA1IC	XXXX X000b
008Dh	UART4 Transmit/NACK Interrupt Control Register	S4TIC	XXXX X000b
008Eh	Timer A3 Interrupt Control Register	TA3IC	XXXX X000b
008Fh	UART2 Bus Conflict Detection Interrupt Control Register	BCN2IC	XXXX X000b

X: Undefined
Blank spaces are all reserved. No access is allowed.

Table 4.4 SFR Address Map (4/11)

Address	Register	Symbol	After Reset
0090h	UART0 Transmit/NACK Interrupt Control Register	S0TIC	XXXX X000b
0091h	UART1/UART4 Bus Conflict Detection Interrupt Control Register	BCN1IC/BCN4IC	XXXX X000b
0092h	UART1 Transmit/NACK Interrupt Control Register	S1TIC	XXXX X000b
0093h	Key Input Interrupt Control Register	KUPIC	XXXX X000b
0094h	Timer B0 Interrupt Control Register	TB0IC	XXXX X000b
0095h			
0096h	Timer B2 Interrupt Control Register	TB2IC	XXXX X000b
0097h			
0098h	Timer B4 Interrupt Control Register	TB4IC	XXXX X000b
0099h			
009Ah	$\overline{\text{INT4}}$ Interrupt Control Register	INT4IC	XX00 X000b
009Bh			
009Ch	$\overline{\text{INT2}}$ Interrupt Control Register	INT2IC	XX00 X000b
009Dh			
009Eh	$\overline{\text{INT0}}$ Interrupt Control Register	INT0IC	XX00 X000b
009Fh	Exit Priority Register	RLVL	XXXX 0000b
00A0h to 02BFh			

X: Undefined
Blank spaces are all reserved. No access is allowed.

Table 4.5 SFR Address Map (5/11)

Address	Register	Symbol	After Reset
02C0h	X0 Register, Y0 Register	X0R, Y0R	XXXXh
02C1h			
02C2h	X1 Register, Y1 Register	X1R, Y1R	XXXXh
02C3h			
02C4h	X2 Register, Y2 Register	X2R, Y2R	XXXXh
02C5h			
02C6h	X3 Register, Y3 Register	X3R, Y3R	XXXXh
02C7h			
02C8h	X4 Register, Y4 Register	X4R, Y4R	XXXXh
02C9h			
02CAh	X5 Register, Y5 Register	X5R, Y5R	XXXXh
02CBh			
02CCh	X6 Register, Y6 Register	X6R, Y6R	XXXXh
02CDh			
02CEh	X7 Register, Y7 Register	X7R, Y7R	XXXXh
02CFh			
02D0h	X8 Register, Y8 Register	X8R, Y8R	XXXXh
02D1h			
02D2h	X9 Register, Y9 Register	X9R, Y9R	XXXXh
02D3h			
02D4h	X10 Register, Y10 Register	X10R, Y10R	XXXXh
02D5h			
02D6h	X11 Register, Y11 Register	X11R, Y11R	XXXXh
02D7h			
02D8h	X12 Register, Y12 Register	X12R, Y12R	XXXXh
02D9h			
02DAh	X13 Register, Y13 Register	X13R, Y13R	XXXXh
02DBh			
02DCh	X14 Register, Y14 Register	X14R, Y14R	XXXXh
02DDh			
02DEh	X15 Register, Y15 Register	X15R, Y15R	XXXXh
02DFh			
02E0h	X/Y Control Register	XYC	XXXX XX00b
02E1h			
02E2h			
02E3h			
02E4h	UART1 Special Mode Register 4	U1SMR4	00h
02E5h	UART1 Special Mode Register 3	U1SMR3	00h
02E6h	UART1 Special Mode Register 2	U1SMR2	00h
02E7h	UART1 Special Mode Register	U1SMR	00h
02E8h	UART1 Transmit/Receive Mode Register	U1MR	00h
02E9h	UART1 Baud Rate Register	U1BRG	XXh
02EAh	UART1 Transmit Buffer Register	U1TB	XXXXh
02EBh			
02ECh	UART1 Transmit/Receive Control Register 0	U1C0	0000 1000b
02EDh	UART1 Transmit/Receive Control Register 1	U1C1	0000 0010b
02EEh	UART1 Receive Buffer Register	U1RB	XXXXh
02EFh			

X: Undefined
Blank spaces are all reserved. No access is allowed.

Table 4.6 SFR Address Map (6/11)

Address	Register	Symbol	After Reset
02F0h			
02F1h			
02F2h			
02F3h			
02F4h	UART4 Special Mode Register 4	U4SMR4	00h
02F5h	UART4 Special Mode Register 3	U4SMR3	00h
02F6h	UART4 Special Mode Register 2	U4SMR2	00h
02F7h	UART4 Special Mode Register	U4SMR	00h
02F8h	UART4 Transmit/Receive Mode Register	U4MR	00h
02F9h	UART4 Baud Rate Register	U4BRG	XXh
02FAh	UART4 Transmit Buffer Register	U4TB	XXXXh
02FBh			
02FCh	UART4 Transmit/Receive Control Register 0	U4C0	0000 1000b
02FDh	UART4 Transmit/Receive Control Register 1	U4C1	0000 0010b
02FEh	UART4 Receive Buffer Register	U4RB	XXXXh
02FFh			
0300h	Timer B3, B4, B5 Count Start Register	TBSR	000X XXXXb
0301h			
0302h	Timer A11 Register	TA11	XXXXh
0303h			
0304h	Timer A21 Register	TA21	XXXXh
0305h			
0306h	Timer A41 Register	TA41	XXXXh
0307h			
0308h	Three-Phase PWM Control Register 0	INVC0	00h
0309h	Three-Phase PWM Control Register 1	INVC1	00h
030Ah	Three-Phase Output Buffer Register 0	IDB0	XX11 1111b
030Bh	Three-Phase Output Buffer Register 1	IDB1	XX11 1111b
030Ch	Dead Time Timer	DTT	XXh
030Dh	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	XXh
030Eh			
030Fh			
0310h	Timer B3 Register	TB3	XXXXh
0311h			
0312h	Timer B4 Register	TB4	XXXXh
0313h			
0314h	Timer B5 Register	TB5	XXXXh
0315h			
0316h			
0317h			
0318h			
0319h			
031Ah			
031Bh	Timer B3 Mode Register	TB3MR	00XX 0000b
031Ch	Timer B4 Mode Register	TB4MR	00XX 0000b
031Dh	Timer B5 Mode Register	TB5MR	00XX 0000b
031Eh			
031Fh	External Interrupt Source Select Register	IFSR	00h

X: Undefined
Blank spaces are all reserved. No access is allowed.

Table 4.7 SFR Address Map (7/11)

Address	Register	Symbol	After Reset
0320h			
0321h			
0322h			
0323h			
0324h	UART3 Special Mode Register 4	U3SMR4	00h
0325h	UART3 Special Mode Register 3	U3SMR3	00h
0326h	UART3 Special Mode Register 2	U3SMR2	00h
0327h	UART3 Special Mode Register	U3SMR	00h
0328h	UART3 Transmit/Receive Mode Register	U3MR	00h
0329h	UART3 Baud Rate Register	U3BRG	XXh
032Ah	UART3 Transmit Buffer Register	U3TB	XXXXh
032Bh			
032Ch	UART3 Transmit/Receive Control Register 0	U3C0	0000 1000b
032Dh	UART3 Transmit/Receive Control Register 1	U3C1	0000 0010b
032Eh	UART3 Receive Buffer Register	U3RB	XXXXh
032Fh			
0330h			
0331h			
0332h			
0333h			
0334h	UART2 Special Mode Register 4	U2SMR4	00h
0335h	UART2 Special Mode Register 3	U2SMR3	00h
0336h	UART2 Special Mode Register 2	U2SMR2	00h
0337h	UART2 Special Mode Register	U2SMR	00h
0338h	UART2 Transmit/Receive Mode Register	U2MR	00h
0339h	UART2 Baud Rate Register	U2BRG	XXh
033Ah	UART2 Transmit Buffer Register	U2TB	XXXXh
033Bh			
033Ch	UART2 Transmit/Receive Control Register 0	U2C0	0000 1000b
033Dh	UART2 Transmit/Receive Control Register 1	U2C1	0000 0010b
033Eh	UART2 Receive Buffer Register	U2RB	XXXXh
033Fh			
0340h	Count Start Register	TABSR	00h
0341h	Clock Prescaler Reset Register	CPSRF	0XXX XXXXb
0342h	One-Shot Start Register	ONSF	00h
0343h	Trigger Select Register	TRGSR	00h
0344h	Up/Down Select Register	UDF	00h
0345h			
0346h	Timer A0 Register	TA0	XXXXh
0347h			
0348h	Timer A1 Register	TA1	XXXXh
0349h			
034Ah	Timer A2 Register	TA2	XXXXh
034Bh			
034Ch	Timer A3 Register	TA3	XXXXh
034Dh			
034Eh	Timer A4 Register	TA4	XXXXh
034Fh			

X: Undefined
Blank spaces are all reserved. No access is allowed.

Table 4.8 SFR Address Map (8/11)

Address	Register	Symbol	After Reset
0350h	Timer B0 Register	TB0	XXXXh
0351h			
0352h	Timer B1 Register	TB1	XXXXh
0353h			
0354h	Timer B2 Register	TB2	XXXXh
0355h			
0356h	Timer A0 Mode Register	TA0MR	00h
0357h	Timer A1 Mode Register	TA1MR	00h
0358h	Timer A2 Mode Register	TA2MR	00h
0359h	Timer A3 Mode Register	TA3MR	00h
035Ah	Timer A4 Mode Register	TA4MR	00h
035Bh	Timer B0 Mode Register	TB0MR	00XX 0000b
035Ch	Timer B1 Mode Register	TB1MR	00XX 0000b
035Dh	Timer B2 Mode Register	TB2MR	00XX 0000b
035Eh	Timer B2 Special Mode Register	TB2SC	XXXX XXX0b
035Fh	Count Source Prescaler Register(1)	TCSPR	0XXX 0000b
0360h			
0361h			
0362h			
0363h			
0364h	UART0 Special Mode Register 4	U0SMR4	00h
0365h	UART0 Special Mode Register 3	U0SMR3	00h
0366h	UART0 Special Mode Register 2	U0SMR2	00h
0367h	UART0 Special Mode Register	U0SMR	00h
0368h	UART0 Transmit/Receive Mode Register	U0MR	00h
0369h	UART0 Baud Rate Register	U0BRG	XXh
036Ah	UART0 Transmit Buffer Register	U0TB	XXXXh
036Bh			
036Ch	UART0 Transmit/Receive Control Register 0	U0C0	0000 1000b
036Dh	UART0 Transmit/Receive Control Register 1	U0C1	0000 0010b
036Eh	UART0 Receive Buffer Register	U0RB	XXXXh
036Fh			
0370h			
0371h			
0372h			
0373h			
0374h			
0375h			
0376h			
0377h			
0378h	DMA0 Request Source Select Register	DM0SL	0X00 0000b
0379h	DMA1 Request Source Select Register	DM1SL	0X00 0000b
037Ah	DMA2 Request Source Select Register	DM2SL	0X00 0000b
037Bh	DMA3 Request Source Select Register	DM3SL	0X00 0000b
037Ch	CRC Data Register	CRCD	XXXXh
037Dh			
037Eh	CRC Input Register	CRCIN	XXh
037Fh			

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTE:

1. The TCSPR register maintains values set before reset, even after software reset or watchdog timer reset has been performed.

Table 4.9 SFR Address Map (9/11)

Address	Register	Symbol	After Reset
0380h	A/D0 Register 0	AD00	00XXh
0381h			
0382h	A/D0 Register 1	AD01	00XXh
0383h			
0384h	A/D0 Register 2	AD02	00XXh
0385h			
0386h	A/D0 Register 3	AD03	00XXh
0387h			
0388h	A/D0 Register 4	AD04	00XXh
0389h			
038Ah	A/D0 Register 5	AD05	00XXh
038Bh			
038Ch	A/D0 Register 6	AD06	00XXh
038Dh			
038Eh	A/D0 Register 7	AD07	00XXh
038Fh			
0390h			
0391h			
0392h	A/D0 Control Register 4	AD0CON4	XXXX 00XXb
0393h			
0394h	A/D0 Control Register 2	AD0CON2	XX0X X000b
0395h	A/D0 Control Register 3	AD0CON3	XXXX X000b
0396h	A/D0 Control Register 0	AD0CON0	00h
0397h	A/D0 Control Register 1	AD0CON1	00h
0398h	D/A Register 0	DA0	XXh
0399h			
039Ah	D/A Register 1	DA1	XXh
039Bh			
039Ch	D/A Control Register	DACON	XXXX XX00b
039Dh			
039Eh			
039Fh			

X: Undefined
Blank spaces are all reserved. No access is allowed.

Table 4.10 SFR Address Map (10/11)

Address	Register	Symbol	After Reset
03A0h			
03A1h			
03A2h			
03A3h			
03A4h			
03A5h			
03A6h			
03A7h			
03A8h			
03A9h			
03AAh			
03ABh			
03ACh			
03ADh			
03AEh			
03AFh	Function Select Register C	PSC	00X0 0000b
03B0h	Function Select Register A0	PS0	00h
03B1h	Function Select Register A1	PS1	00h
03B2h	Function Select Register B0	PSL0	00h
03B3h	Function Select Register B1	PSL1	00h
03B4h	Function Select Register A2	PS2	00X0 0000b
03B5h	Function Select Register A3	PS3	00h
03B6h	Function Select Register B2	PSL2	00X0 0000b
03B7h	Function Select Register B3	PSL3	00h
03B8h			
03B9h			
03BAh			
03BBh			
03BCh			
03BDh			
03BEh			
03BFh			
03C0h	Port P6 Register	P6	XXh
03C1h	Port P7 Register	P7	XXh
03C2h	Port P6 Direction Register	PD6	00h
03C3h	Port P7 Direction Register	PD7	00h
03C4h	Port P8 Register	P8	XXh
03C5h	Port P9 Register	P9	XXh
03C6h	Port P8 Direction Register	PD8	00X0 0000b
03C7h	Port P9 Direction Register	PD9	00h
03C8h	Port P10 Register	P10	XXh
03C9h	Port P11 Register(1)	P11	XXh
03CAh	Port P10 Direction Register	PD10	00h
03CBh	Port P11 Direction Register(1)(2)	PD11	XXX0 0000b
03CCh	Port P12 Register(1)	P12	XXh
03CDh	Port P13 Register(1)	P13	XXh
03CEh	Port P12 Direction Register(1)(2)	PD12	00h
03CFh	Port P13 Direction Register(1)(2)	PD13	00h

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:

1. These registers cannot be used in the 100-pin package.
2. Set to FFh in the 100-pin package.

Table 4.11 SFR Address Map (11/11)

Address	Register	Symbol	After Reset
03D0h	Port P14 Register(1)	P14	XXh
03D1h	Port P15 Register(1)	P15	XXh
03D2h	Port P14 Direction Register(1)(2)	PD14	X000 0000b
03D3h	Port P15 Direction Register(1)(2)	PD15	00h
03D4h			
03D5h			
03D6h			
03D7h			
03D8h			
03D9h			
03DAh	Pull-Up Control Register 2	PUR2	00h
03DBh	Pull-Up Control Register 3	PUR3	00h
03DCh	Pull-Up Control Register 4(1)(3)	PUR4	XXXX 0000b
03DDh			
03DEh			
03DFh			
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			
03EDh			
03EEh			
03EFh			
03F0h	Pull-Up Control Register 0	PUR0	00h
03F1h	Pull-Up Control Register 1	PUR1	XXXX 0000b
03F2h			
03F3h			
03F4h			
03F5h			
03F6h			
03F7h			
03F8h			
03F9h			
03FAh			
03FBh			
03FCh			
03FDh			
03FEh			
03FFh	Port Control Register	PCR	XXXX X000b

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:

1. These registers cannot be used in the 100-pin package.
2. Set to FFh in the 100-pin package.
3. Set to 00h in the 100-pin package.

5. Reset

Hardware reset, software reset and watchdog timer reset are implemented to reset the MCU.

5.1 Hardware Reset

Pins, CPU, and SFRs are reset by using the RESET pin. When a low-level (“L”) signal is applied to the RESET pin while the supply voltage meets the recommended operating conditions, ports and I/O pins for peripheral functions are reset. (Refer to **Table 5.1 Pin states while RESET pin is held “L”**.) Also, the oscillation circuit is reset and the main clock starts oscillating. CPU and SFRs are reset when the signal applied to the RESET pin changes from “L” to high-level (“H”) signal, and then the MCU executes a program beginning with the address indicated by the reset vector. The internal RAM is not reset by hardware reset. When an “L” signal is applied to the RESET pin while writing data to the internal RAM, the value written to the internal RAM becomes undefined.

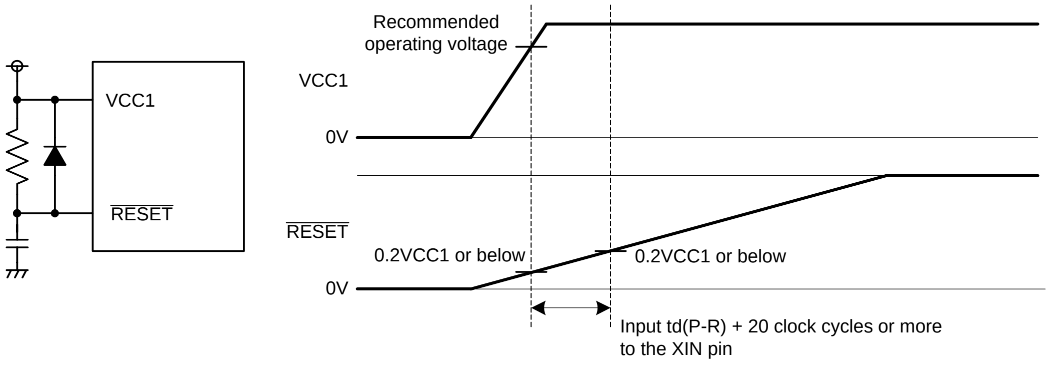
Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows a reset sequence. Table 5.1 lists pin states while the RESET pin is held “L”.

5.1.1 Reset at a Stable Supply Voltage

(1) Apply an “L” signal to the RESET pin.
(2) Input 20 clock cycles or more into the XIN pin.
(3) Apply an “H” signal to the RESET pin.

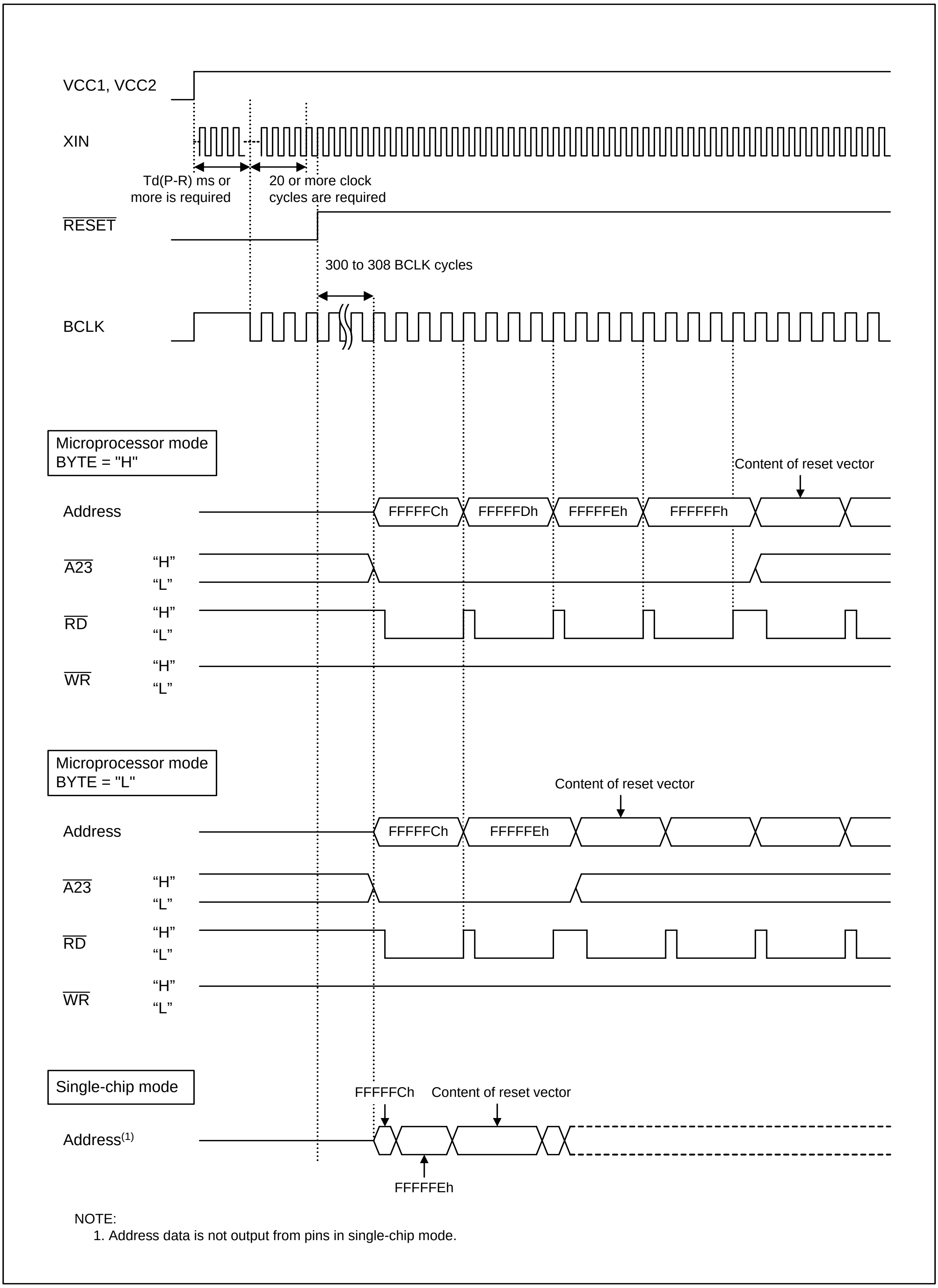
5.1.2 Power-on Reset

(1) Apply an “L” signal to the RESET pin.
(2) Increase the supply voltage until it meets the recommended operating condition.
(3) Wait for *td(P-R)* (internal power supply stabilization time) or more to allow the internal power supply to stabilize.
(4) Inputs 20 clock cycles or more into the XIN pin.
(5) Apply an “H” signal to the RESET pin.


VCC1
RESET
Recommended operating voltage
VCC1
0V
RESET
0.2VCC1 or below
0.2VCC1 or below
0V
Input td(P-R) + 20 clock cycles or more to the XIN pin

NOTE:
1. If operating at VCC1 > VCC2, VCC2 voltage must be lower than VCC1 voltage when powering up and down.

Figure 5.1 Example of Reset Circuit



NOTE:
1. Address data is not output from pins in single-chip mode.

Figure 5.2 Reset Sequence

Table 5.1 Pin States while $\overline{\text{RESET}}$ Pin is Held "L"(2)

Pin Name	Single-Chip Mode	Microprocessor Mode	
	CNVSS = "L"	CNVSS = "H"(4)	
		BYTE = "L"	BYTE = "H"
P0	Input port (high-impedance)	Data input (high-impedance)	
P1	Input port (high-impedance)	Data input (high-impedance)	Input port (high-impedance)
P2 to P4	Input port (high-impedance)	Address output (undefined)	
P5_0	Input port (high-impedance)	$\overline{\text{WR}}$ signal output ("H")(3)	
P5_1	Input port (high-impedance)	$\overline{\text{BHE}}$ signal output (undefined)	
P5_2	Input port (high-impedance)	$\overline{\text{RD}}$ signal output ("H")(3)	
P5_3	Input port (high-impedance)	BCLK output(3)	
P5_4	Input port (high-impedance)	$\overline{\text{HLDA}}$ signal output (output level depends on an input level to the $\overline{\text{HOLD}}$ pin)(3)	
P5_5	Input port (high-impedance)	$\overline{\text{HOLD}}$ signal input (high-impedance)	
P5_6	Input port (high-impedance)	"H" signal output(3)	
P5_7	Input port (high-impedance)	$\overline{\text{RDY}}$ signal input (high-impedance)	
P6 to P15(1)	Input port (high-impedance)	Input port (high-impedance)	

NOTES:

1. Ports P11 to P15 are provided in the 144-pin package only.
2. The availability of the pull-up resistors is undefined until the internal supply voltage stabilizes.
3. These pin states are defined after the power is turned on and the internal supply voltage stabilizes. Until then, the pin states are undefined.
4. EPM (P5_5) must be "H" in the flash memory version.

5.2 Software Reset

When the PM03 bit in the PM0 register is set to 1 (MCU is reset), the MCU resets the CPU, SFRs, ports, and I/O pins for peripheral functions. And then the MCU executes a program in an address indicated by the reset vector.

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

The software reset does not reset the following SFRs; bits PM01 and PM00 in the PM0 register, and the TCSPR register.

Processor mode remains unchanged since bits PM01 and PM00 are not reset.

5.3 Watchdog Timer Reset

When the CM06 bit in the CM0 register is set to 1 (reset) and the watchdog timer underflows, the MCU resets the CPU, SFRs, ports, and I/O pins for peripheral functions. And then the MCU executes a program in an address indicated by the reset vector.

The watchdog timer reset does not reset the following SFRs; bits PM01 and PM00 in the PM0 register, and the TCSPR register.

Processor mode remains unchanged since bits PM01 and PM00 are not reset.

5.4 Internal Registers

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

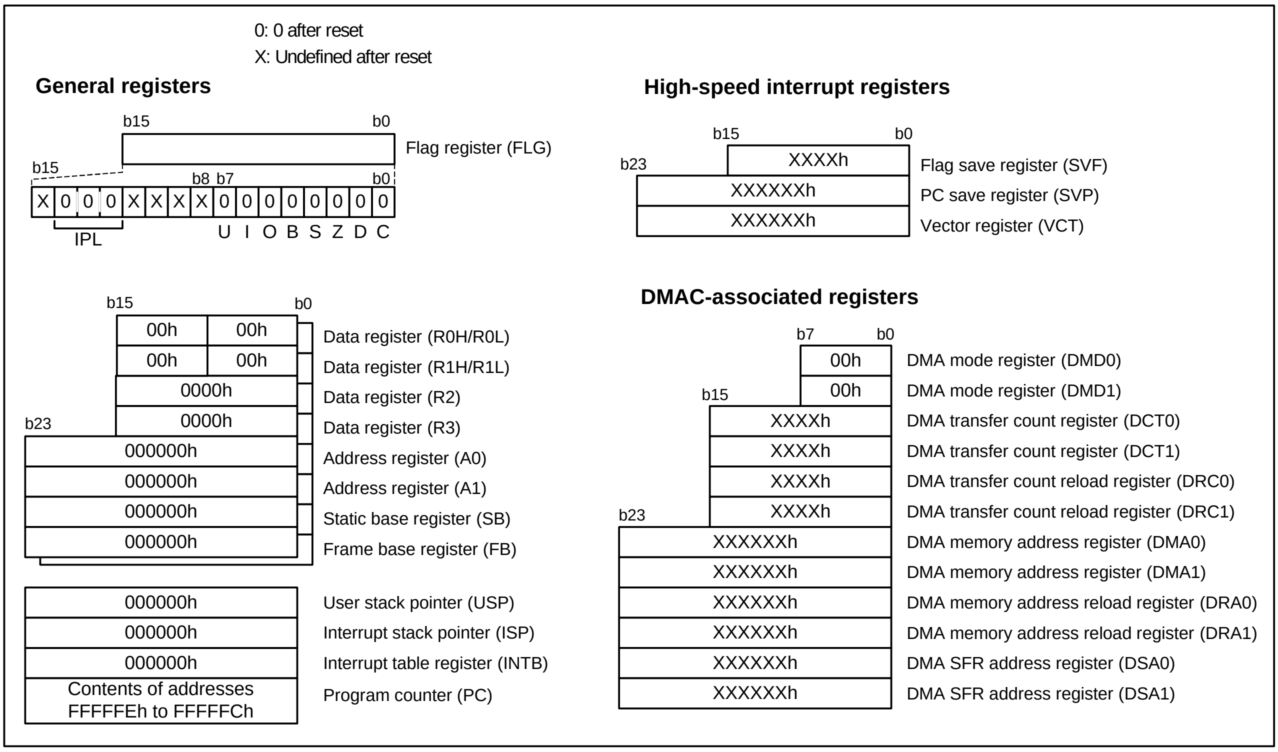
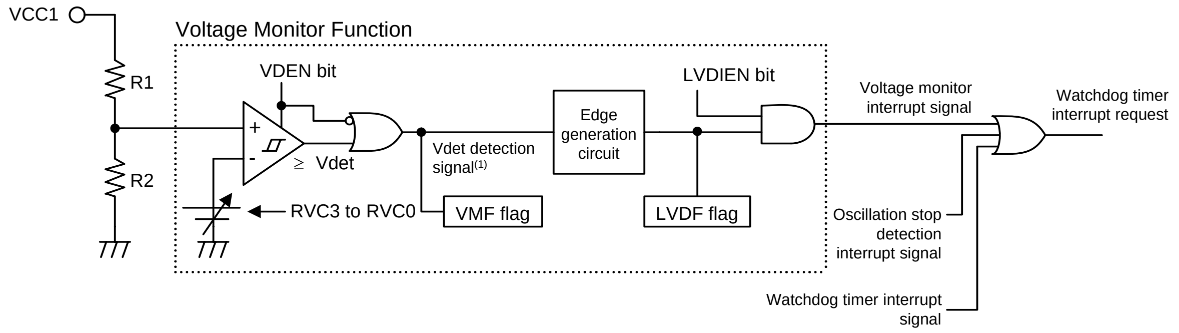


Figure 5.3 CPU Register States after Reset

6. Power Supply Voltage Monitor Function

The power supply voltage monitor function detects the changes in voltage and triggers the interrupts.

Figure 6.1 shows a block diagram of the voltage monitor function. Figures 6.2 and 6.3 show registers associated with the function.


VCC1
R1
R2
Voltage Monitor Function
VDEN bit
+
-
≥ Vdet
RVC3 to RVC0
Vdet detection signal(1)
VMF flag
Edge generation circuit
LVDIEN bit
LVDF flag
Voltage monitor interrupt signal
Watchdog timer interrupt request
Oscillation stop detection interrupt signal
Watchdog timer interrupt signal

VDEN bit, LVDIEN bit, LVDF flag, and VMF flag: Bits in the LVDC register
RVC3 to RVC0: Bits in the DVCR register

NOTE:
1. When the VDEN bit in the LVDC register is set to 0, (Voltage monitor function not used), Vdet detection signal becomes “H” and the VMF bit becomes 1.

Figure 6.1 Power Supply Voltage Monitor Function Block Diagram

Voltage Monitor Register(1)

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

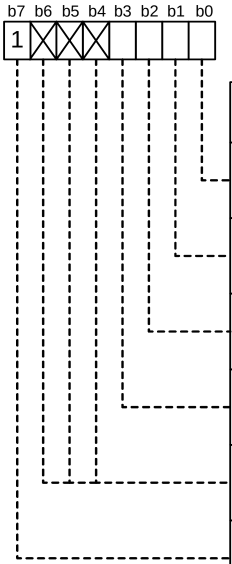
Symbol	Address	After Reset
LVDC	001Bh	0000 1000b

Bit Symbol	Bit Name	Function	RW
VDEN	Voltage monitor function select bit(2)	0: Voltage monitor function not used 1: Voltage monitor function used	RW
LVDIEN	Voltage monitor interrupt enable bit	0: Voltage monitor interrupt disabled 1: Voltage monitor interrupt enabled	RW
LVDF	Voltage change detect flag(3)(4)	0: Not detected 1: Voltage crosses Vdet	RW
VMF	Voltage change monitor flag(3)	0: VCC1 < Vdet (F) 1: VCC1 ≥ Vdet (R)	RO
– (b7-b4)	Reserved bits	Set to 0	RW

NOTES:
1. Set the LVDC register after setting the PRC3 bit in the PRCR register is set to 1 (write enable).
2. Wait for td(E-A) to elapse after the VDEN bit is set to 1 to use the voltage monitor function. To set the LVDIEN bit to 1, wait for td(E-A) to elapse after the VDEN bit is set to 1.
3. Bits LVDF and VMF are enabled when the VDEN bit is 1. When the VDEN bit is 0, the LVDF bit becomes 0 and the VMF bit becomes 1.
4. The LVDF bit can be set to 0 by a program. Writing a 1 has no effect.

Figure 6.2 LVDC Register

Reference Voltage Configuration Register



Symbol	Address	After Reset
DVCR	0017h	1000 1111b

Bit Symbol	Bit Name	Function	RW
RVC0	Reference voltage select bits[2]	b3 b2 b1 b0 0 0 0 0 : 3.80 V 0 0 0 1 : 3.65 V 0 0 1 0 : 3.50 V 0 0 1 1 : 3.35 V 0 1 0 0 : 3.20 V 1 0 1 1 : 4.55 V 1 1 0 0 : 4.40 V 1 1 0 1 : 4.25 V 1 1 1 0 : 4.10 V 1 1 1 1 : 3.95 V Do not set to the value other than the above.	RW
RVC1			RW
RVC2			RW
RVC3			RW
– (b6-b4)	Unimplemented. Write 0. Read as undefined value.		–
– (b7)	Reserved bit	Set to 1	RW

NOTES:

1. Set the DVCR register after setting the PRC3 bit in the PRCR register is set to 1 (write enable) while the VDEN bit in the LVDC register is 0 (Voltage monitor function not used).
2. The detection voltage levels of Vdet(F) and Vdet(R) lists as below.

Bits RVC3 to RVC0	Reference voltage	Detection voltage level (Voltage drops) Vdet(F)	Detection voltage level (Voltage rises) Vdet(R)
1011b	4.55V	4.55V	4.77V
1100b	4.40V	4.40V	4.62V
1101b	4.25V	4.25V	4.47V
1110b	4.10V	4.10V	4.32V
1111b	3.95V	3.95V	4.17V
0000b	3.80V	3.80V	4.02V
0001b	3.65V	3.65V	3.87V
0010b	3.50V	3.50V	3.72V
0011b	3.35V	3.35V	3.57V
0100b	3.20V	3.20V	3.42V

Figure 6.3 DVCR Register

6.1 Operation of Voltage Monitor Function

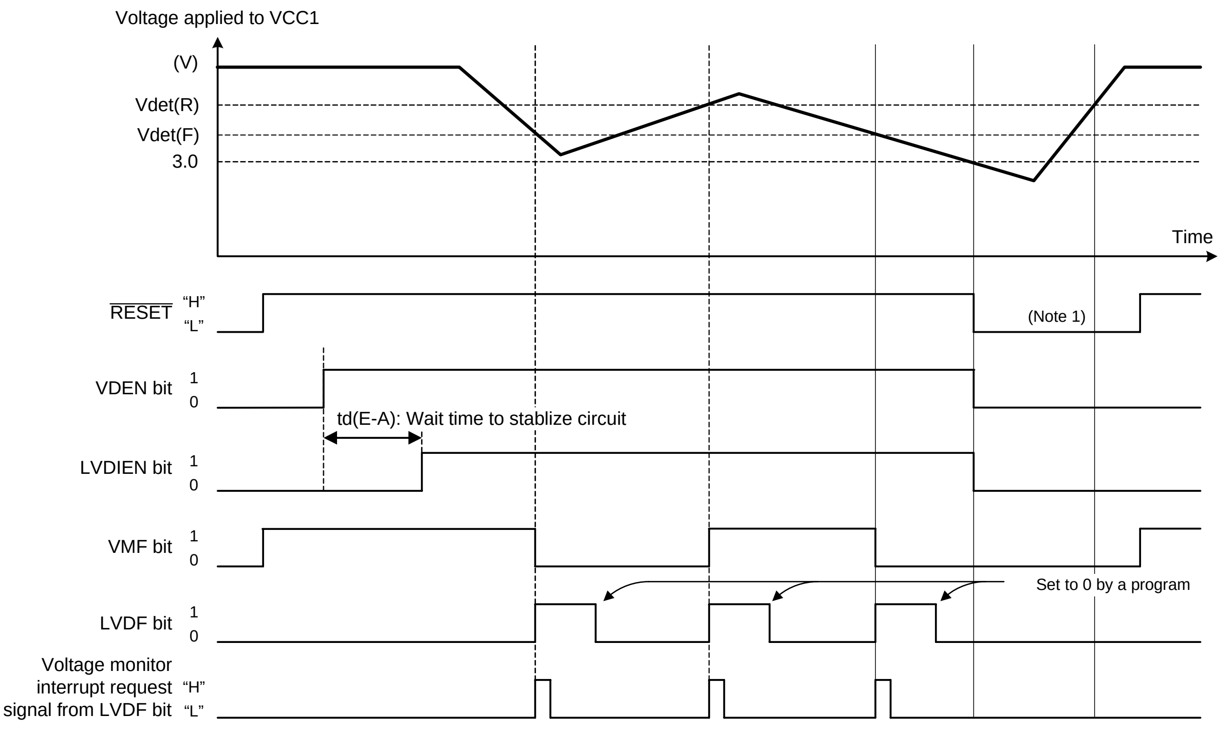
When the VDEN bit in the LVDC register is set to 1 (voltage monitor function used), the voltage monitor function can be used after td(E-A) has elapsed.

When the voltage applied to the VCC1 pin has dropped below Vdet(F), the VMF bit in the LVDC register becomes 0 (VCC1 < Vdet(F)) and the LVDF bit in the LVDC register becomes 1 (voltage crosses Vdet). When the voltage applied to the VCC1 pin has risen above Vdet(R), the VMF bit becomes 1 (VCC1 ≥ Vdet(R)) and the LVDF bit becomes 1.

If the LVDIEN bit in the LVDC register is 1 (voltage monitor interrupt enabled), when the value of the VMF bit is changed, the LVDF bit becomes 1 and a voltage monitor interrupt request is generated. The LVDF bit does not automatically become 0 when an interrupt request is acknowledged. Set it to 0 by a program. Whether the voltage has dropped below Vdet(F) or risen above Vdet(R) can be determined by reading the VMF bit.

The voltage monitor interrupt shares the same interrupt vector with watchdog timer interrupt and oscillation stop detection interrupt. When using the voltage monitor interrupt simultaneously with these interrupts, determine whether the voltage monitor interrupt is generated by reading the LVDF bit in the interrupt routine.

Figure 6.4 shows a voltage monitor function operation example.



VDEN bit, LVDIEN bit, LVDF bit, and VMF bit: Bits in the LVDC register

NOTE:
1. Apply an "L" to the RESET pin when the voltage input to the VCC1 pin drops to 3.0 V or below. After the voltage rises above 3.0 V, and the voltage of the main voltage regulator and the main clock oscillation stabilize, apply an "H" to the RESET pin.

Figure 6.4 Voltage Monitor Function Operation Example

7. Processor Mode

7.1 Processor Mode

Single-chip mode, memory expansion mode, microprocessor mode, or boot mode can be selected as the processor mode. Table 7.1 lists the features of the processor mode.

Table 7.1 Processor Mode Features

Processor Mode	Accessible Space	Pins assigned to I/O Port
Single-chip mode	SFR, internal RAM, internal ROM (user ROM area)	Used as I/O ports or I/O pins for peripheral functions
Memory expansion mode(1)	SFR, internal RAM, internal ROM (user ROM area), external space	P0 to P5 become bus control pins
Microprocessor mode(1)	SFR, internal RAM, external space	P0 to P5 become bus control pins
Boot mode(2)	SFR, internal RAM, internal ROM (boot ROM area)	Used as I/O ports or I/O pins for peripheral functions

NOTES:
1. Refer to **8. Bus** for details.
2. Refer to **25.** Flash Memory for details.

7.2 Setting of Processor Mode

The CNVSS pin, $\overline{\text{EPM}}$(P5_5) pin, and bits PM01 and PM00 in the PM0 register determine which processor mode to select. Table 7.2 lists processor mode after hardware reset. Table 7.3 lists the processor mode selected by bits PM01 and PM00.

Table 7.2 Processor Mode after Hardware Reset

Input to mode entry pins		Chip Type	Processor Mode
CNVSS pin	$\overline{\text{EPM}}$(P5_5)		
L	H or L	Flash memory version	Single-chip mode
H	H(1)	Flash memory version, ROMless version	Microprocessor mode
H	L	Flash memory version	Boot mode

NOTE:
1. P5_5 functions as the $\overline{\text{HOLD}}$ pin after reset.

Table 7.3 PM01 and PM00 Bits Setting and Processor Mode

Bits PM01 and PM00	Processor Mode
00b	Single-chip mode
01b	Memory expansion mode
11b	Microprocessor mode

Rewriting bits PM01 and PM00 in the PM0 register places the MCU in the corresponding processor mode regardless of the CNVSS input level. When using memory expansion mode or microprocessor mode, first set bits PM02, PM05 and PM04, and PM07 in the PM0 register, and also set bits PM11 and PM10, PM15 and PM14 in the PM1 register. Then, set bits PM01 and PM00.

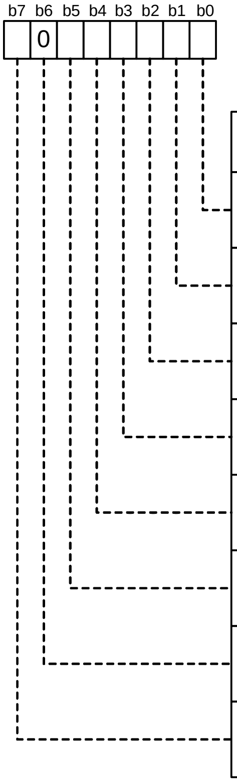
Do not enter microprocessor mode while the CPU is executing the program in the internal ROM.

Do not enter single-chip mode from microprocessor mode while the CPU is executing the program in an external space.

The internal ROM cannot be accessed regardless of the PM01 and PM00 bits setting if the MCU starts up in microprocessor mode after reset.

Figures 7.1 and 7.2 show the PM0 register and PM1 register. Figure 7.3 shows a memory map in each processor mode.

Processor Mode Register 0[1]



Symbol	Address	After Reset
PM0	0004h	1000 0000b (CNVSS = “L”) 0000 0011b (CNVSS = “H”)

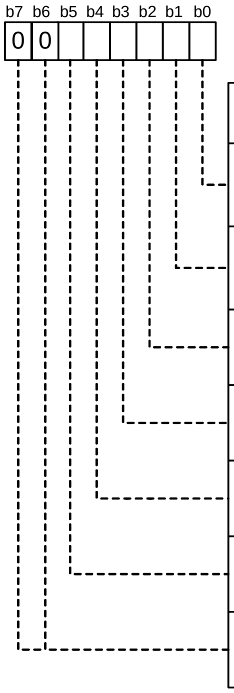
Bit Symbol	Bit Name	Function	RW
PM00	Processor mode bits[2, 3]	b1 b0 0 0: Single-chip mode 0 1: Memory expansion mode 1 0: Do not set to this value 1 1: Microprocessor mode	RW
PM01			RW
PM02	R/W mode select bit	0: $\overline{RD}$/$\overline{BHE}$/$\overline{WR}$ 1: $\overline{RD}$/$\overline{WRH}$/$\overline{WRL}$	RW
PM03	Software reset bit	The MCU is reset when this bit is set to 1. Read as 0.	RW
PM04	Multiplexed bus space select bits[4]	b5 b4 0 0: Multiplexed bus is not used 0 1: Access the $\overline{CS2}$ area using multiplexed bus 1 0: Access the $\overline{CS1}$ area using multiplexed bus 1 1: Access all $\overline{CS}$ areas using multiplexed bus	RW
PM05			RW
– (b6)	Reserved bit	Set to 0	RW
PM07	BCLK output function select bit	0: BCLK output [5] 1: No BCLK output	RW

NOTES:

1. Set the PM0 register after the PRC1 bit in the PRCR register is set to 1 (write enable).
2. Bits PM01 and PM00 maintain values set before reset, even after software reset or watchdog timer reset has performed.
3. When using memory expansion mode or microprocessor mode, first set bits PM02, PM05 and PM04, and PM07 in the PM0 register, and also set bits PM11 and PM10, PM15 and PM14 in the PM1 register. Then, set bits PM01 and PM00.
4. The PM05 and PM04 bits setting is enabled in memory expansion mode and microprocessor mode. Set these bits in the combination with bits PM11 and PM10 in the PM 1 register. Do not set bits PM05 and PM04 to 11b in microprocessor mode since the MCU starts up with the separate bus after reset. Refer to the **Table “Multiplexed Bus Settings and Chip-Select Areas”** in the Bus chapter.
5. No BCLK is output in single-chip mode even if the PM07 bit is set to 0.
   To output BCLK from P5_3 in memory expansion mode and microprocessor mode, set the PM07 bit to 0, bits CM01 and CM00 in the CM0 register to “00b” (I/O port P5_3), and bits PM15 and PM14 in the PM1 register to 00b, 10b, or 11b.

Figure 7.1 PM0 Register

Processor Mode Register 1[1]

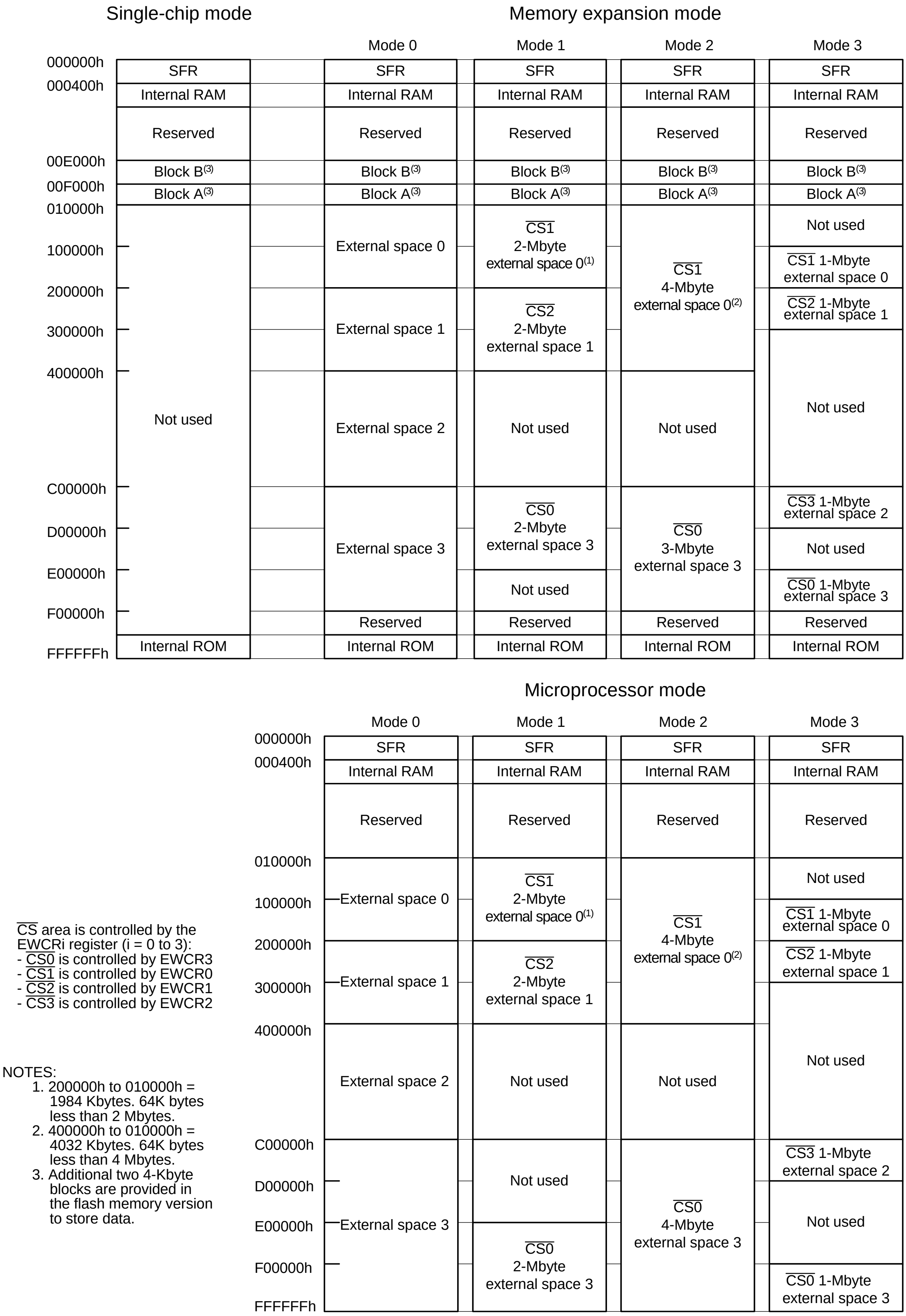


Symbol	Address	After Reset
PM1	0005h	00h

Bit Symbol	Bit Name	Function	RW
PM10	External space mode bits[2]	b1 b0 0 0: Mode 0 (A20 to $\overline{\text{A23}}$ for P4_4 to P4_7) 0 1: Mode 1 (A20 for P4_4, $\overline{\text{CS2}}$ to $\overline{\text{CS0}}$ for P4_5 to P4_7) 1 0: Mode 2 (A20 and A21 for P4_4 and P4_5, $\overline{\text{CS1}}$ and $\overline{\text{CS0}}$ for P4_6 and P4_7) 1 1: Mode 3 ($\overline{\text{CS3}}$ to $\overline{\text{CS0}}$ for P4_4 to P4_7)	RW
PM11			RW
PM12	Internal memory wait bit	0: No wait state 1: 1 wait state	RW
PM13	SFR area wait bit	0: 1 wait state 1: 2 wait states	RW
PM14	ALE pin select bits[2]	b5 b4 0 0: No ALE 0 1: P5_3[3] 1 0: P5_6 1 1: P5_4	RW
PM15			RW
– (b7-b6)	Reserved bits	Set to 0	RW

NOTES:
1. Set the PM1 register after the PRC1 bit in the PRCR register is set to 1 (write enable).
2. The PM11 and PM10 bits settings are enabled in memory expansion mode and microprocessor mode. Set bits PM01 and PM00 after setting bits PM15 and PM14, and bits PM11 and PM10.
3. To output ALE signal from P5_3, set bits PM15 and PM14 to 01b, and bits CM01 and CM00 in the CM0 register to 00b (I/O port P5_3).

Figure 7.2 PM1 Register



$\overline{\text{CS}}$ area is controlled by the $\overline{\text{EWCR}}$i register (i = 0 to 3):
- $\overline{\text{CS0}}$ is controlled by EWCR3
- $\overline{\text{CS1}}$ is controlled by EWCR0
- $\overline{\text{CS2}}$ is controlled by EWCR1
- $\overline{\text{CS3}}$ is controlled by EWCR2

NOTES:
1. 200000h to 010000h = 1984 Kbytes. 64K bytes less than 2 Mbytes.
2. 400000h to 010000h = 4032 Kbytes. 64K bytes less than 4 Mbytes.
3. Additional two 4-Kbyte blocks are provided in the flash memory version to store data.

Figure 7.3 Memory Map in Each Processor Mode

8. Bus

In memory expansion mode or microprocessor mode, the following pins become bus control pins: D0 to D15, A0 to A22, $\overline{A23}$, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WRL}/\overline{WR}$, $\overline{WRH}/\overline{BHE}$, $\overline{RD}$, CLKOUT/BCLK/ALE, $\overline{HLDA}$/ALE, $\overline{HOLD}$, ALE, and $\overline{RDY}$.

8.1 Bus Settings

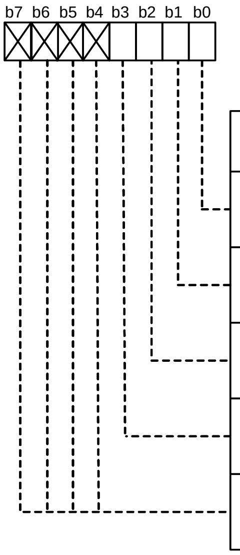
Bus setting is determined by the BYTE pin, the DS register, bits PM05 and PM04 in the PM0 register, and bits PM11 and PM10 in the PM1 register.

Table 8.1 lists bus settings. Figure 8.1 shows the DS register.

Table 8.1 Bus Settings

Bus Setting	Pin & Registers Used for Setting
Selecting external data bus width	DS register
Setting bus width after reset	BYTE pin (for external space 3 only)
Selecting separate bus or multiplexed bus	Bits PM05 and PM04 in the PM0 register
Number of chip-select pins	Bits PM11 and PM10 in the PM1 register

External Data Bus Width Control Register



Symbol	Address	After Reset
DS	000Bh	XXXX 1000b (BYTE pin = "L") XXXX 0000b (BYTE pin = "H")

Bit Symbol	Bit Name	Function	RW
DS0	External space 0 data bus width select bit	0: 8 bits wide 1: 16 bits wide	RW
DS1	External space 1 data bus width select bit	0: 8 bits wide 1: 16 bits wide	RW
DS2	External space 2 data bus width select bit	0: 8 bits wide 1: 16 bits wide	RW
DS3	External space 3 data bus width select bit	0: 8 bits wide 1: 16 bits wide	RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

Figure 8.1 DS Register

8.1.1 Selecting External Address Bus

The number of external address bus pins, the number of chip-select pins, and chip-select-assigned address space ($\overline{\text{CS}}$ area) vary in each external space mode. Bits PM11 and PM10 in the PM1 register select external space mode.

8.1.2 Selecting External Data Bus

The DS register selects either external 8-bit data bus or 16-bit data bus per each external space. After reset, the data bus in the external space 3 becomes 16 bits wide when a low-level (“L”) signal is applied to the BYTE pin, and 8 bits wide when a high-level (“H”) signal is applied. Do not change the BYTE pin level while the MCU is operating. Internal bus is always 16 bits wide.

8.1.3 Selecting Separate Bus/Multiplexed Bus

Bits PM05 and PM04 in the PM0 register select either the separate bus or multiplexed bus. The MCU starts up with the separate bus after reset.

8.1.3.1 Separate Bus

With the separate bus format, the MCU performs data input/output and address output using individual buses. The DS register selects 8-bit or 16-bit external data bus for each external space. If all DSi bits in the DS register (i = 0 to 3) are set to 0 (8-bit data bus), port P0 functions as the data bus and port P1 as the programmable I/O port.

If any of the DSi bits is set to 1 (16-bit data bus), ports P0 and P1 function as the data bus. Port P1 output is undefined when the MCU accesses the space where its DSi bit is set to 0.

8.1.3.2 Multiplexed Bus

With the multiplexed bus format, the MCU performs data input/output and address output using the same bus by time-sharing. D0 to D7 are time-multiplexed with A0 to A7 in the space accessed by the 8-bit data bus. D0 to D15 are time-multiplexed with A0 to A15 in the space accessed by the 16-bit data bus.

When bits PM05 and PM04 in the PM0 register are set to 11b (access all $\overline{\text{CS}}$ area using multiplexed bus), address bus has only 16 bits using A0 to A15. In this case, the accessible space is 64 Kbytes per each chip-select output. Refer to **Table 8.3 Processor Mode and Pin Function** for details.

Table 8.2 lists multiplexed bus settings and chip-select areas.

Table 8.2 Multiplexed Bus Settings and Chip-Select Areas

PM05 and PM04 bits setting	PM11 and PM10 Bits Setting			
	00b (external space mode 0	01b (external space mode 1)	10b (external space mode 2)	11b (external space mode 3)
00b (multiplexed bus not used)	Separate bus			
01b (access the $\overline{\text{CS2}}$ area using multiplexed bus)	Do not set to these values	$\overline{\text{CS2}}$	Do not set to this value	$\overline{\text{CS2}}$
10b (access the $\overline{\text{CS1}}$ area using multiplexed bus)		$\overline{\text{CS1}}$	$\overline{\text{CS1}}$	$\overline{\text{CS1}}$
11b (access the all $\overline{\text{CS}}$ areas using multiplexed bus)(1)		$\overline{\text{CS0}}$ $\overline{\text{CS1}}$ $\overline{\text{CS2}}$	$\overline{\text{CS0}}$ $\overline{\text{CS1}}$	$\overline{\text{CS0}}$ $\overline{\text{CS1}}$ $\overline{\text{CS2}}$ $\overline{\text{CS3}}$

NOTE:

1. In microprocessor mode, do not set bits PM05 and PM04 in the PM0 register to 11b (access all $\overline{\text{CS}}$ areas using multiplexed bus).

Table 8.3 Processor Mode and Pin Function

Processor Mode	Single-chip Mode	Memory Expansion Mode/Microprocessor Mode				Memory Expansion Mode	
PM05 and PM04 bits setting[1]		00b (Multiplexed bus not used)		01b (Access $\overline{\mathrm{CS2}}$ area using multiplexed bus) 10b (Access $\overline{\mathrm{CS1}}$ area using multiplexed bus)		11b (Access all $\overline{\mathrm{CS}}$ areas using multiplexed bus)	
Data bus width		Access all external spaces with 8-bit data bus	Access any external spaces with 16-bit data bus	Access all external spaces with 8-bit data bus	Access any external spaces with 16-bit data bus	Access all external spaces with 8-bit data bus	Access any external spaces with 16-bit data bus
P0_0 to P0_7	I/O port	Data bus (D0 to D7)				I/O port	
P1_0 to P1_7		I/O port	Data bus (D8 to D15)	I/O port	Data bus (D8 to D15)		
P2_0 to P2_7		Address bus (A0 to A7)		Address bus/data bus (A0/D0 to A7/D7)[2]			
P3_0 to P3_7		Address bus (A8 to A15)			Address bus/ data bus (A8/D8 to A15/D15)[2]	Address bus (A8 to A15)	Address bus/ data bus (A8/D8 to A15/D15)[2]
P4_0 to P4_3		Address Bus (A16 to A19)				I/O port	
P4_4 to P4_6		$\overline{\mathrm{CS}}$ or address bus (A20 to A22) (Refer to **8.2 Bus Control** for details)[6]					
P4_7		$\overline{\mathrm{CS}}$ or address bus ($\overline{\mathrm{A23}}$) (Refer to **8.2 Bus Control** for details)[6]					
P5_0 to P5_2		$\overline{\mathrm{RD}}$, $\overline{\mathrm{WRL}}$, $\overline{\mathrm{WRH}}$ outputs or $\overline{\mathrm{RD}}$, $\overline{\mathrm{BHE}}$, $\overline{\mathrm{WR}}$ outputs (Refer to **8.2 Bus Control** for details)[4]					
P5_3	I/O port/ CLKOUT	CLKOUT/BCLK/ALE[7]					
P5_4	I/O port	$\overline{\mathrm{HLDA}}$/ALE[3]					
P5_5		$\overline{\mathrm{HOLD}}$					
P5_6		ALE[3][5]					
P5_7		$\overline{\mathrm{RDY}}$					

NOTES:

1. Do not set bits PM05 and PM04 in the PM0 register to 11b (access all $\overline{\mathrm{CS}}$ areas using multiplexed bus) in microprocessor mode since the MCU starts up with the separate bus after reset. When bits PM05 and PM04 are set to 11b in memory expansion mode, the accessible space is 64-Kbyte per each chip-select output.
2. These pins are used as address bus when selecting separate bus.
3. Bits PM15 and PM14 in the PM1 register determine which pin is used to output the ALE signal.
4. The PM02 bit in the PM0 register selects either combination “$\overline{\mathrm{RD}}$, $\overline{\mathrm{WRL}}$, and $\overline{\mathrm{WRH}}$” or “$\overline{\mathrm{RD}}$, $\overline{\mathrm{BHE}}$, and $\overline{\mathrm{WR}}$”.
5. P5_6 outputs undefined value when bits PM15 and PM14 are set to 00b (no ALE). In this case, it cannot be used as an I/O port.
6. Bits PM11 and PM10 in the PM1 register determine whether these pins are used as chip-select outputs or address bus.
7. Use bits CM01 and CM00 in the CM0 register, bits PM15 and PM14 in the PM1 register, and the PM07 bit in the PM0 register to select among CLKOUT, BCLK, and ALE function.

8.2 Bus Control

Described below are the signals and bus timing required to access external devices. The signals are available in memory expansion mode and microprocessor mode only.

8.2.1 Address Bus and Data Bus

Address bus is the signals to access 16-Mbyte space, and consists of 24 control pins; A0 to A22 and $\overline{\text{A23}}$. $\overline{\text{A23}}$ is an inverse output signal of the highest-order address bit.

Data bus is the signals for data input and output. The DS register selects either an 8-bit data bus width from D0 to D7 or a 16-bit data bus width from D0 to D15 for each external space. When a high-level (“H”) signal is applied to the BYTE pin, the data bus accessing the external space 3 is 8 bits wide after reset. When a low-level (“L”) signal is applied to the BYTE pin, the data bus accessing the external space 3 is 16 bits wide.

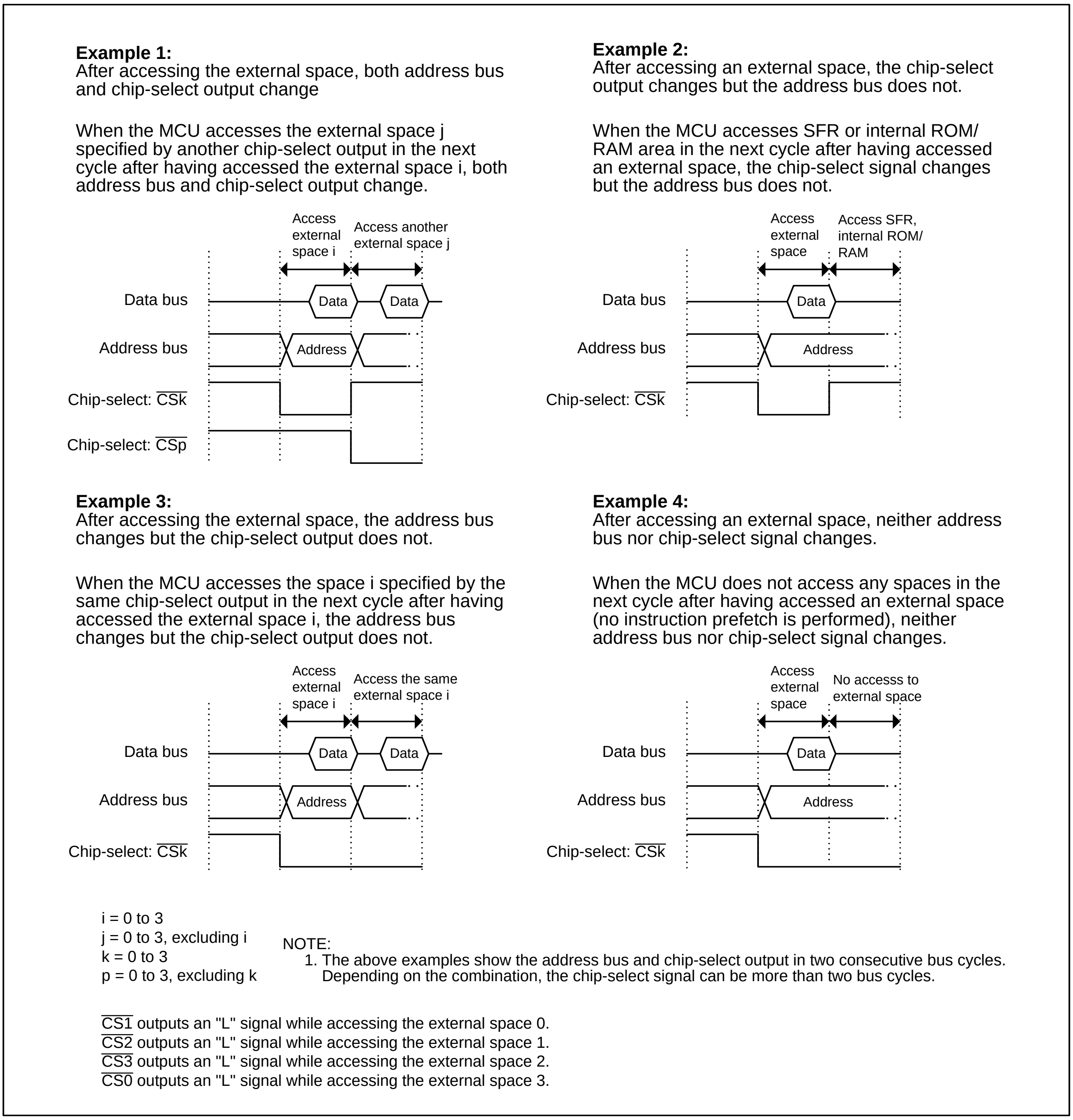
When changing single-chip mode to memory expansion mode, the address bus value is undefined until the MCU accesses an external space.

8.2.2 Chip-Select Output

Chip-select outputs share pins with address bus, A20 to A22 and $\overline{\text{A23}}$. Bits PM11 and PM10 in the PM1 register determine the $\overline{\text{CS}}$ areas to be accessed and the number of chip-select outputs. Maximum of four chip-select outputs are provided.

In microprocessor mode, no chip-select signal is output after reset. Only $\overline{\text{A23}}$, however, can perform as a chip-select output.

The $\overline{\text{CSi}}$ pin (i = 0 to 3) outputs an “L” signal while accessing its corresponding external space. An “H” signal is output while the MCU is accessing other external spaces. Figure 8.2 shows an example of address bus and chip-select outputs (separate bus).



Example 1:
After accessing the external space, both address bus and chip-select output change

When the MCU accesses the external space j specified by another chip-select output in the next cycle after having accessed the external space i, both address bus and chip-select output change.

Access external space i
Access another external space j
Data bus
Data
Data
Address bus
Address
Chip-select: $\overline{\text{CSk}}$
Chip-select: $\overline{\text{CSp}}$

Example 2:
After accessing an external space, the chip-select output changes but the address bus does not.

When the MCU accesses SFR or internal ROM/RAM area in the next cycle after having accessed an external space, the chip-select signal changes but the address bus does not.

Access external space
Access SFR, internal ROM/RAM
Data bus
Data
Address bus
Address
Chip-select: $\overline{\text{CSk}}$

Example 3:
After accessing the external space, the address bus changes but the chip-select output does not.

When the MCU accesses the space i specified by the same chip-select output in the next cycle after having accessed the external space i, the address bus changes but the chip-select output does not.

Access external space i
Access the same external space i
Data bus
Data
Data
Address bus
Address
Chip-select: $\overline{\text{CSk}}$

Example 4:
After accessing an external space, neither address bus nor chip-select signal changes.

When the MCU does not access any spaces in the next cycle after having accessed an external space (no instruction prefetch is performed), neither address bus nor chip-select signal changes.

Access external space
No accesss to external space
Data bus
Data
Address bus
Address
Chip-select: $\overline{\text{CSk}}$

i = 0 to 3
j = 0 to 3, excluding i
k = 0 to 3
p = 0 to 3, excluding k

NOTE:
1. The above examples show the address bus and chip-select output in two consecutive bus cycles. Depending on the combination, the chip-select signal can be more than two bus cycles.

$\overline{\text{CS1}}$ outputs an "L" signal while accessing the external space 0.
$\overline{\text{CS2}}$ outputs an "L" signal while accessing the external space 1.
$\overline{\text{CS3}}$ outputs an "L" signal while accessing the external space 2.
$\overline{\text{CS0}}$ outputs an "L" signal while accessing the external space 3.

Figure 8.2 Address Bus and Chip-Select Outputs (Separate Bus)

8.2.3 Read/Write Output Signals

When using a 16-bit data bus, the PM02 bit in the PM0 register selects either a combination of the “$\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$” outputs or the “$\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$” outputs to determine the read/write output signals. When bits DS3 to DS0 in the DS register are set to 0 (8-bit external data bus width), set the PM02 bit to 0 ($\overline{\text{RD}}$/$\overline{\text{WR}}$/$\overline{\text{BHE}}$). When any of bits DS3 to DS0 is set to 1 (16-bit external data bus width) to access an 8-bit space, the combination of “$\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$” is automatically selected regardless of the PM02 bit setting. Table 8.4 lists $\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$ outputs. Table 8.5 list $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$ outputs.

The $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$ outputs are selected for the read/write output signals after reset. When changing to “$\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$” outputs, set the PM02 bit first to write data to an external memory.

Table 8.4 $\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$ Outputs

Data Bus Width	$\overline{\text{RD}}$	$\overline{\text{WRL}}$	$\overline{\text{WRH}}$	A0	CPU Processing on External Space
16 bits	L	H	H	Not used	Read data
	H	L	H	Not used	Write 1-byte data to even address
	H	H	L	Not used	Write 1-byte data to odd address
	H	L	L	Not used	Write data to both even and odd addresses
8 bits	H	L(1)	Not used	H/L	Write 1-byte data
	L	H(1)	Not used	H/L	Read 1-byte data

NOTE:
1. These become $\overline{\text{WR}}$ output.

Table 8.5 $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$ Outputs

Data Bus Width	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{BHE}}$	A0	CPU Processing on External Space
16 bits	H	L	L	H	Write 1-byte data to odd address
	L	H	L	H	Read 1-byte data from odd address
	H	L	H	L	Write 1-byte data to even address
	L	H	H	L	Read 1-byte data from 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	H	L	Not used	H/L	Write 1-byte data
	L	H	Not used	H/L	Read 1-byte data

8.2.4 Bus Timing

Software wait states for the internal ROM and internal RAM can be set using the PM12 bit in the PM1 register, for the SFR area using the PM13 bit, and for external spaces using the EWCRi register (i = 0 to 3). Table 8.6 lists a software wait state and bus cycle.

The basic bus cycle for the internal ROM, internal RAM, and SFR area is one bus clock (BCLK) cycle. A read from the internal ROM takes the basic bus cycle. However, when the PLL clock is the source for the CPU clock, and also the FMR40 bit in the FMR4 register is set to 0 (normal-speed access), a read for the block A and block B in the user ROM area takes two BCLK cycles, regardless of the PM12 bit setting. A read or write to the internal RAM takes the basic bus cycle. When the PM12 bit in the PM1 register to 1 (1 wait state), an access to the internal ROM or internal RAM takes two BCLK cycles.

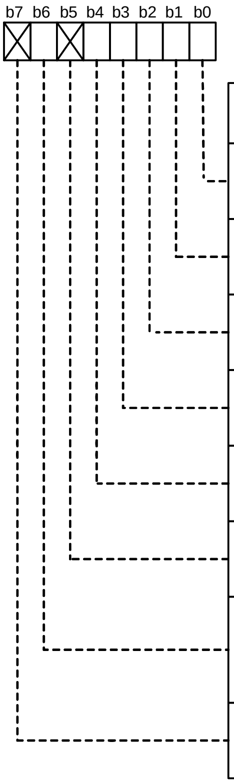
A read or write to the SFR area takes two BCLK cycles (1 wait state). When the PM13 bit in the PM1 register is set to 1 (2 wait states), an access takes three BCLK cycles.

The external bus cycle is divided into two phases: the number of BCLK cycles in the period from the beginning of the bus access until the read or write output signal becomes “L” (first ϕ), and the number of BCLK cycles in the period from the read or write output signal becomes “L” until the signal changes to “H” (second ϕ).

The minimum read or write cycle for the external bus is two BCLK cycles (1 ϕ + 1 ϕ). The EWCRi register (i = 0 to 3) selects an external bus cycle from 12 types for the separate bus and seven types for the multiplexed bus. For example, when bits EWCRi4 to EWCRi0 in the EWCRi register are set to 00011b (1 ϕ + 3 ϕ), the external bus cycle is four BCLK cycles.

Figure 8.3 shows the EWCRi register. Figures 8.4 to 8.8 show external bus timings.

External Space Wait Control Register i (i = 0 to 3)



Symbol	Address	After Reset
EWCR0 to EWCR3	0048h, 0049h, 004Ah, 004Bh	X0X0 0011b

Bit Symbol	Bit Name	Function	RW
EWCRi0	Bus cycle select bits$^{(3)}$	b4 b3 b2 b1 b0 (1) (2) 0 0 0 0 1: 1 ϕ + 1 ϕ 0 0 0 1 0: 1 ϕ + 2 ϕ 0 0 0 1 1: 1 ϕ + 3 ϕ 0 0 1 0 0: 1 ϕ + 4 ϕ 0 0 1 0 1: 1 ϕ + 5 ϕ 0 0 1 1 0: 1 ϕ + 6 ϕ 0 1 0 1 0: 2 ϕ + 2 ϕ 0 1 0 1 1: 2 ϕ + 3 ϕ 0 1 1 0 0: 2 ϕ + 4 ϕ 0 1 1 0 1: 2 ϕ + 5 ϕ 1 0 0 1 1: 3 ϕ + 3 ϕ 1 0 1 0 0: 3 ϕ + 4 ϕ 1 0 1 0 1: 3 ϕ + 5 ϕ 1 0 1 1 0: 3 ϕ + 6 ϕ Do not set to values other than the above	RW
EWCRi1			RW
EWCRi2			RW
EWCRi3			RW
EWCRi4			RW
– (b5)	Unimplemented. Write 0. Read as undefined value.		–
EWCRi6	Recovery cycle insert select bit	0: Insert no recovery cycle when accessing external space i 1: Insert a recovery cycle when accessing external space i	RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. The number of BCLK cycles in the period from the beginning of the bus access until the read or write output signal becomes "L".
2. The number of BCLK cycles in the period from the read or write output signal becomes "L" until the signal changes to "H".

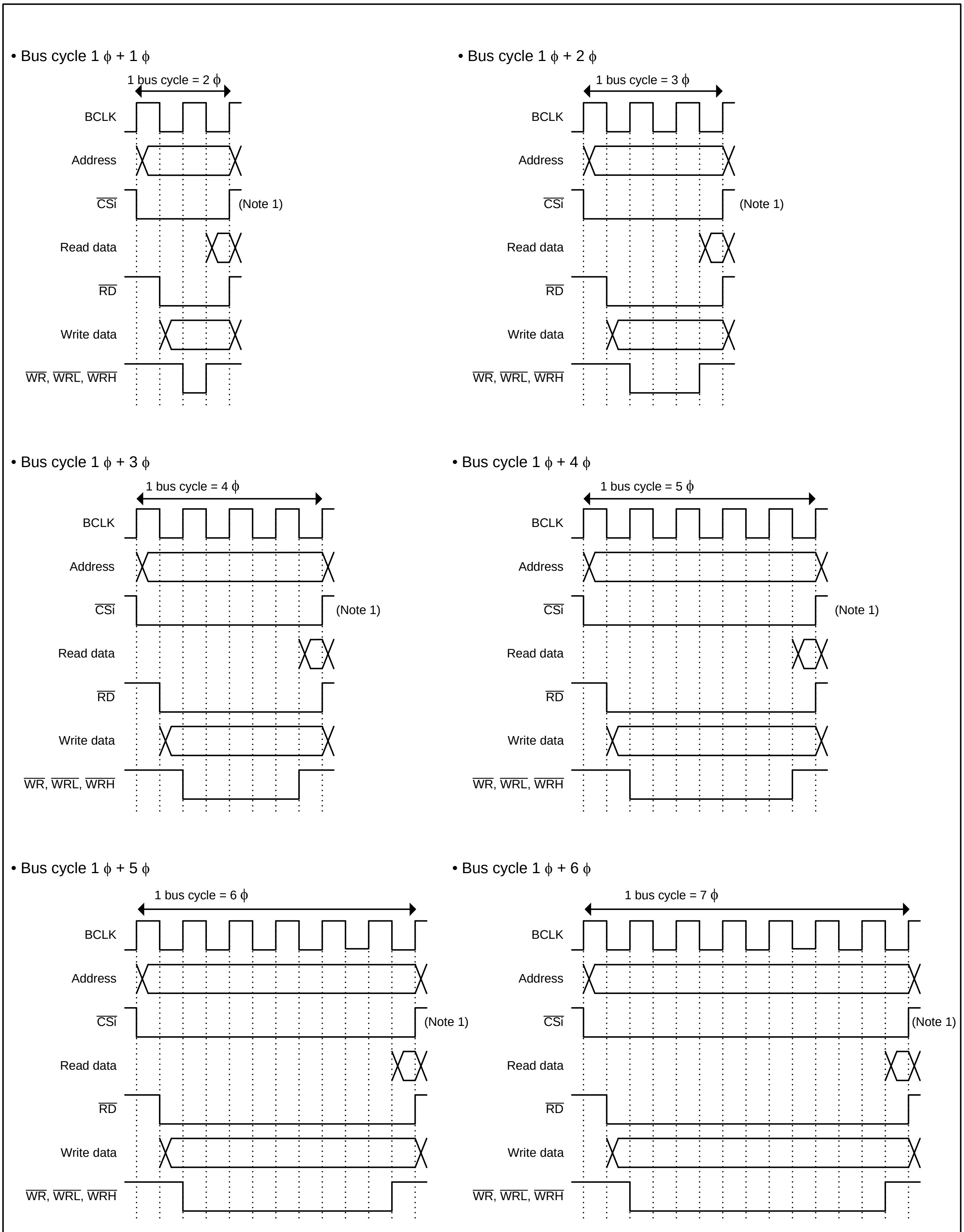
Figure 8.3 EWCR0 to EWCR3 Registers

Table 8.6 Software Wait State and Bus Cycle

Space	External Bus Status	PM1 Register		EWCRi Register (i=0 to 3)	Bus Cycle
		PM13 Bit	PM12 Bit	Bits EWCRi4 to EWCRi0	
SFR area	–	0	–	–	2 BCLK cycles
		1			3 BCLK cycles
Internal ROM[(1)]/ RAM	–	–	0	–	1 BCLK cycle
			1		2 BCLK cycles
External memory	Separate bus	–	–	00001b	2 BCLK cycles
				00010b	3 BCLK cycles
				00011b	4 BCLK cycles
				00100b	5 BCLK cycles
				00101b	6 BCLK cycles
				00110b	7 BCLK cycles
				01010b	4 BCLK cycles
				01011b	5 BCLK cycles
				01100b	6 BCLK cycles
				10011b	6 BCLK cycles
				10100b	7 BCLK cycles
				10110b	9 BCLK cycles
	Multiplexed bus	–	–	01010b	4 BCLK cycles
				01011b	5 BCLK cycles
				01101b	7 BCLK cycles
				10011b	6 BCLK cycles
				10100b	7 BCLK cycles
				10101b	8 BCLK cycles
				10110b	9 BCLK cycles

NOTE:

1. When the PLL clock is the source for the CPU clock, and also the FMR40 bit in the FMR4 register is set to 0 (normal-speed access), a read for the block A and block B in the user ROM area takes two BCLK cycles, regardless of the PM12 bit setting.

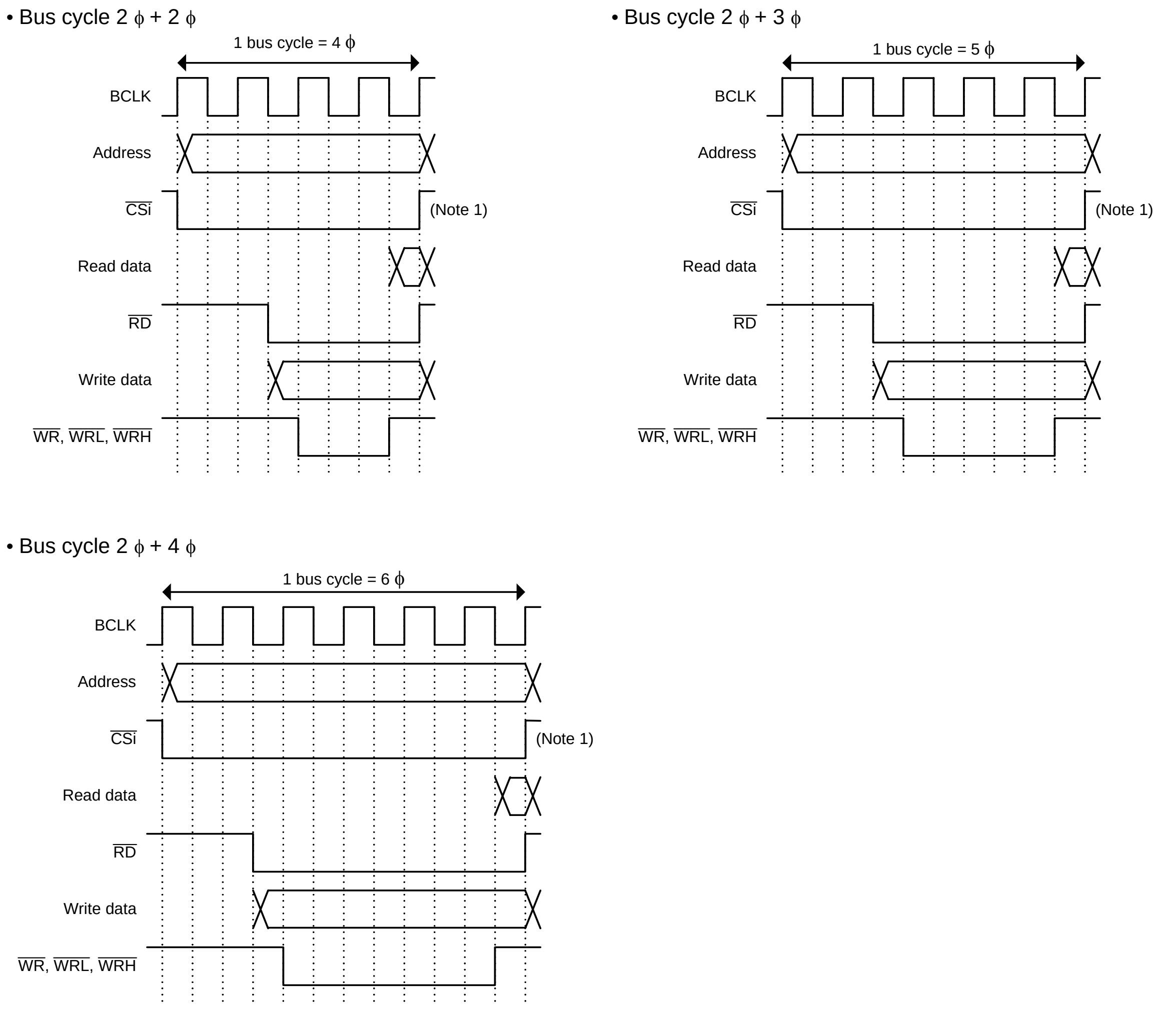


i = 0 to 3

NOTE:

1. When the MCU accesses the same $\overline{\text{CS}}$ area consecutively, the $\overline{\text{CSi}}$ pin keeps outputting "L".

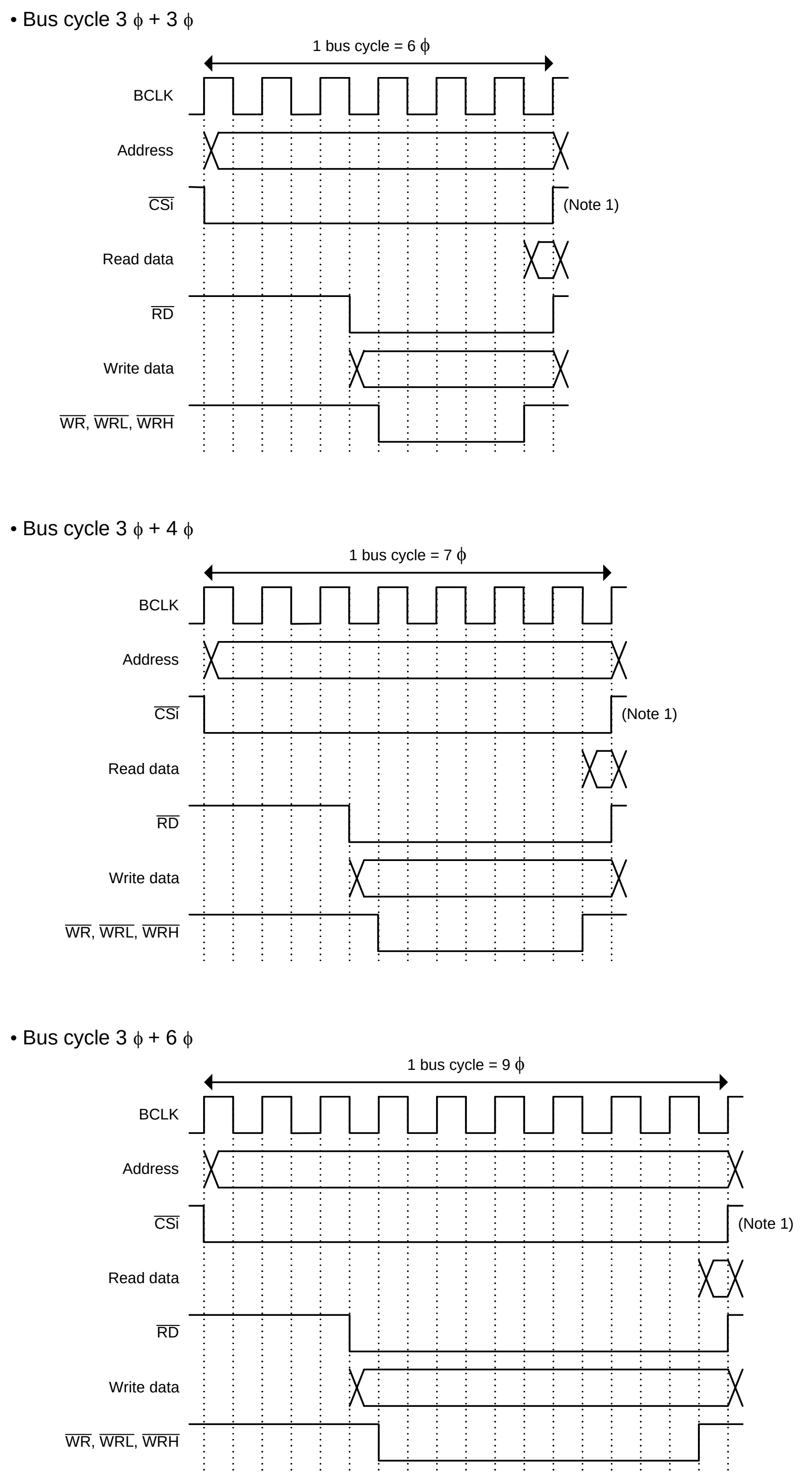
Figure 8.4 Bus Cycles when Separate Bus is Selected (1/3)



i = 0 to 3

NOTE:
1. When the MCU accesses the same $\overline{\text{CS}}$ area consecutively, the $\overline{\text{CSi}}$ pin keeps outputting "L".

Figure 8.5 Bus Cycles when Separate Bus is Selected (2/3)

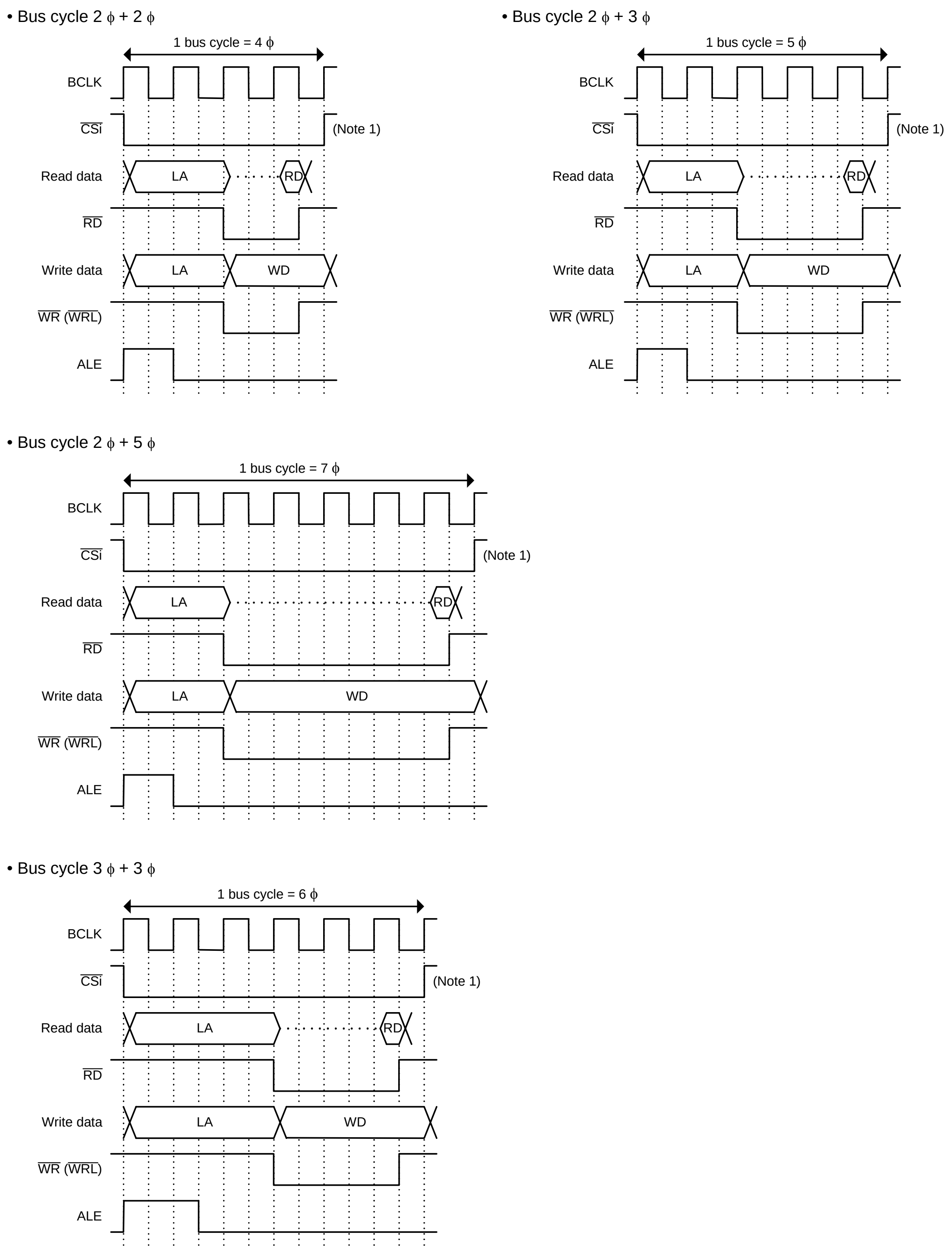


i = 0 to 3

NOTE:

1. When the MCU accesses the same $\overline{\text{CS}}$ area consecutively, the $\overline{\text{CSi}}$ pin keeps outputting "L".

Figure 8.6 Bus Cycle with Separate Bus is Selected (3/3)



• Bus cycle 2 ϕ + 2 ϕ

• Bus cycle 2 ϕ + 3 ϕ

• Bus cycle 2 ϕ + 5 ϕ

• Bus cycle 3 ϕ + 3 ϕ

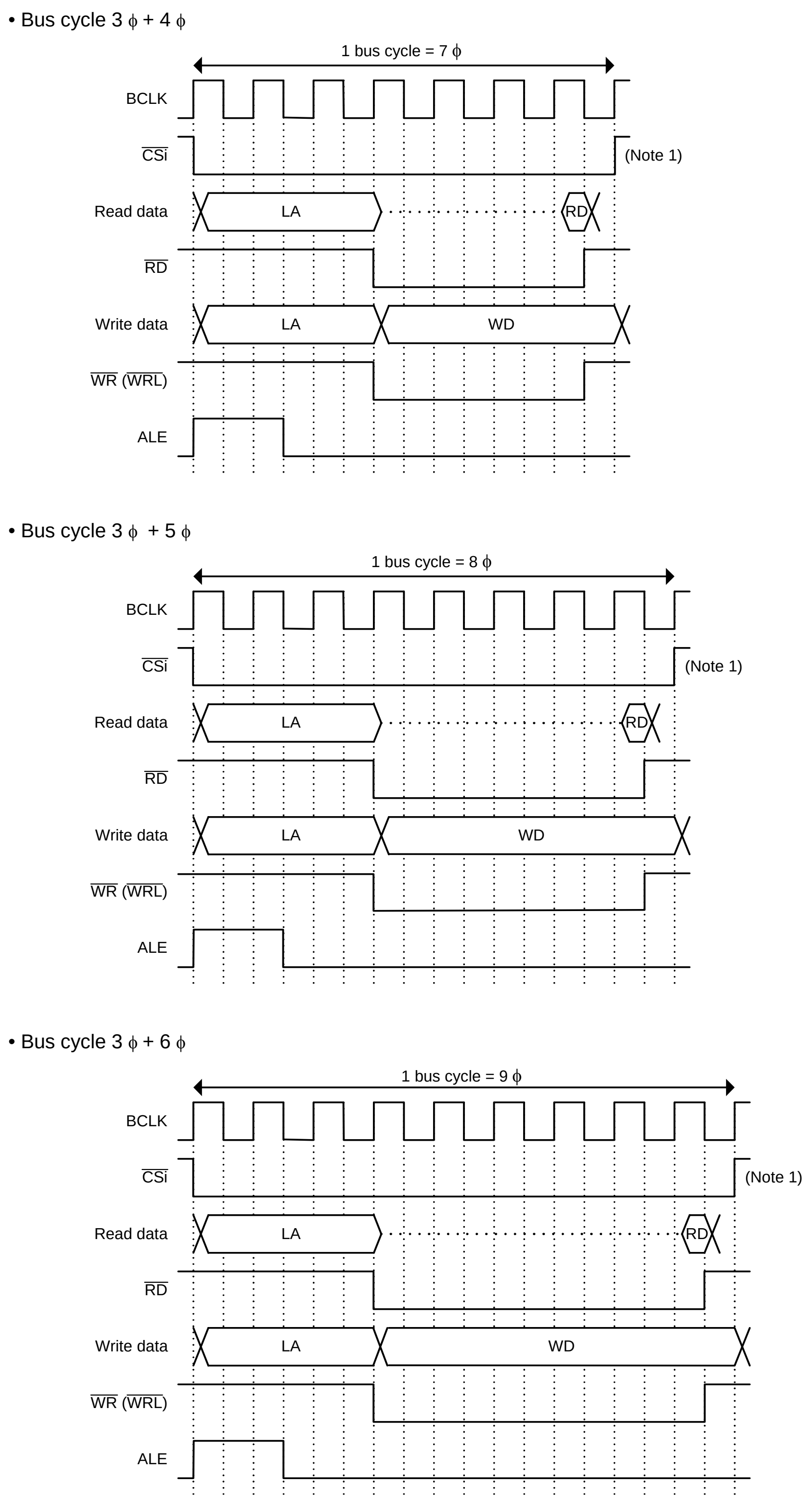
LA: Latch address RD: Read data WD: Write data

i=0 to 3

NOTE:

1. When the MCU accesses the same $\overline{\text{CS}}$ area consecutively, the $\overline{\text{CSi}}$ pin keeps outputting "L".

Figure 8.7 Bus Cycles when Multiplexed Bus is Selected (1/2)



• Bus cycle 3 ϕ + 4 ϕ

• Bus cycle 3 ϕ + 5 ϕ

• Bus cycle 3 ϕ + 6 ϕ

LA: Latch address RD: Read data WD: Write data

i = 0 to 3

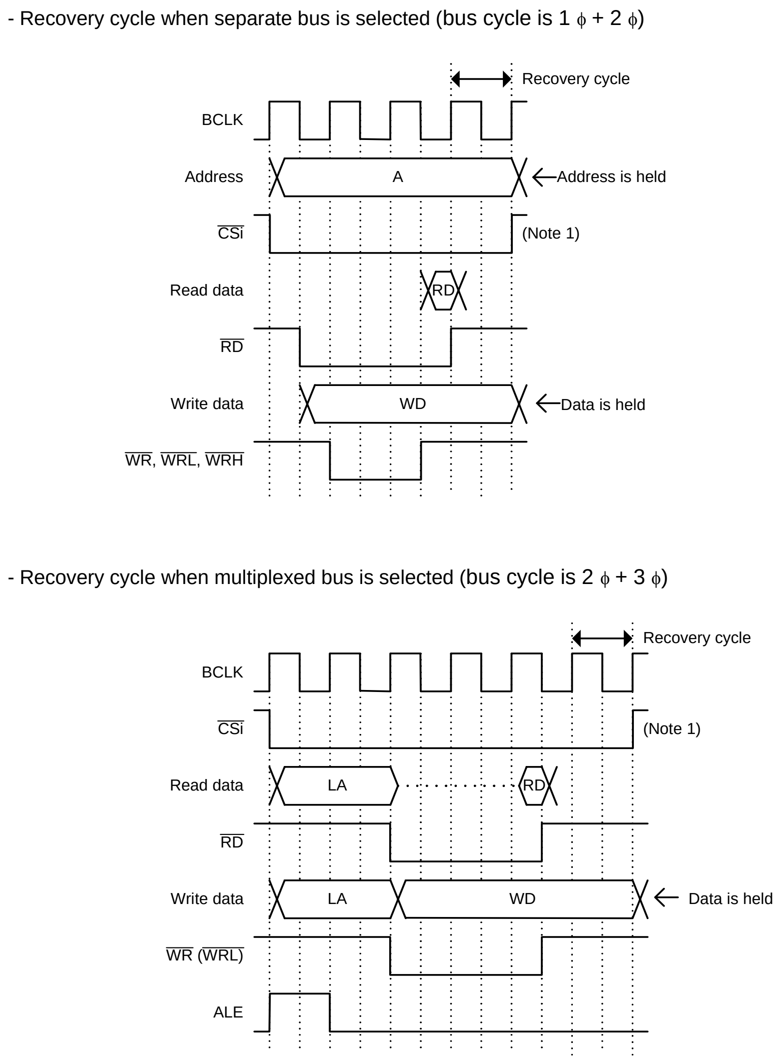
NOTE:

1. When the MCU accesses the same $\overline{\text{CS}}$ area consecutively, the $\overline{\text{CSi}}$ pin keeps outputting "L".

Figure 8.8 Bus Cycles when Multiplexed Bus is Selected (2/2)

8.2.4.1 Bus Cycle with Recovery Cycle Inserted

The EWCRi6 bit in the EWCRi register (i = 0 to 3) determines whether the recovery cycle is inserted or not. Address output or data output is held during the recovery cycle (only when using the separate bus). Devices, which require longer address hold time or data hold time, are connectable.



- Recovery cycle when separate bus is selected (bus cycle is 1 ϕ + 2 ϕ)

- Recovery cycle when multiplexed bus is selected (bus cycle is 2 ϕ + 3 ϕ)

A: address LA: Latch address RD: Read data WD: Write data

i = 0 to 3

NOTE:

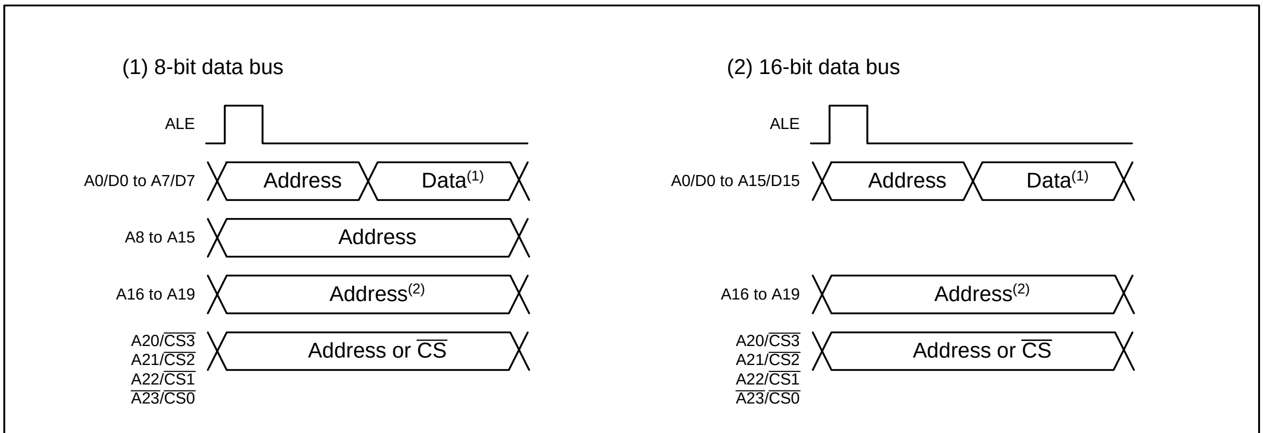
1. When the MCU accesses the same CS area consecutively, the CSi pin keeps outputting "L".

Figure 8.9 Recovery Cycle

8.2.5 ALE Output

The ALE output signal is provided for the external devices to latch the address when using the multiplexed bus. Latch the address at the falling edge of the ALE output. Bits PM15 and PM14 in the PM1 register determine to what pin the ALE output is assigned.

The ALE signal is output even when accessing the internal space.



NOTES:
1. A0/D0 to A15/D15 are placed in high-impedance states when read.
2. When the multiplexed bus is selected for all CS areas, A16 to A19 become I/O ports.

Figure 8.10 ALE Output and Address/Data Bus

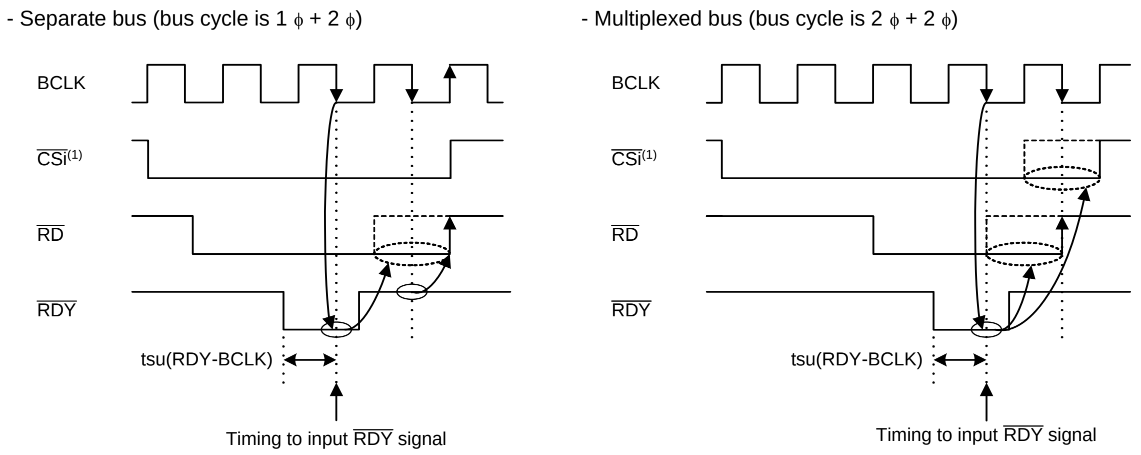
8.2.6 RDY Input

The RDY signal facilitates access to external devices requiring longer access time. When RDY input is “L” at the falling edge of the last BCLK cycle, wait states are inserted into the bus cycle. Then, when an “H” signal is input to the RDY pin at the falling edge of BCLK, the MCU resumes executing the remaining bus clock.

Table 8.7 lists MCU states when placed in wait state by RDY input. Figure 8.11 shows an example of the RD signal that is extended by the RDY signal.

Table 8.7 MCU States while “L” is Input to the RDY Pin

Item	State
Clock generation circuits	Operating (oscillating)
RD, WR, A0 to A22, A23, D0 to D15, CS0 to CS3, ALE, HLDA, programmable I/O ports	Maintains the same state as when “L” is input to RDY pin.
Internal peripheral circuits	Operating


- Separate bus (bus cycle is 1 ϕ + 2 ϕ)
- Multiplexed bus (bus cycle is 2 ϕ + 2 ϕ)
BCLK
$\overline{CSi}$(1)
$\overline{RD}$
$\overline{RDY}$
tsu(RDY-BCLK)
Timing to input $\overline{RDY}$ signal

i = 0 to 3

◌ : Wait states inserted by $\overline{RDY}$ input

tsu(RDY-BCLK): $\overline{RDY}$ input setup time

NOTE:

1. Chip-select output ($\overline{CSi}$) may be extended depending on the CPU state such as the instruction queue buffer.

Figure 8.11 $\overline{RD}$ Output Signal Extended by $\overline{RDY}$ Input

8.2.7 $\overline{HOLD}$ Input

The $\overline{HOLD}$ input signal is used to transfer ownership of the bus from the CPU to external devices. When a low-level ("L") signal is applied to the $\overline{HOLD}$ pin, the MCU enters a hold state after the bus access in progress is completed. While the $\overline{HOLD}$ pin is held "L", the MCU remains in a hold state and the $\overline{HLDA}$ pin outputs an "L" signal. Table 8.8 lists the MCU states in hold state.

Bus is used in the following priority order: $\overline{HOLD}$, DMAC, CPU.

Table 8.8 MCU States in Hold State

Item	State
Clock generation circuits	Operating (oscillating)
CPU	Stopped
Internal peripheral circuits	Operating (Watchdog timer is stopped)(1)
$\overline{RD}$, $\overline{WR}$, A0 to A22, $\overline{A23}$, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$	High-impedance
$\overline{HLDA}$	Outputs "L"
ALE	Outputs "L"
Programmable I/O ports	Maintains the same state as when "L" is input to $\overline{HOLD}$ pin.

NOTE:

1. When the PM22 bit in the PM2 register is set to 1 (selects the on-chip oscillator clock as count source for the watchdog timer), watchdog timer does not stop.

8.2.8 External Bus States when Accessing Internal Space

Table 8.9 lists external bus states when the internal space is accessed.

Table 8.9 External Bus States when Accessing Internal Space

Item	State when Accessing SFR, Internal ROM, and Internal RAM
A0 to A22, $\overline{\text{A23}}$	Hold the last accessed address in the external space
D0 to D15	High-impedance
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$	Outputs “H”
$\overline{\text{BHE}}$	Holds the output level at the time when the MCU accessed the external space or SFR area for the last time
$\overline{\text{CS}}$	Outputs “H”
ALE	Outputs ALE signal

8.2.9 BCLK Output

The bus clock can be output from the BCLK pin in memory expansion mode and microprocessor mode. To output the bus clock, set the PM07 bit in the PM0 register to 0 (BCLK output) and bits CM01 and CM00 in the CM0 register to 00b (I/O port P5_3). No BCLK is output in single-chip mode.
Refer to 9. Clock Generation Circuits for details.

8.3 Page Mode Control Function

NOTE

The page mode control function is available only in the ROMless version.

The page mode control function allows high-speed read access to the external memory compatible with the page mode control. While the MCU accesses data within the eight-byte block of consecutive addresses which have the same 21 high-order bits, less cycles are taken for the subsequent bus read accesses to a maximum of seven-byte addresses than the first bus access.

The EWCRi register (i = 0 to 3) determines how many wait states are inserted for the first bus access. Registers PWCR0 and PWCR1 determine how many wait states are inserted for the subsequent bus accesses. Use the following procedure to enable the page mode control.

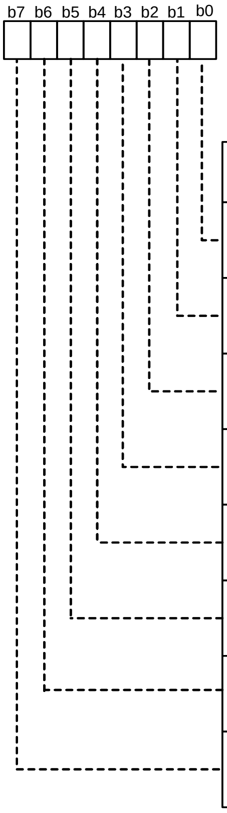
(1) Set bits EWCRi4 to EWCRi0 in the EWCRi register.
(2) Set bits PWCRj02 to PWCRj00 and bits PWCRj06 to PWCRj04 in the PWCRj register (j = 0, 1).
(3) Set bits PWCRj03 and PWCRj07 to 1 (page mode control enabled).

When using the page mode control function, access all the external spaces using page mode control. It is not allowed to combine the page mode controlled access and the normal access to external spaces.

Set bits PM05 and PM04 to 00b (multiplexed bus is not used). The page mode control function and multiplexed bus cannot be used at the same time.

Figure 8.12 and 8.13 show registers PWCR0 and PWCR1. Figure 8.14 shows a diagram of external bus timing with page mode function.

Page Mode Wait Control Register 0



Symbol	Address	After Reset
PWCR0	004Ch	0001 0001b

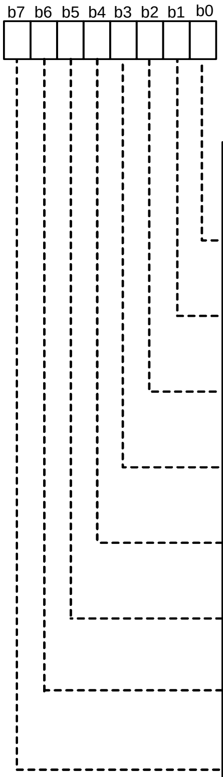
Bit Symbol	Bit Name	Function	RW
PWCR000	External space 0 subsequent access wait select bits	b2 b1 b0 0 0 1: 1 ϕ + 1 ϕ 0 1 0: 1 ϕ + 2 ϕ 0 1 1: 1 ϕ + 3 ϕ 1 0 0: 1 ϕ + 4 ϕ Do not set to values other than the above.	RW
PWCR001			RW
PWCR002			RW
PWCR003	External space 0 page mode control enable bit	0: Page mode control disabled 1: Page mode control enabled(1)	RW
PWCR004	External space 1 subsequent access wait select bits	b6 b5 b4 0 0 1: 1 ϕ + 1 ϕ 0 1 0: 1 ϕ + 2 ϕ 0 1 1: 1 ϕ + 3 ϕ 1 0 0: 1 ϕ + 4 ϕ Do not set to values other than the above.	RW
PWCR005			RW
PWCR006			RW
PWCR007	External space 1 page mode control enable bit	0: Page mode control disabled 1: Page mode control enabled(1)	RW

NOTE:

1. When page mode control is enabled, set the EWCRi6 bit in the EWCRi register (i = 0 to 3) to 0 (add no recovery cycle when accessing external space i).

Figure 8.12 PWCR0 Register

Page Mode Wait Control Register 1

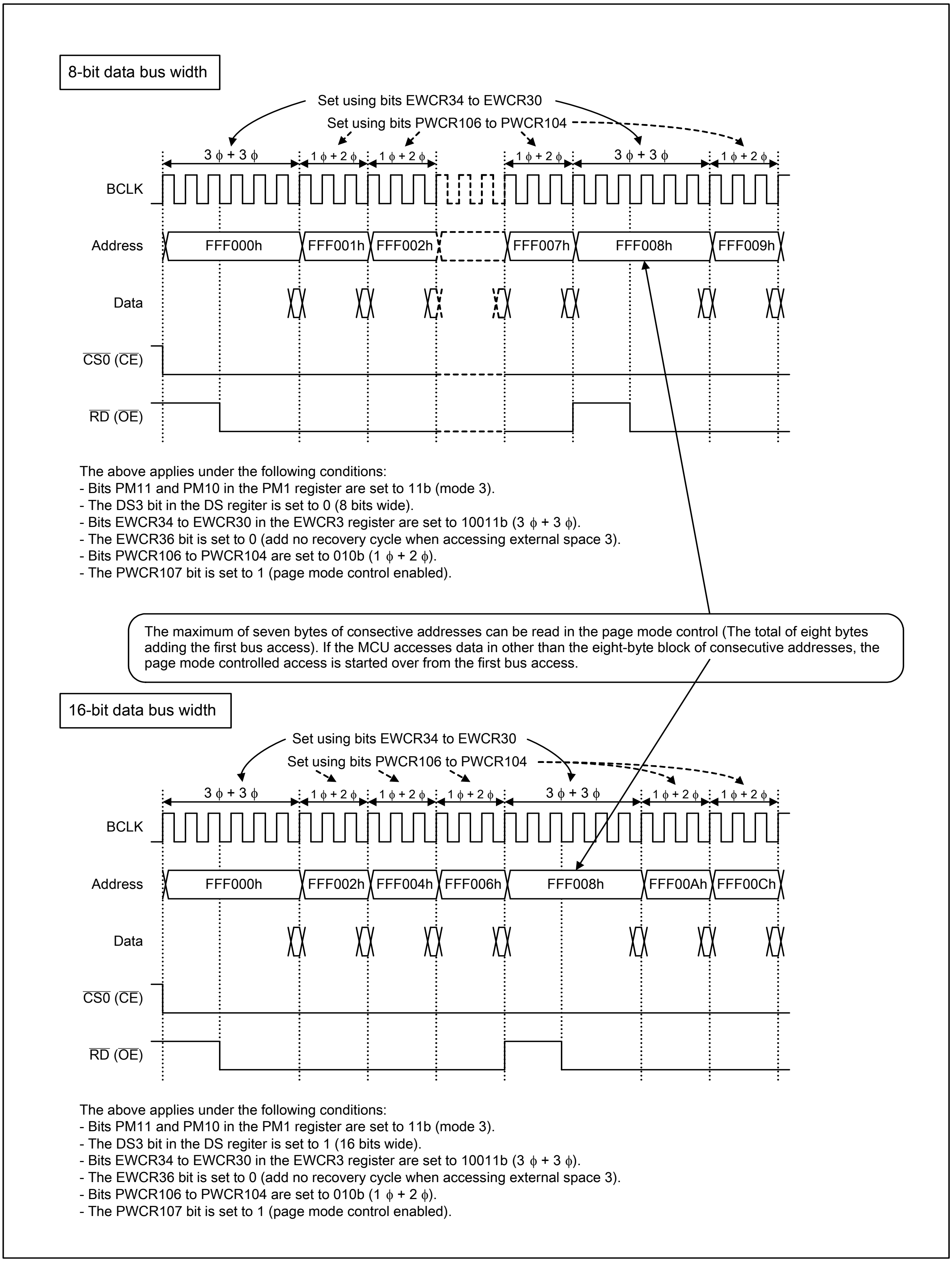


Symbol	Address	After Reset
PWCR1	004Dh	0001 0001b

Bit Symbol	Bit Name	Function	RW
PWCR100	External space 2 subsequent access wait select bits	b2 b1 b0 0 0 1: 1 ϕ + 1 ϕ 0 1 0: 1 ϕ + 2 ϕ 0 1 1: 1 ϕ + 3 ϕ 1 0 0: 1 ϕ + 4 ϕ Do not set to values other than the above.	RW
PWCR101			RW
PWCR102			RW
PWCR103	External space 2 page mode control enable bit	0: Page mode control disabled 1: Page mode control enabled(1)	RW
PWCR104	External space 3 subsequent access wait select bits	b6 b5 b4 0 0 1: 1 ϕ + 1 ϕ 0 1 0: 1 ϕ + 2 ϕ 0 1 1: 1 ϕ + 3 ϕ 1 0 0: 1 ϕ + 4 ϕ Do not set to values other than the above.	RW
PWCR105			RW
PWCR106			RW
PWCR107	External space 3 page mode control enable bit	0: Page mode control disabled 1: Page mode control enabled(1)	RW

NOTE:
1. When page mode control is enabled, set the EWCRi6 bit in the EWCRi register (i = 0 to 3) to 0 (add no recovery cycle when accessing external space i).

Figure 8.13 PWCR1 Register



The above applies under the following conditions:
- Bits PM11 and PM10 in the PM1 register are set to 11b (mode 3).
- The DS3 bit in the DS regiter is set to 0 (8 bits wide).
- Bits EWCR34 to EWCR30 in the EWCR3 register are set to 10011b (3 ϕ + 3 ϕ).
- The EWCR36 bit is set to 0 (add no recovery cycle when accessing external space 3).
- Bits PWCR106 to PWCR104 are set to 010b (1 ϕ + 2 ϕ).
- The PWCR107 bit is set to 1 (page mode control enabled).

The maximum of seven bytes of consective addresses can be read in the page mode control (The total of eight bytes adding the first bus access). If the MCU accesses data in other than the eight-byte block of consecutive addresses, the page mode controlled access is started over from the first bus access.

The above applies under the following conditions:
- Bits PM11 and PM10 in the PM1 register are set to 11b (mode 3).
- The DS3 bit in the DS regiter is set to 1 (16 bits wide).
- Bits EWCR34 to EWCR30 in the EWCR3 register are set to 10011b (3 ϕ + 3 ϕ).
- The EWCR36 bit is set to 0 (add no recovery cycle when accessing external space 3).
- Bits PWCR106 to PWCR104 are set to 010b (1 ϕ + 2 ϕ).
- The PWCR107 bit is set to 1 (page mode control enabled).

Figure 8.14 External Bus Timing with Page Mode Control Function

9. Clock Generation Circuits

9.1 Types of the Clock Generation Circuit

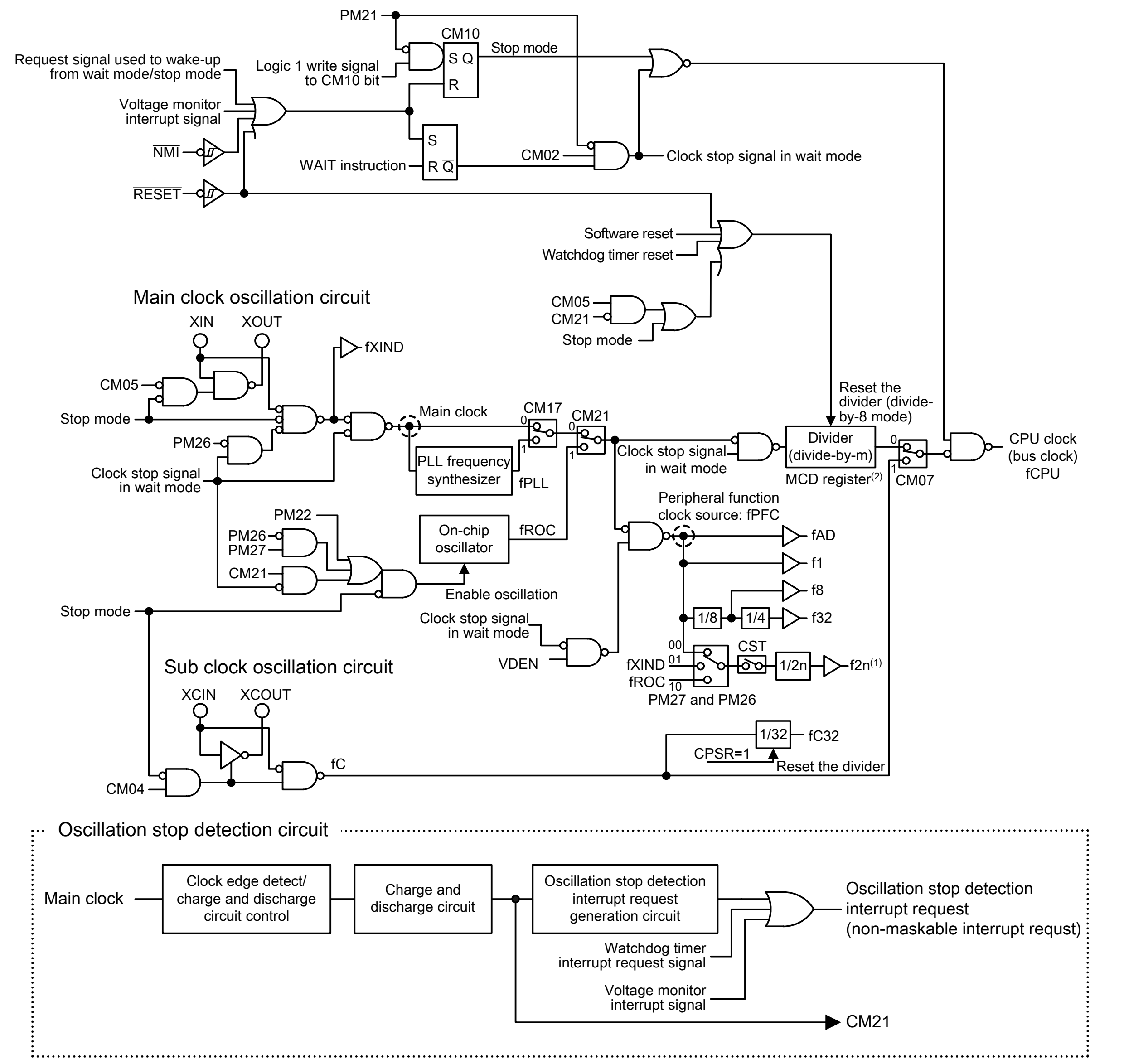
The MCU has four on-chip clock generation circuits to generate system clock signals.

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

Table 9.1 lists the specifications of the clock generation circuit. Figure 9.1 shows a block diagram of the clock generation circuit. Figures 9.2 to 9.8 show clock-associated registers.

Table 9.1 Clock Generation Circuit Specifications

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit	On-chip Oscillator	PLL Frequency Synthesizer
Applications	• CPU clock source • Peripheral function clock source	• CPU clock source • Count source for timer A and timer B	• CPU clock source • Peripheral function clock source	• CPU clock source • Peripheral function clock source
Clock frequency	Up to 16 MHz	32.768 kHz	Approx. 1 MHz	10 MHz to 32 MHz (see **Table 9.3**)
Connectable oscillator or resonator	• Ceramic resonator • Crystal oscillator	Crystal oscillator	–	–
Oscillator or resonator connect pins	XIN, XOUT	XCIN, XCOUT	–	–
Oscillation stop/ restart function	Available	Available	Available	Available
Oscillator state after reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally generated clock can be used.	Externally generated clock can be used.	Oscillation stop detect function: When the main clock stops, the on-chip oscillator starts oscillating automatically and becomes the CPU and peripheral function clock source	20 MHz: Input 5 MHz or 10 MHz to the main clock 32 MHz Input 8 MHz or 16 MHz to the main clock

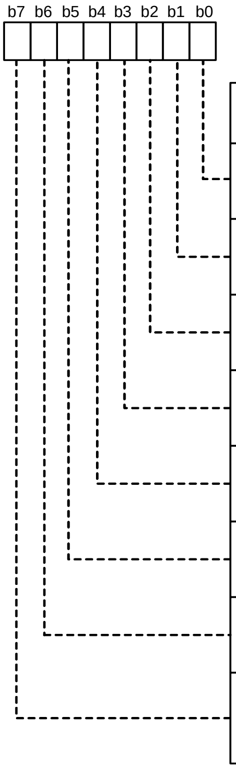

PM21
CM10
S Q
R
Stop mode
Request signal used to wake-up from wait mode/stop mode
Logic 1 write signal to CM10 bit
Voltage monitor interrupt signal
NMI
S
R Q
WAIT instruction
CM02
Clock stop signal in wait mode
RESET
Software reset
Watchdog timer reset
CM05
CM21
Stop mode
Main clock oscillation circuit
XIN
XOUT
fXIND
CM05
Stop mode
PM26
Clock stop signal in wait mode
Main clock
CM17
CM21
PLL frequency synthesizer
fPLL
Clock stop signal in wait mode
Reset the divider (divide-by-8 mode)
Divider (divide-by-m)
MCD register(2)
CM07
CPU clock (bus clock) fCPU
Peripheral function clock source: fPFC
fAD
f1
f8
1/8
1/4
f32
PM22
PM26
PM27
CM21
On-chip oscillator
fROC
Enable oscillation
Stop mode
Clock stop signal in wait mode
VDEN
fXIND
fROC
00
01
10
CST
1/2n
f2n(1)
PM27 and PM26
Sub clock oscillation circuit
XCIN
XCOUT
fC
CM04
1/32
fC32
CPSR=1
Reset the divider
Oscillation stop detection circuit
Main clock
Clock edge detect/ charge and discharge circuit control
Charge and discharge circuit
Oscillation stop detection interrupt request generation circuit
Watchdog timer interrupt request signal
Voltage monitor interrupt signal
Oscillation stop detection interrupt request (non-maskable interrupt requst)
CM21

VDEN: bit in the LVDC register
VC27: bit in the VCR2 register
CM02, CM04, CM05, and CM07: bits in the CM0 register
CM10 and CM17: bits in the CM1 register
CM21: bit in the CM2 regsiter
PM21, PM22, PM26, and PM27: bits in the PM2 register
CST: bit in the TCSPR register
CPSR: bit in the CPSRF register
NOTES:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Bits MCD4 to MCD0 in the MCD register select the dividing ratio (divide-by-m mode: m = 1, 2, 3, 4, 6, 8, 10, 12, 14, 16).

Figure 9.1 Clock Generation Circuit

System Clock Control Register 0[1]



Symbol	Address	After Reset
CM0	0006h	0000 1000b

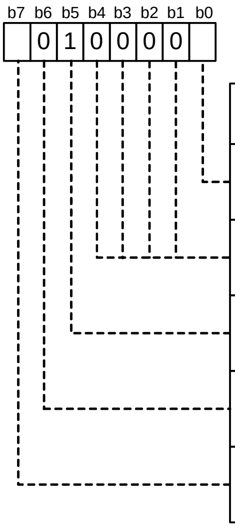
Bit Symbol	Bit Name	Function	RW
CM00	Clock output function select bits[2]	b1 b0 0 0: I/O port P5_3[2] 0 1: Outputs fC 1 0: Outputs f8 1 1: Outputs f32	RW
CM01			RW
CM02	Peripheral function clock stop in wait mode bit[9]	0: Peripheral clocks do not stop in wait mode 1: Peripheral clocks stop in wait mode[3]	RW
CM03	XCIN-XCOUT drive capability select bit[10]	0: Low 1: High	RW
CM04	Port XC switch bit	0: I/O port function 1: XCIN-XCOUT oscillation function[4]	RW
CM05	Main clock (XIN-XOUT) stop bit[5, 9]	0: Main clock oscillates 1: Main clock stops[6]	RW
CM06	Watchdog timer function select bit	0: Watchdog timer interrupt 1: Reset[7]	RW
CM07	CPU clock select bit 0[8, 9]	0: Clock selected by the CM21 bit divided by the MCD register 1: Sub clock	RW

NOTES:

1. Set the CM0 register after the PRC0 bit in the PRCR register is set to 1 (write enable).
2. The BCLK, ALE, or "L" signal is output from the P5_3 pin in memory expansion mode or microprocessor mode. The P5_3 does not function as an I/O port.
3. fC32 does not stop running.
4. To set the CM04 bit to 1, set bits PD8_7 and PD8_6 in the PD8 register to 00b (ports P8_6 and P8_7 in input mode) and the PU25 bit in the PUR2 register to 0 (not pulled up).
5. The CM05 bit stops the main clock oscillation when entering low-power consumption mode or on-chip oscillator low-power consumption mode. The CM05 bit cannot be used to determine whether the main clock stops or not. To stop the main clock oscillation, set the PLC07 bit in the PLC0 register to 0 and the CM05 bit to 1 after setting the CM07 bit to 1 or setting the CM21 bit in the CM2 register to 1 (on-chip oscillator clock).
   When the CM05 bit is set to 1, the XOUT pin outputs "H". Since an on-chip feedback resistor remains ON, the XIN pin is pulled up to the XOUT pin via the feedback resistor.
6. When the CM05 bit is set to 1, bits MCD4 to MCD0 in the MCD register become 01000b (divide-by-8 mode). In on-chip oscillator mode, bits MCD4 to MCD0 do not become 01000b even if the CM05 bit is set to 1.
7. Once the CM06 bit is set to 1, it cannot be set to 0 by a program.
8. Change the CM07 bit setting from 0 to 1, after the CM04 bit is set to 1 and the sub clock oscillation stabilizes.
   Change the CM07 bit setting from 1 to 0, after the CM05 bit is set to 0 and the main clock oscillation stabilizes.
   Do not change the CM07 bit simultaneously with the CM04 or CM05 bit.
9. If the PM21 bit in the PM2 register is set to 1 (disables a clock change), a write to bits CM02, CM05, and CM07 has no effect.
10. When stop mode is entered, the CM03 bit becomes 1.

Figure 9.2 CM0 Register

System Clock Control Register 1[1]



Symbol	Address	After Reset
CM1	0007h	0010 0000b

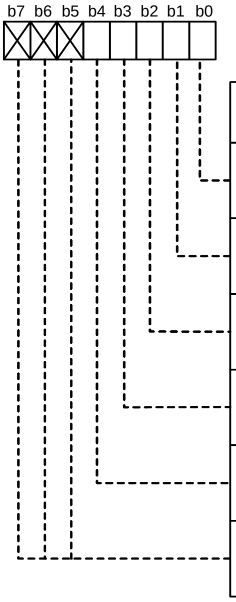
Bit Symbol	Bit Name	Function	RW
CM10	All clock stop control bit[2, 3, 5]	0: Clock oscillates 1: All clocks stop (stop mode)	RW
– (b4-b1)	Reserved bits	Set to 0	RW
– (b5)	Reserved bit	Set to 1	RW
– (b6)	Reserved bit	Set to 0	RW
CM17	CPU clock select bit 1[4, 5]	0: Main clock 1: PLL clock	RW

NOTES:

1. Set the CM1 register after the PRC0 bit in the PRCR register is set to 1 (write enable).
2. When the CM10 bit is set to 1, the XOUT pin outputs "H" and the on-chip feedback resistor is disconnected. Pins XIN, XCIN, and XCOUT are placed in high-impedance states.
3. When the CM10 bit is set to 1, bits MCD4 to MCD0 in the MCD register become 01000b (divide-by-8 mode). Do not set the CM10 bit to 1, when the CM20 bit in the CM2 register is set to 1 (oscillation stop detect function used) or the CM21 bit in the CM2 register is set to 1 (on-chip oscillator clock).
4. Set the CM17 bit to 1 after the PLL clock oscillation stablilizes.
5. If the PM21 bit in the PM2 register is set to 1 (disables a clock change), writes to bits CM10 and CM17 have no effect. If the PM22 bit in the PM2 register is set to 1 (on-chip oscillator clock as count source for watchdog timer), a write to the CM10 bit has no effect.

Figure 9.3 CM1 Register

Main Clock Division Register(1)



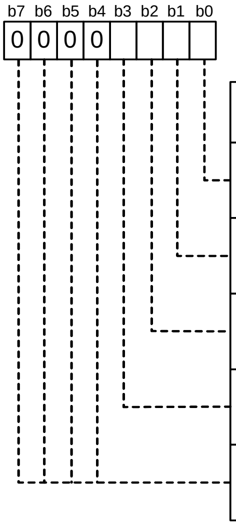
Symbol	Address	After Reset
MCD	000Ch	XXX0 1000b

Bit Symbol	Bit Name	Function	RW
MCD0	Main clock division rate select bits(2)	b4 b3 b2 b1 b0 1 0 0 1 0: Divide-by-1 (no division) mode 0 0 0 1 0: Divide-by-2 mode 0 0 0 1 1: Divide-by-3 mode 0 0 1 0 0: Divide-by-4 mode 0 0 1 1 0: Divide-by-6 mode 0 1 0 0 0: Divide-by-8 mode 0 1 0 1 0: Divide-by-10 mode 0 1 1 0 0: Divide-by-12 mode 0 1 1 1 0: Divide-by-14 mode 0 0 0 0 0: Divide-by-16 mode Do not set to values other than the above	RW
MCD1			RW
MCD2			RW
MCD3			RW
MCD4			RW
– (b7-b5)	Reserved bits	Read as undefined value	–

NOTES:
1. Set the MCD register after the PRC0 bit in the PRCR register is set to 1 (write enable).
2. When stop mode or low-power consumption mode is entered, bits MCD4 to MCD0 become 01000b. Bits MCD4 to MCD0, however, do not become 01000b if the CM05 bit in the CM0 register is set to 1 (main clock stops) while the CM21 bit in the CM2 register is set to 1.

Figure 9.4 MCD Register

Oscillation Stop Detection Register[1]



Symbol	Address	After Reset
CM2	000Dh	00h

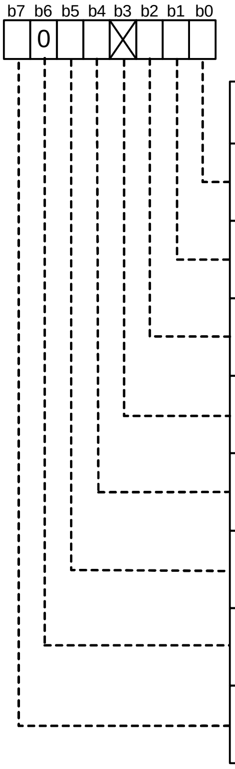
Bit Symbol	Bit Name	Function	RW
CM20	Oscillation stop detection enable bit[2]	0: Oscillation stop detect function not used 1: Oscillation stop detect function used	RW
CM21	CPU clock select bit 2[3, 4]	0: Clock selected by the CM17 bit 1: On-chip oscillator clock	RW
CM22	Oscillation stop detection flag[5]	0: Loss of main clock not detected 1: Loss of main clock detected	RW
CM23	Main clock monitor flag[6]	0: Main clock oscillates 1: Main clock stops	RO
– (b7-b4)	Reserved bits	Set to 0	RW

NOTES:

1. Set the CM2 register after the PRC0 bit in the PRCR register is set to 1 (write enable).
2. If the PM21 bit in the PM2 register is set to 1 (disables a clock change), a write to the CM20 bit has no effect.
3. When a loss of the main clock is detected while the CM20 bit is set to 1, the CM21 bit becomes 1. Although the main clock restarts oscillating, the CM21 bit does not become 0. To use the main clock as the CPU clock source after the main clock restarts oscillating, set the CM21 bit to 0 by a program.
4. When both the CM20 and CM23 bits are 1, do not set the CM21 bit to 0.
5. When a loss of the main clock is detected, the CM22 bit becomes 1. The CM22 bit can only be set to 0, not 1, by a program. If the CM22 bit is set to 0 by a program while the main clock is stopped, the CM22 bit does not become 1 until another loss of the main clock is detected after the main clock restarts oscillating.
6. Determine the main clock state by reading the CM23 bit several times after the oscillation stop detection interrupt is generated.

Figure 9.5 CM2 Register

PLL Control Register 0 [(1)(2)]



Symbol: PLC0 Address: 0026h After Reset: 0001 X010b

Bit Symbol	Bit Name	Function	RW
PLC00	PLL clock multiplication factor select bits[(3)(5)]	b2 b1 b0 0 1 0: Multiply-by-4 1 0 0: Multiply-by-8 Do not set to values other than the above	RW
PLC01			RW
PLC02			RW
– (b3)	Unimplemented. Write 0. Read as undefined value.		–
PLC04	Reference clock division rate select bits[(3)(5)]	b5 b4 0 0: No division 0 1: Divide-by-2 1 0: Divide-by-4 Do not set to values other than the above	RW
PLC05			RW
– (b6)	Reserved bit	Set to 0	RW
PLC07	Operation enable bit[(4)]	0: PLL stops 1: PLL runs	RW

NOTES:
1. Set the PLC0 register after the PRC0 bit in the PRCR register is set to 1 (write enable).
2. If the PM21 bit in the PM2 register is set to 1 (disables a clock change), a write to the PLC0 register has no effect.
3. Set bits PLC02 to PLC00 while the PLC07 bit is 0. Bits PLC02 to PLC00 can be written only once.
4. Enter wait mode or stop mode after the CM17 bit is set to 0 (main clock as CPU clock source) and then the PLC07 bit to 0.
5. The frequency of PLL clock is calculated by the following equation.

Set by bits PLC02 to PLC00.

$$\text{PLL clock frequency} = \text{Main clock frequency} \times \frac{1}{\text{Reference clock division rate}} \times \text{PLL clock multiplication factor}$$

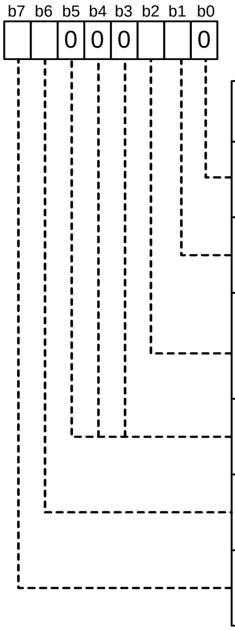
Set by bits PLC05 and PLC04.

e.g.) Main clock frequency: 10 MHz
Bits PLC02 to PLC00: 100b (multiply-by-8)
Bits PLC05 and PLC04: 10b (divide-by-4)

$$\text{PLL clock frequency} = 10\text{ MHz} \times \frac{8}{4} = 20\text{ MHz}$$

Figure 9.6 PLC0 Register

Processor Mode Register 2(1)



Symbol	Address	After Reset
PM2	0013h	00h

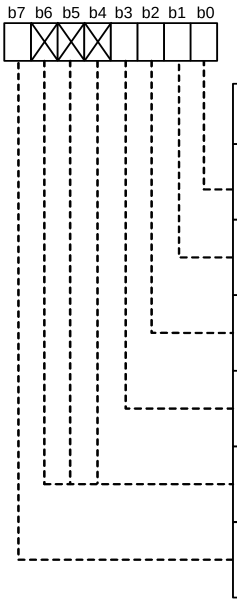
Bit Symbol	Bit Name	Function	RW
– (b0)	Reserved bit	Set to 0	RW
PM21	System clock protect bit(2, 3)	0: Protects a clock by the PRCR register 1: Disables a clock change	RW
PM22	WDT count source protect bit(2, 4)	0: CPU clock as count source for the watchdog timer 1: On-chip oscillator clock as count source for the watchdog timer	RW
– (b5-b3)	Reserved bits	Set to 0	RW
PM26	f2n clock source select bits	b7 b6 0 0: Clock selected by the CM21 bit 0 1: XIN clock (fXIND) 1 0: On-chip oscillator clock (fROC) 1 1: Do not set to this value	RW
PM27			RW

NOTES:

1. Set the PM2 register after the PRC1 bit in the PRCR register is set to 1 (write enable).
2. Once bits PM22 and PM21 are set to 1, they cannot be set to 0 by a program.
3. When the PM21 bit is set to 1;
   - the CPU clock does not stop, even if the WAIT instruction is executed
   - writes to the following bits have no effect
     - the CM02 bit in the CM0 register
     - the CM05 bit in the CM0 register
     - the CM07 bit in the CM0 register (CPU clock source is not changed)
     - the CM10 bit in the CM1 register (the MCU does not enter stop mode)
     - the CM17 bit in the CM1 register (CPU clock source is not changed)
     - the CM20 bit in the CM2 register (oscillation stop detect function setting is not changed)
     - all bits in registers PLC0 and PLC1 (PLL frequency synthesizer setting is not changed)
4. When the PM22 bit is set to 1;
   - the on-chip oscillator starts oscillating and the on-chip oscillator clock becomes the count source for the watchdog timer
   - write to the CM10 bit in the CM1 register is disabled (writing a 1 has no effect and the MCU does not enter stop mode)
   - the watchdog timer keeps operating when the MCU is in wait mode or in hold state

Figure 9.7 PM2 Register

Count Source Prescaler Register



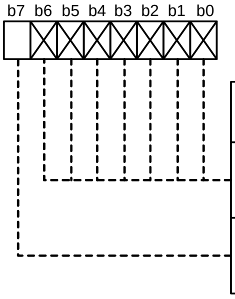
Symbol	Address	After Reset[2]
TCSPR	035Fh	0XXX 0000b

Bit Symbol	Bit Name	Function	RW
CNT0	Division rate select bits[1]	If the setting value is n, f2n is the main clock, on-chip oscillator clock, or PLL clock divided by 2n. When n is set to 0, no division is selected	RW
CNT1			RW
CNT2			RW
CNT3			RW
– (b6-b4)	Reserved bits	Read as undefined value	–
CST	Operation enable bit	0: Divider stops 1: Divider operates	RW

NOTES:

1. Set bits CNT3 to CNT0 after the CST bit is set to 0.
2. The TCSPR register maintains values set before reset, even after the software reset or watchdog timer reset has been performed.

Clock Prescaler Reset Register



Symbol	Address	After Reset
CPSRF	0341h	0XXX XXXXb

Bit Symbol	Bit Name	Function	RW
– (b6-b0)	Unimplemented. Write 0. Read as undefined value.		–
CPSR	Clock prescaler reset bit	When the CPSR bit is set to 1, a divider for fC32 is reset. Read as 0.	RW

Figure 9.8 TCSPR Register, CPSRF Register

Flash Memory Control Register 4[1]

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

Symbol FMR4 Address 0059h After Reset 00h

Bit Symbol	Bit Symbol	Function	RW
FMR40	Flash memory low-speed access enable bit	0: Normal-speed access 1: Low-speed access[2]	RW
– (b7-b1)	Reserved bits	Set to 0	RW

NOTES:

1. Set the FMR4 register in 8-bit units. To set the FMR40 bit to 1, write a 1 to the FMR40 bit immediately after writing a 0 to the bit. Do not generate an interrupt or a DMA or DMACII transfer between these two settings.
2. The FMR40 bit can be set to 1 only when the CPU clock frequency is 2 MHz or lower.

Voltage Regulator Control Register[1]

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒	☒	☒	☒	

Symbol VRCR Address 001Fh After Reset 00h

Bit Symbol	Bit Symbol	Function	RW
MRS	Main voltage regulator stop bit[3]	0 : Main voltage regulator operates 1 : Main voltage regulator stops[2]	RW
– (b7-b1)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:

1. Set the VRCR register after the PRC3 bit in the PRCR register is set to 1 (write enable).
2. The MRS bit can be set to 1 when all the conditions as follows are met;
   a. In low-power consumption mode
   b. The FMR40 bit is 1
   c. On-chip oscillator stops (the CM21 bit in the CM2 register is 0, the PM22 bit in the PM2 register is 0, and bits PM27 and PM26 are 00b)
3. When the PLC07 bit in PLC0 register is 1 (PLL runs) or the CM05 bit in the CM0 register is 0 (Main clock oscillates), the MRS bit becomes 0.

Figure 9.9 FMR4 Register, VRCR Register

9.1.1 Main Clock

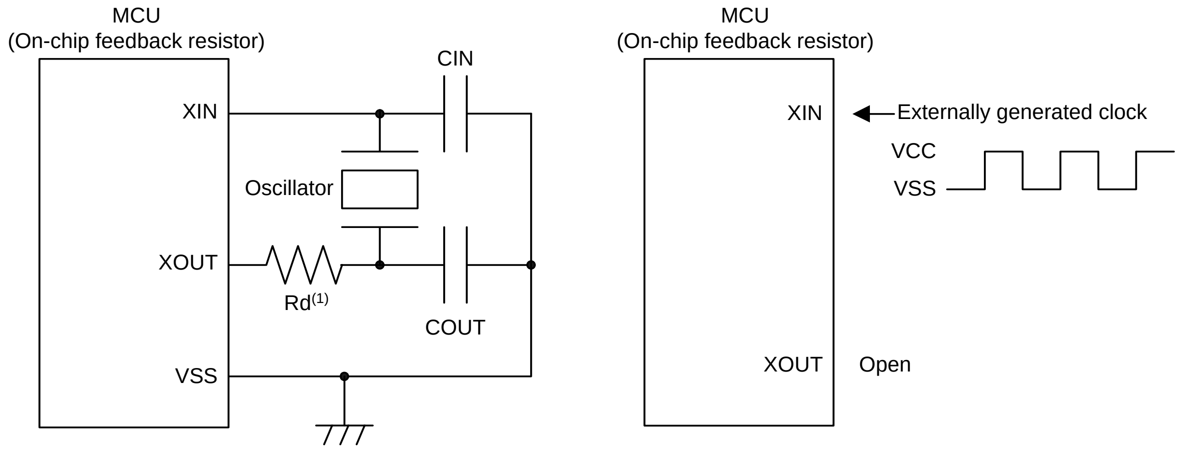
Main clock oscillation circuit generates the main clock. The main clock is used as the clock source for the CPU clock and peripheral function clocks.

The main clock oscillation circuit is configured by connecting an oscillator between the XIN and XOUT pins. The circuit has an on-chip feedback resistor. The feedback resistor is disconnected from the oscillation circuit in stop mode to reduce power consumption. The main clock oscillation circuit may also be configured by feeding an externally generated clock to the XIN pin. Figure 9.10 shows examples of main clock circuit connection. Circuit constants vary depending on each oscillator. Use the circuit constant recommended by each oscillator manufacturer.

The main clock divided-by-eight becomes the CPU clock source after reset.

To reduce power consumption, set the CM05 bit in the CM0 register to 1 (main clock stopped) after the sub clock or on-chip oscillator clock is selected as the CPU clock sources. In this case, the XOUT pin outputs an "H" signal. The XIN pin is pulled up to the XOUT pin via the feedback resistor which remains on. When an external clock is input to the XIN pin, do not set the CM05 bit to 1.

All clocks, including the main clock, stop in stop mode. Refer to **9.5 Power Consumption Control** for details.


MCU
(On-chip feedback resistor)
XIN
XOUT
VSS
CIN
Oscillator
Rd(1)
COUT
MCU
(On-chip feedback resistor)
XIN
Externally generated clock
VCC
VSS
XOUT
Open

NOTE:
1. Insert a damping resistor if required. Resistance values vary depending on the oscillator setting. Use the resistance values recommended by the oscillator manufacturer.
If the oscillator manufacturer recommends that a feedback resistor be added to the chip externally, insert a feedback resistor between XIN and XOUT following the instructions.

Figure 9.10 Main Clock Circuit Connection

9.1.2 Sub Clock

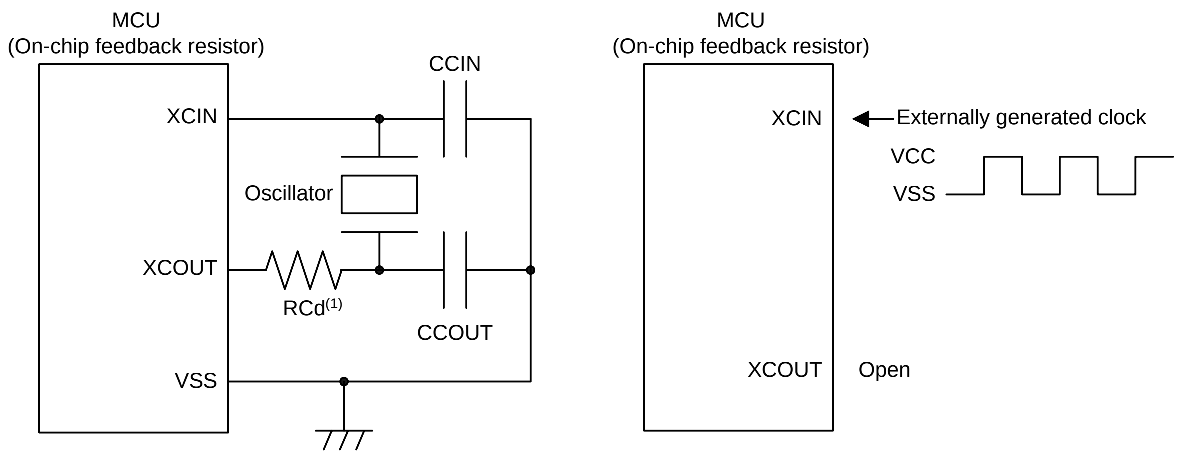
Sub clock oscillation circuit generates the sub clock. The sub clock is used as the clock source for the CPU clock and for timer A and timer B. fC, which has the same frequency as the sub clock can be output from the CLKOUT pin.

The sub clock oscillation circuit is configured by connecting a crystal oscillator between the XCIN and XCOUT pins. The circuit has an on-chip feedback resistor. The feedback resistor is disconnected from the oscillation circuit in stop mode to reduce power consumption. The sub clock oscillation circuit may also be configured by feeding an externally generated clock to the XCIN pin. Figure 9.11 shows an example of sub clock circuit connection. Circuit constants vary depending on each oscillator. Use the circuit constant recommended by each oscillator manufacturer.

The sub clock is stopped after reset, and the feedback resistor is disconnected from the oscillation circuit. To start oscillating the sub clock oscillation circuit, set both the PD8_7 and PD8_6 bits in the PD8 register to 0 (input mode), the PU25 bit in the PUR2 register to 0 (not pulled up), and then the CM04 bit in the CM0 register to 1 (XCIN-XCOUT oscillation function). To input the externally generated clock to the XCIN pin, set the PD8_7 bit to 0, the PU25 bit to 0, and then the CM04 bit to 1. A clock input to the XCIN pin becomes the clock source for the sub clock.

When the CM07 bit in the CM0 register is set to 1 (sub clock) after the sub clock oscillation stabilizes, the sub clock becomes the CPU clock source.

All clocks, including the sub clock, stop in stop mode. Refer to **9.5 Power Consumption Control** for details.


MCU
(On-chip feedback resistor)
XCIN
XCOUT
VSS
CCIN
Oscillator
RCd(1)
CCOUT
MCU
(On-chip feedback resistor)
XCIN
Externally generated clock
VCC
VSS
XCOUT
Open

NOTE:
1. Insert a damping resistor if required. Resistance values vary depending on the oscillator setting. Use the resistance values recommended by the oscillator manufacturer.
If the oscillator manufacturer recommends that a feedback resistor be added to the chip externally, insert a feedback resistor between XCIN and XCOUT following the instructions.

Figure 9.11 Sub Clock Circuit Connection

9.1.3 On-Chip Oscillator Clock

On-chip oscillator generates the 1-MHz on-chip oscillator clock. The on-chip oscillator clock is used as the clock source for the CPU clock and peripheral function clocks.

The on-chip oscillator clock is stopped after reset. When the CM21 bit in the CM2 register is set to 1 (on-chip oscillator clock), the on-chip oscillator starts oscillating and becomes the clock source for the CPU clock and peripheral function clocks in place of the main clock.

Table 9.2 lists on-chip oscillator start conditions.

Table 9.2 On-Chip Oscillator Start Condition

CM2 Register	PM2 Register		Applications
CM21	PM22	PM27, PM26	
1	0	00b	Clock source for the CPU clock and peripheral function clock
0	1	00b	Count source for the watchdog timer
0	0	10b	Clock source for f2n

9.1.3.1 Oscillation Stop Detect Function

When the main clock is terminated running by an external factor, the on-chip oscillator automatically starts oscillating to provide the clock.

When the CM 20 bit in the CM2 register is set to 1 (oscillation stop detect function used), an oscillation stop detection interrupt request is generated as soon as the main clock is lost. Simultaneously, the on-chip oscillator starts oscillating. The on-chip oscillator clock takes the place of the main clock as the clock source for the CPU clock and peripheral function clocks. Associated bits in the CM2 register are changed as follows:

- CM21 bit becomes 1 (on-chip oscillator clock becomes the CPU clock)
- CM22 bit becomes 1 (loss of main clock stop is detected)
- CM23 bit becomes 1 (main clock stops)

The oscillation stop detection interrupt shares the vector with the watchdog timer interrupt and the voltage monitor interrupt. When these interrupts are used simultaneously, verify the CM22 bit in the interrupt routine to determine if an oscillation stop detection interrupt request has been generated.

When the main clock resumes its operation after a loss of the main clock is detected, the main clock can be selected as the clock source for the CPU clock and peripheral function clocks by a program. Figure 9.12 shows the procedure to switch the clock source from the on-chip oscillator clock to the main clock.

In low-speed mode, when the main clock is lost while the CM20 bit is set to 1, an oscillation stop detection interrupt request is generated, and the on-chip oscillator starts oscillating. The sub clock remains as the source for the CPU clock. The on-chip oscillator clock becomes the source for the peripheral function clocks.

When the peripheral function clocks are stopped, the oscillation stop detect function cannot be used. To enter wait mode while using the oscillation stop detect function, set the CM02 bit in the CM0 register to 0 (peripheral clocks do not stop in wait mode).

The oscillation stop detect function is a precaution against the unintended termination of the main clock by an external factor. Set the CM20 bit to 0 (oscillation stop detect function not used) when the main clock is stopped by a program, i.e., entering stop mode or setting the CM05 bit in the CM0 register to 1 (main clock stops).

When the main clock frequency is 2 MHz or lower, the oscillation stop detect function is not available. In this case, set the CM20 bit to 0.

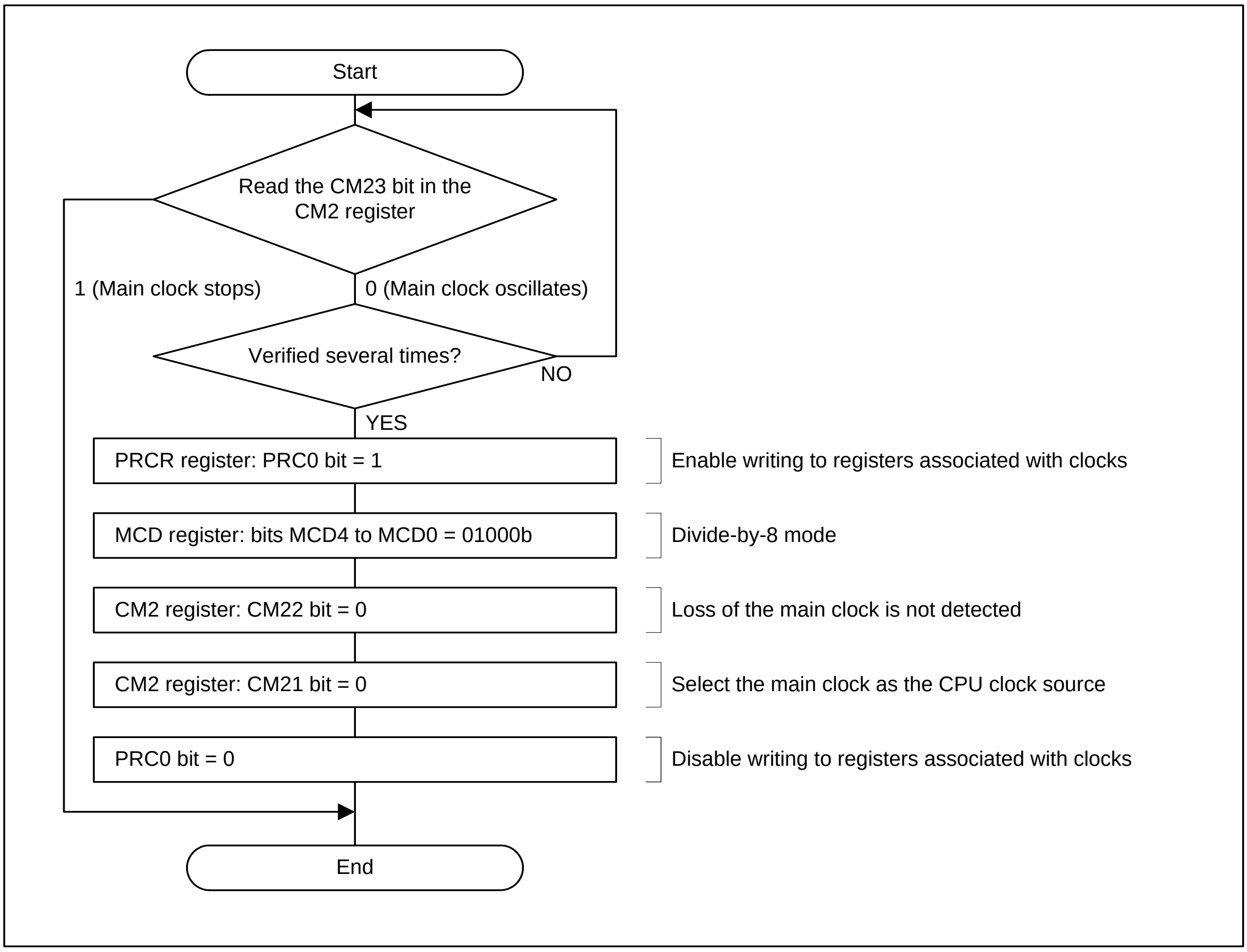
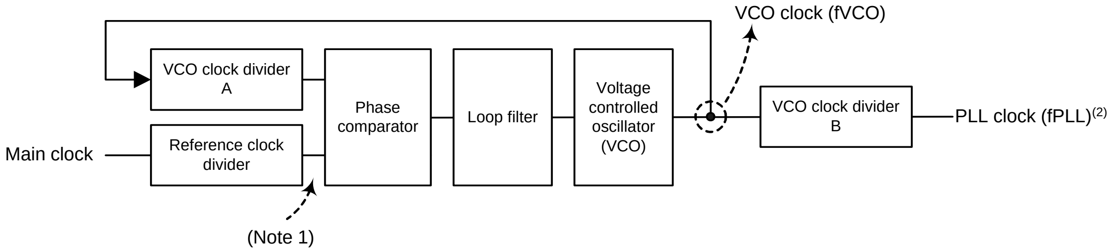

Start
Read the CM23 bit in the CM2 register
1 (Main clock stops)
0 (Main clock oscillates)
Verified several times?
NO
YES
PRCR register: PRC0 bit = 1
Enable writing to registers associated with clocks
MCD register: bits MCD4 to MCD0 = 01000b
Divide-by-8 mode
CM2 register: CM22 bit = 0
Loss of the main clock is not detected
CM2 register: CM21 bit = 0
Select the main clock as the CPU clock source
PRC0 bit = 0
Disable writing to registers associated with clocks
End

Figure 9.12 Procedure to Switch from On-chip Oscillator Clock to Main Clock

9.1.4 PLL Clock

The PLL frequency synthesizer generates the PLL clock by multiplying the main clock. The PLL clock can be used as the clock source for the CPU clock and peripheral function clocks. Figure 9.13 shows the block diagram of PLL frequency synthesizer.



NOTES:
1. The frequency after the main clock is divided by the reference clock divider is set to be from 2 MHz to 4 MHz.
2. Set the PLL clock frequency as follows: 10 MHz ≤ the PLL clock frequency ≤ 32 MHz

Figure 9.13 PLL Frequency Synthesizer Block Diagram

The PLL clock frequency is calculated by the following equation. Table 9.3 lists the PLL clock frequency settings examples.

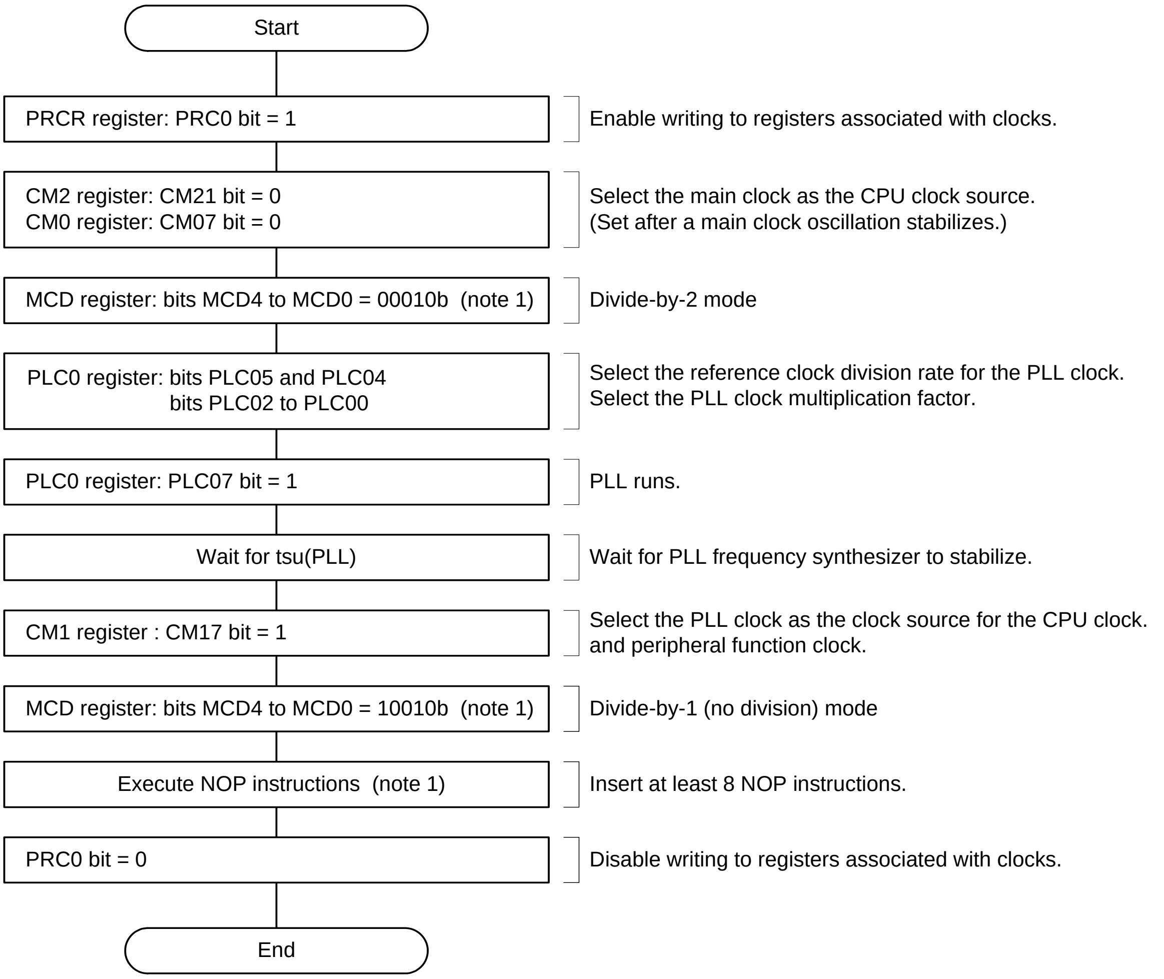
$$\text{PLL clock frequency} = \text{Main clock frequency} \times \frac{\text{Division rate of VCO clock divider A}}{\underbrace{\text{Division rate of the reference clock divider}}_{\text{Set by bits PLC05 and PLC04}}} \times \frac{1}{\underbrace{\text{Division rate of VCO clock divider B}}_{\text{Set by bits PLC02 to PLC00}}}$$

PLC05, PLC04, and PLC02 to PLC00: Bits in the PLC0 register

Table 9.3 Example of the PLL Clock Frequency Setting

Main Clock (fXIN)	PLC0 Register		PLL Clock (fPLL)
	Bits PLC05 and PLC04	Bits PLC02 to PLC00	
5 MHz	01b (Divide-by-2)	100b (Multiply-by-8)	fPLL = fXIN × 1/2 × 8 = 20 MHz
10 MHz	10b (Divide-by-4)	100b (Multiply-by-8)	fPLL = fXIN × 1/4 × 8 = 20 MHz
8 MHz	01b (Divide-by-2)	100b (Multiply-by-8)	fPLL = fXIN × 1/2 × 8 = 32 MHz
16 MHz	10b (Divide-by-4)	100b (Multiply-by-8)	fPLL = fXIN × 1/4 × 8 = 32 MHz

The PLL frequency synthesizer is stopped after reset. When the PLC07 bit in the PLC0 register is set to 1 (PLL runs), the PLL frequency synthesizer starts operating. Waiting time, *tsu(PLL)*, is required before the PLL clock is stabilized.
Prior to entering wait mode or stop mode, set the CM17 bit in the CM1 register to 0 (main clock as CPU clock source), and then set the PLC07 bit to 0 (PLL stops).
Figure 9.14 shows the procedure to use the PLL clock as the CPU clock source.


Start
PRCR register: PRC0 bit = 1
Enable writing to registers associated with clocks.
CM2 register: CM21 bit = 0
CM0 register: CM07 bit = 0
Select the main clock as the CPU clock source.
(Set after a main clock oscillation stabilizes.)
MCD register: bits MCD4 to MCD0 = 00010b (note 1)
Divide-by-2 mode
PLC0 register: bits PLC05 and PLC04
bits PLC02 to PLC00
Select the reference clock division rate for the PLL clock.
Select the PLL clock multiplication factor.
PLC0 register: PLC07 bit = 1
PLL runs.
Wait for tsu(PLL)
Wait for PLL frequency synthesizer to stabilize.
CM1 register : CM17 bit = 1
Select the PLL clock as the clock source for the CPU clock.
and peripheral function clock.
MCD register: bits MCD4 to MCD0 = 10010b (note 1)
Divide-by-1 (no division) mode
Execute NOP instructions (note 1)
Insert at least 8 NOP instructions.
PRC0 bit = 0
Disable writing to registers associated with clocks.
End

NOTE:
1. Execute these steps in the flash memory version.

Figure 9.14 Procedure to Use PLL Clock as CPU Clock Source

9.2 CPU Clock and BCLK

The CPU clock is used to operate the CPU and also used as the count source for the watchdog timer. After reset, the CPU clock is the main clock divided by eight. The bus clock (BCLK) has the same frequency as the CPU clock and can be output from the BCLK pin in memory expansion mode or microprocessor mode. Refer to **9.4 Clock Output Function** for details.

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

When the main clock, on-chip oscillator clock, or PLL clock is selected as the clock source for the CPU clock, the selected clock source divided by 1 (no division), 2, 3, 4, 6, 8, 10, 12, 14, or 16 becomes the CPU clock. Bits MCD4 to MCD0 in the MCD register select the clock division. When the MCU enters stop mode or low-power consumption mode, bits MCD4 to MCD0 are set to 01000b (divide-by-8 mode). Therefore, when the CPU clock source is switched to the main clock next time, the CPU clock is the main clock divided by eight. Refer to **9.5 Power Consumption Control** for details.

In the flash memory version, if CPU clock frequency is set to be 20MHz or more, first set bits MCD4 to MCD0 to 00010b (divide-by-2 mode), and then set to 10010b (no division mode). Also, insert at least eight NOP instructions after the instruction to set to no division mode.

9.3 Peripheral Function Clock

The peripheral function clocks are used to operate the peripheral functions excluding the watchdog timer. The clock selected by the CM17 bit in the CM1 register and the CM21 bit in the CM2 register (any of the main clock, PLL clock, or on-chip oscillator clock) becomes the peripheral function clock source (fPFC).

9.3.1 f1, f8, f32, and f2n

f1, f8 and f32 are fPFC divided by 1, 8, or 32.

Bits PM27 and PM 26 in the PM2 register select the f2n clock source from fPFC, XIN clock (fXIND), and the on-chip oscillator clock (fROC). Bits CNT3 to CNT0 in the TCSPR register select the f2n division. (n = 1 to 15. No division when n = 0.)

When wait mode is entered while the CM02 bit in the CM0 register is set to 1 (peripheral clocks stop in wait mode) or when the CM05 bit is set to 1 using the main clock as the peripheral function clock source, fPFC stops. When bits PM27 and PM26 in the PM2 register are set to 10b (on-chip oscillator clock is selected for the f2n clock source), f2n does not stop in wait mode.

f1, f8, and f2n are used to operate the serial interface and also is used as the count source for timer A and timer B.

The CLKOUT pin outputs f8 and f32. Refer to **9.4 Clock Output Function** for details.

9.3.2 fAD

fAD is used to operate the A/D converter and has the same frequency as fPFC.

When wait mode is entered while the CM02 bit in the CM0 register is set to 1 (peripheral clocks stop in wait mode) or when the CM05 bit is set to 1 using the main clock as the peripheral function clock source, fAD stops.

9.3.3 fC32

fC32 is the sub clock divided by 32. fC32 is used as the count source for timer A and timer B. fC32 is available if the sub clock is running.

9.4 Clock Output Function

The CLKOUT pin outputs fC, f8, or f32.

The BCLK clock, which has the same frequency as the CPU clock, can be output from the BCLK pin in memory expansion mode or microprocessor mode.

Table 9.4 lists CLKOUT pin function in single-chip mode. Table 9.5 lists CLKOUT pin function in memory expansion mode and microprocessor mode.

Table 9.4 CLKOUT Pin Function in Single-Chip Mode

CM0 Register[1]	P5_3/CLKOUT Pin Function
Bits CM01 and CM00	
00b	I/O port P5_3
01b	Outputs fC
10b	Outputs f8
11b	Outputs f32

NOTE:

1. Rewrite the CM0 register after setting the PRC0 bit in the PRCR register to 1 (write enable).

Table 9.5 CLKOUT Pin Function in Memory Expansion Mode and Microprocessor Mode

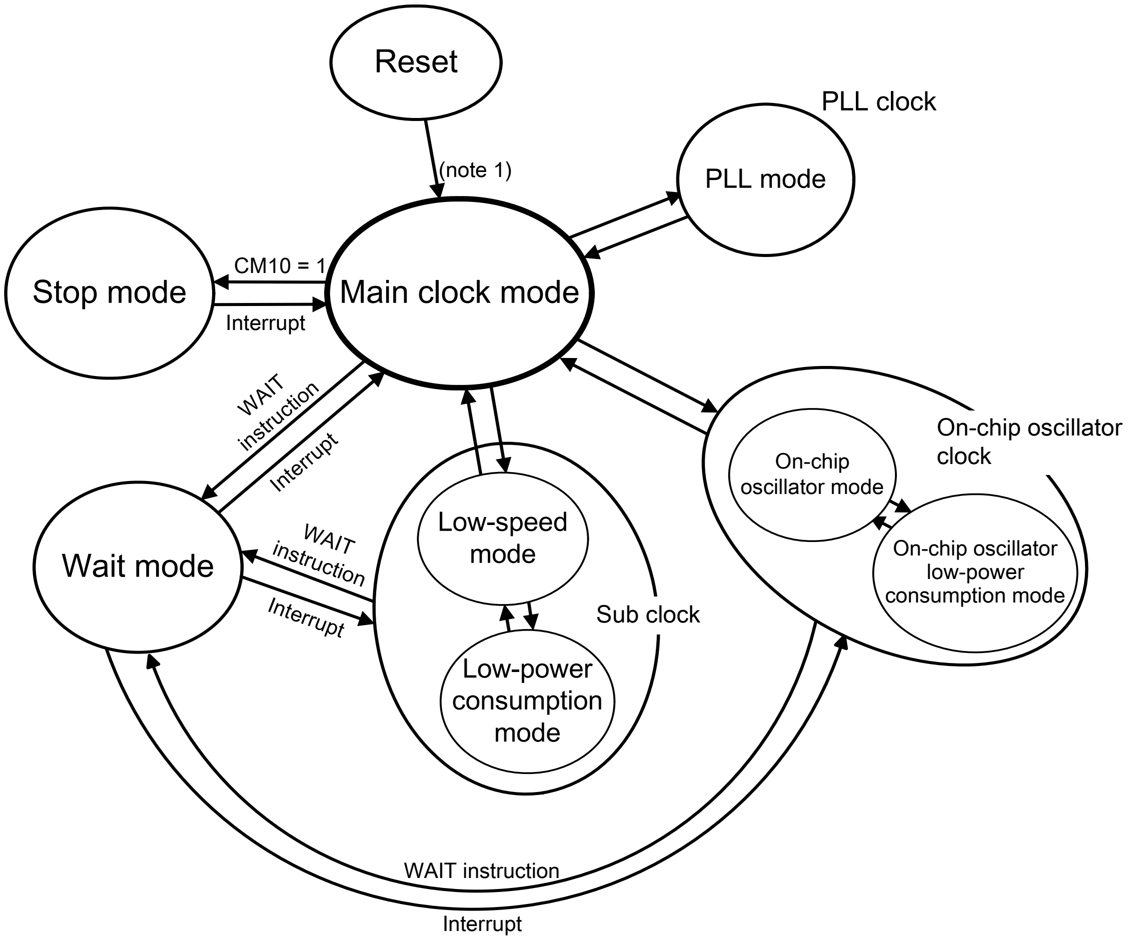
CM0 Register[1]	PM1 Register[2]	PM0 Register[2]	CLKOUT/BCLK/ALE Pin Function
Bits CM01 and CM00	Bits PM15 and PM14	PM07 bit	
00b	00b 10b 11b	0	Outputs BCLK
		1	Outputs “L” (does not function as P5_3)
	01b	0 or 1	Outputs ALE
01b	0 or 1	0 or 1	Outputs fC
10b	0 or 1	0 or 1	Outputs f8
11b	0 or 1	0 or 1	Outputs f32

NOTES:

1. Change the CM0 register after setting the PRC0 bit in the PRCR register to 1 (write enable).
2. Change registers PM0 and PM1 after setting the PRC1 bit in the PRCR register to 1 (write enable).

9.5 Power Consumption Control

The power consumption control is enabled by controlling a CPU clock frequency. The higher the CPU clock frequency is, the more the processing power is available. The lower the CPU clock frequency is, the less power is consumed. When unnecessary oscillation circuits are stopped, power consumption is further reduced.
CPU operating mode, wait mode, and stop mode are provided as the power consumption control. CPU operating mode is further separated into the following modes; main clock mode, PLL mode, low-speed mode, low-power consumption mode, on-chip oscillator mode, and on-chip oscillator low-power consumption mode.
Figure 9.15 shows a mode transition diagram.



CM10: bit in the CM1 register
NOTE:
1. Bits MCD4 to MCD0 in the MCD register become 01000b (divide-by-8 mode) after reset.

Figure 9.15 Mode Transition

9.5.1 CPU operating mode

The CPU clock can be selected from the main clock, sub clock, on-chip oscillator clock, or PLL clock. When switching the CPU clock source, wait until the new CPU clock source stabilizes. To change the CPU clock source from the sub clock, on-chip oscillator clock, or PLL clock, set it to the main clock once and then switch it to another clock.
To switch the CPU clock source from the on-chip oscillator clock to the main clock, set bits MCD4 to MCD0 in the MCD register to 01000b (divided-by-8 mode) in on-chip oscillator mode.
Table 9.6 lists bit setting and operation mode associated with clocks.

9.5.1.1 Main Clock Mode

The main clock divided by 1 (no division), 2, 3, 4, 6, 8, 10, 12, 14, or 16 is used as the source for the CPU clock. The main clock is also used as the source for fPFC. When the sub clock is running, fC32 can be used as the count source for timer A and timer B.

9.5.1.2 PLL Mode

The PLL clock divided by 1 (no division), 2, 3, 4, 6, 8, 10, 12, 14, or 16 is used as the source for the CPU clock. The PLL clock is also used as the source for fPFC. When the sub clock is running, fC32 can be used as the count source for timer A and timer B.

In the flash memory version, if CPU clock frequency is set to be 20MHz or more, first set bits MCD4 to MCD0 in the MCD register to 00010b (divide-by-2 mode), and then set to 10010b (no division mode). Also, insert at least eight NOP instructions after the instruction to set to no division mode.

9.5.1.3 Low-Speed Mode

The sub clock is used as the source for the CPU clock. The main clock, PLL clock, or on-chip oscillator clock can be selected as the source for fPFC by setting bits CM17 and CM21 after the CPU clock is switched to the sub clock using the CM07 bit. In low-speed mode, fC32 can be used as the count source for timer A and timer B.

Out of CPU operating modes, only main clock mode and low-power consumption mode can be entered from low-speed mode. Enter main clock mode first prior to entering different CPU operating modes other than the low-power consumption mode.

9.5.1.4 Low-Power Consumption Mode

The MCU enters low-power consumption mode when the main clock stops in low-speed mode. The sub clock is used as the source for the CPU clock. The on-chip oscillator clock can be selected as the source for fPFC by setting the CM21 bit after entering low-power consumption mode. fC32 can be used as the count source for timer A and timer B. When low-power consumption mode is entered, bits MCD4 to MCD0 in the MCD register become 01000b (divide-by-8 mode). Therefore, when next time the CPU clock source is switched to the main clock, the CPU clock is the main clock divided by eight. However, bits MCD4 to MCD0 do not become 01000b if the main clock is stopped by setting the CM05 bit to 1 while the on-ship oscillator clock is selected as the source for fPFC in low-speed mode. In this case, set bits MCD4 to MCD0 to 01000b by a program and then switch the CPU clock source to the main clock.Figure 9.16 shows the procedure to enter low-power consumption mode from main clock mode.

9.5.1.5 On-Chip Oscillator Mode

The on-chip oscillator clock divided by 1 (no division), 2, 3, 4, 6, 8, 10, 12, 14, or 16 is used as the source for the CPU clock. The on-chip oscillator clock is also used as the source for fPFC. When the sub clock is running, fC32 can be used as the count source for timer A and timer B.

9.5.1.6 On-Chip Oscillator Low-Power Consumption Mode

The MCU enters on-chip oscillator low-power consumption mode when the main clock stops in on-chip oscillator mode. The on-chip oscillator clock divided by 1 (no division), 2, 3, 4, 6, 8, 10, 12, 14, or 16 is used as the source for the CPU clock. The on-chip oscillator clock is also used as the source for fPFC. When the sub clock is running, fC32 can be used as the count source for timer A and timer B.

9.5.1.7 Flash Memory Low-Speed Access

When the CPU clock frequency is 2 MHz or lower, power consumption can be reduced by setting the FMR40 bit in the FMR4 register to 1 (low-speed access). To configure low-speed access, set the FMR40 bit to 1 after setting the CPU clock frequency to 2 MHz or lower. To set the CPU clock frequency to higher than 2 MHz, change the frequency after setting the FMR40 bit to 0 (normal-speed access).

9.5.1.8 Main Voltage Regulator Stops

Power consumption can be reduced by stopping the main voltage regulator in low-power consumption mode. To stop the main voltage regulator, set the MRS bit in the VRCR register to 1 (main voltage regulator stops) after all the following conditions are met.

(1) Low-power consumption mode
(2) On-chip oscillator stops
- The CM21 bit in the CM2 register is set to 0
- The PM22 bit in the PM2 register is set to 0
- Bits PM27 and PM26 in the PM2 register are set to 00b

(3) The FMR40 bit in the FMR4 register is set to 1 (flash memory low-speed access)

Set the MRS bit to 0 (main voltage regulator operates) and wait for 50 µs or more before performing any of the following settings: changing CPU operating mode to other than low-power consumption mode, starting the on-chip oscillator, or setting the FMR40 bit 0 (flash memory normal-speed access).

Table 9.6 Operation Mode Setting

CPU Clock Source	Operating Mode	Oscillation Control				Selector	
		CM0 Register		PLC0 Register	CM2 Register	CM1 Register	CM0 Register
		CM05	CM04	PLC07	CM21[1]	CM17	CM07
Main clock	Main clock mode	0	0 or 1	0 or 1	0	0	0
PLL clock	PLL mode	0	0 or 1	1	0	1	0
Sub clock	Low-speed mode	0	1	0 or 1	0	0	1
	Low power consumption mode	1	1	0	0	0	1
On-chip oscillator clock	On-chip oscillator mode	0	0 or 1	0 or 1	1	0	0
	On-chip oscillator low-power consumption mode	1	0 or 1	0	1	0	0

NOTE:
1. The CM21 bit in the CM2 register has both the oscillation control and selector functions.

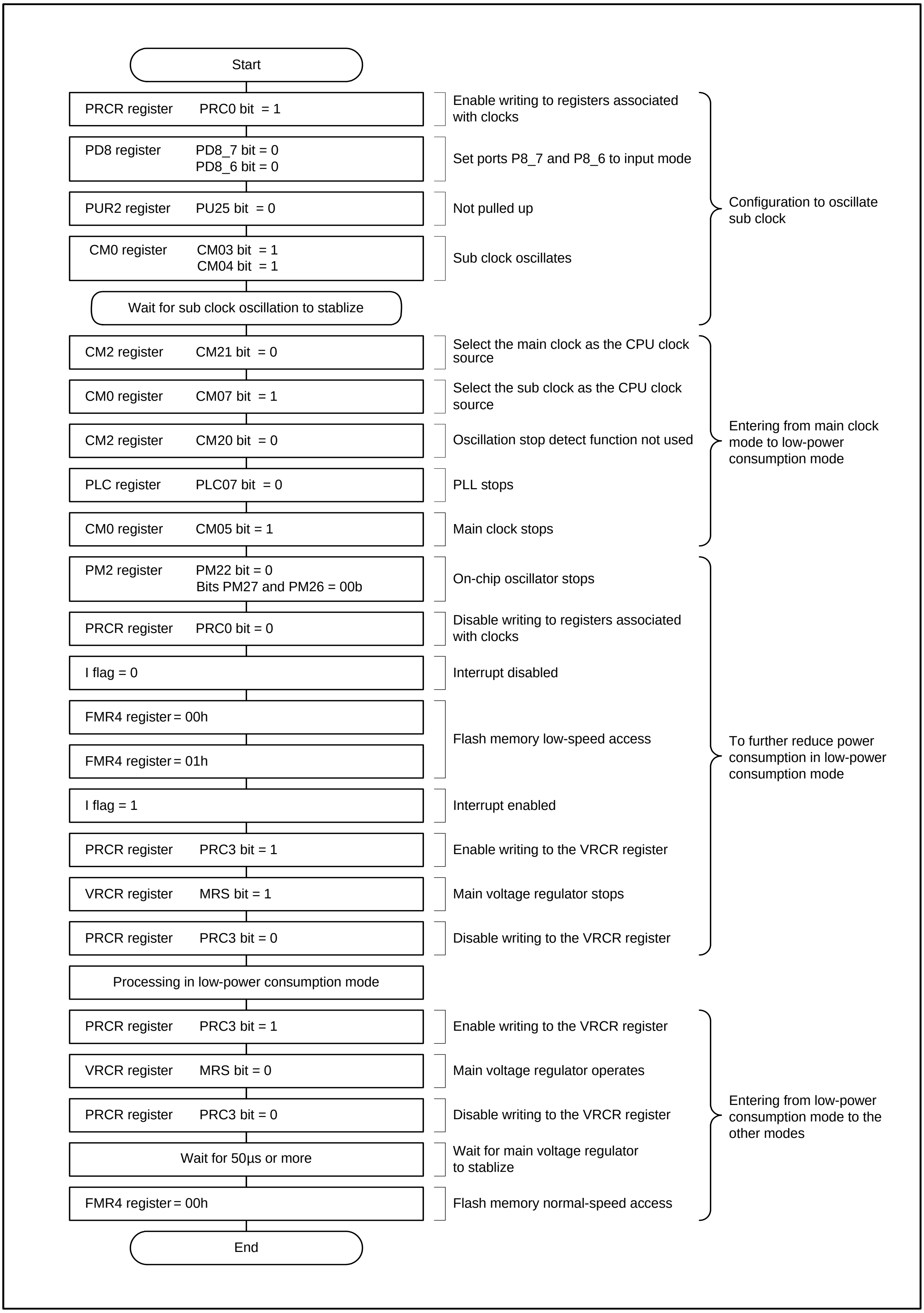

Start
PRCR register PRC0 bit = 1
Enable writing to registers associated with clocks
PD8 register PD8_7 bit = 0 PD8_6 bit = 0
Set ports P8_7 and P8_6 to input mode
PUR2 register PU25 bit = 0
Not pulled up
CM0 register CM03 bit = 1 CM04 bit = 1
Sub clock oscillates
Wait for sub clock oscillation to stablize
Configuration to oscillate sub clock
CM2 register CM21 bit = 0
Select the main clock as the CPU clock source
CM0 register CM07 bit = 1
Select the sub clock as the CPU clock source
CM2 register CM20 bit = 0
Oscillation stop detect function not used
PLC register PLC07 bit = 0
PLL stops
CM0 register CM05 bit = 1
Main clock stops
Entering from main clock mode to low-power consumption mode
PM2 register PM22 bit = 0 Bits PM27 and PM26 = 00b
On-chip oscillator stops
PRCR register PRC0 bit = 0
Disable writing to registers associated with clocks
I flag = 0
Interrupt disabled
FMR4 register = 00h
FMR4 register = 01h
Flash memory low-speed access
I flag = 1
Interrupt enabled
PRCR register PRC3 bit = 1
Enable writing to the VRCR register
VRCR register MRS bit = 1
Main voltage regulator stops
PRCR register PRC3 bit = 0
Disable writing to the VRCR register
To further reduce power consumption in low-power consumption mode
Processing in low-power consumption mode
PRCR register PRC3 bit = 1
Enable writing to the VRCR register
VRCR register MRS bit = 0
Main voltage regulator operates
PRCR register PRC3 bit = 0
Disable writing to the VRCR register
Wait for 50µs or more
Wait for main voltage regulator to stablize
FMR4 register = 00h
Flash memory normal-speed access
Entering from low-power consumption mode to the other modes
End

Figure 9.16 Procedure to Enter Low-Power Consumption Mode From Main Clock Mode

9.5.2 Wait Mode

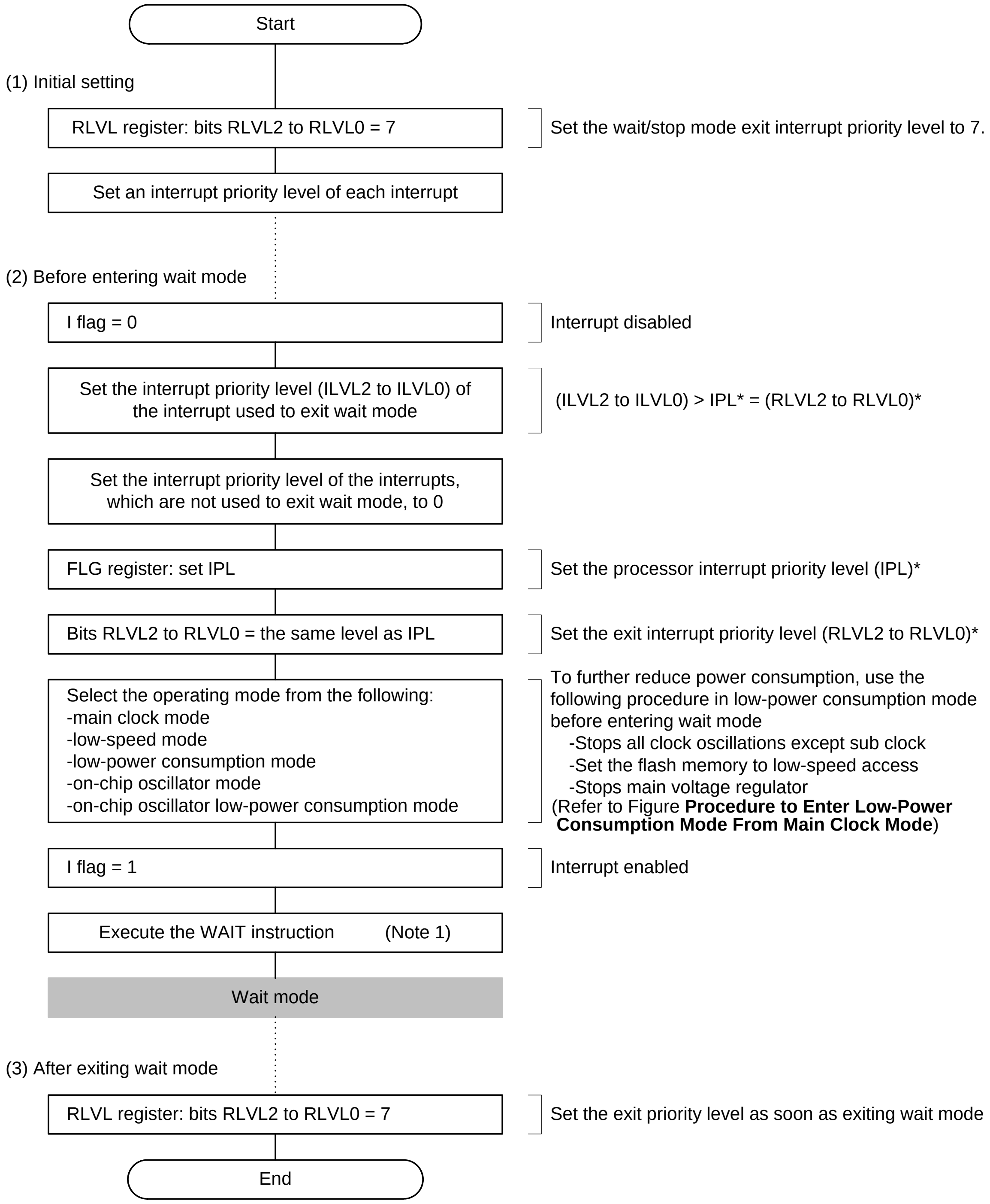
In wait mode, the CPU and watchdog timer stop operating. If the PM22 bit in the PM2 register is set to 1 (on-chip oscillator clock as watchdog timer count source), the watchdog timer continues operating. Since the main clock, sub clock, and on-chip oscillator clock continue running, peripheral functions using these clocks as their clock source also continue to operate.

9.5.2.1 Peripheral Function Clock Stop Function

If the CM02 bit in the CM0 register is set to 1 (peripheral clocks stop in wait mode), fAD, f1, f8, and f32 stop in wait mode. f2n, which uses the clock selected by the CM21 bit in the CM2 register as its clock source, also stops in wait mode. Power consumption can be reduced by stopping these peripheral clocks. f2n, which uses the XIN clock (fXIND) or on-chip oscillator clock as its clock source, and fC32 do not stop even in wait mode.

9.5.2.2 Entering Wait Mode

Figure 9.17 shows a procedure to enter wait mode.


Start
(1) Initial setting
RLVL register: bits RLVL2 to RLVL0 = 7
Set the wait/stop mode exit interrupt priority level to 7.
Set an interrupt priority level of each interrupt
(2) Before entering wait mode
I flag = 0
Interrupt disabled
Set the interrupt priority level (ILVL2 to ILVL0) of the interrupt used to exit wait mode
(ILVL2 to ILVL0) > IPL* = (RLVL2 to RLVL0)*
Set the interrupt priority level of the interrupts, which are not used to exit wait mode, to 0
FLG register: set IPL
Set the processor interrupt priority level (IPL)*
Bits RLVL2 to RLVL0 = the same level as IPL
Set the exit interrupt priority level (RLVL2 to RLVL0)*
Select the operating mode from the following:
-main clock mode
-low-speed mode
-low-power consumption mode
-on-chip oscillator mode
-on-chip oscillator low-power consumption mode
To further reduce power consumption, use the following procedure in low-power consumption mode before entering wait mode
-Stops all clock oscillations except sub clock
-Set the flash memory to low-speed access
-Stops main voltage regulator
(Refer to Figure **Procedure to Enter Low-Power Consumption Mode From Main Clock Mode**)
I flag = 1
Interrupt enabled
Execute the WAIT instruction (Note 1)
Wait mode
(3) After exiting wait mode
RLVL register: bits RLVL2 to RLVL0 = 7
Set the exit priority level as soon as exiting wait mode
End

NOTE:
1. Insert at least 4 NOP's after WAIT instruction.

Figure 9.17 Procedure to Enter Wait Mode

9.5.2.3 Pin States in Wait Mode

Table 9.7 lists pin states in wait mode.

Table 9.7 Pin States in Wait Mode

Pin		Memory Expansion Mode Microprocessor Mode	Single-Chip Mode
Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BHE}}$		Maintain the state immediately before entering wait mode	
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$		“H”	
$\overline{\text{HLDA}}$, BCLK		“H”	
ALE		“L”	
Ports		Maintain the state immediately before entering wait mode	
CLKOUT	When fC is selected	Continue to output the clock	
	When f8, f32 are selected	• When the CM02 bit in the CM0 register is 0 (peripheral clocks do not stop in wait mode): Continue to output the clock • When the CM02 bit is 1 (peripheral clocks stop in wait mode): The clock is stopped and holds the level immediately before entering wait mode	

9.5.2.4 Exiting Wait Mode

Wait mode is exited by the hardware reset, $\overline{\text{NMI}}$ interrupt, voltage monitor interrupt, or peripheral function interrupts.

As for a peripheral function interrupt that is not used to exit wait mode, set bits ILVL2 to ILVL0 in the corresponding Interrupt Control Register to 000b (interrupt disabled) before executing the WAIT instruction.

The CM02 bit setting in the CM0 register affects the use of the peripheral function interrupts to exit wait mode. When the CM02 bit is set to 0 (peripheral clocks do not stop in wait mode), any peripheral function interrupts can be used to exit wait mode. When the CM02 bit is set to 1 (peripheral clocks stop in wait mode), the peripheral functions clocked by the peripheral function clocks stop, and therefore, the peripheral function interrupts cannot be used to exit wait mode. However, the peripheral functions clocked by the external clock and fC32 do not stop regardless of the CM02 bit setting. Also, f2n, which uses the XIN clock (fXIND) or on-chip oscillator clock as its clock source does not stop. The interrupts generated by the peripheral functions which operate using these clocks can be used to exit wait mode.

When the MCU exits wait mode by the peripheral function interrupts or $\overline{\text{NMI}}$ interrupt, the CPU clock does not change before and after the WAIT instruction is executed.

Table 9.8 lists interrupts to be used to exit wait mode and usage conditions.

Table 9.8 Interrupts to Exit Wait Mode and Usage Conditions

Interrupt	When CM02 = 0	When CM02 = 1
$\overline{\text{NMI}}$ interrupt	Available	Available
Voltage monitor interrupt	Available	Available
Serial interface interrupt	Available when the source clock is the internal clock or external clock.	Available when the source clock is the external clock or f2n (when fXIND or on-chip oscillator clock is selected).
Key input interrupt	Available	Available
A/D conversion interrupt	Available in one-shot mode or single-sweep mode	Not available
Timer A interrupt Timer B interrupt	Available in all modes	Available in event counter mode or when the count source is fC32 or f2n (when fXIND or on-chip oscillator clock is selected)
$\overline{\text{INT}}$ interrupt	Available	Available

9.5.3 Stop Mode

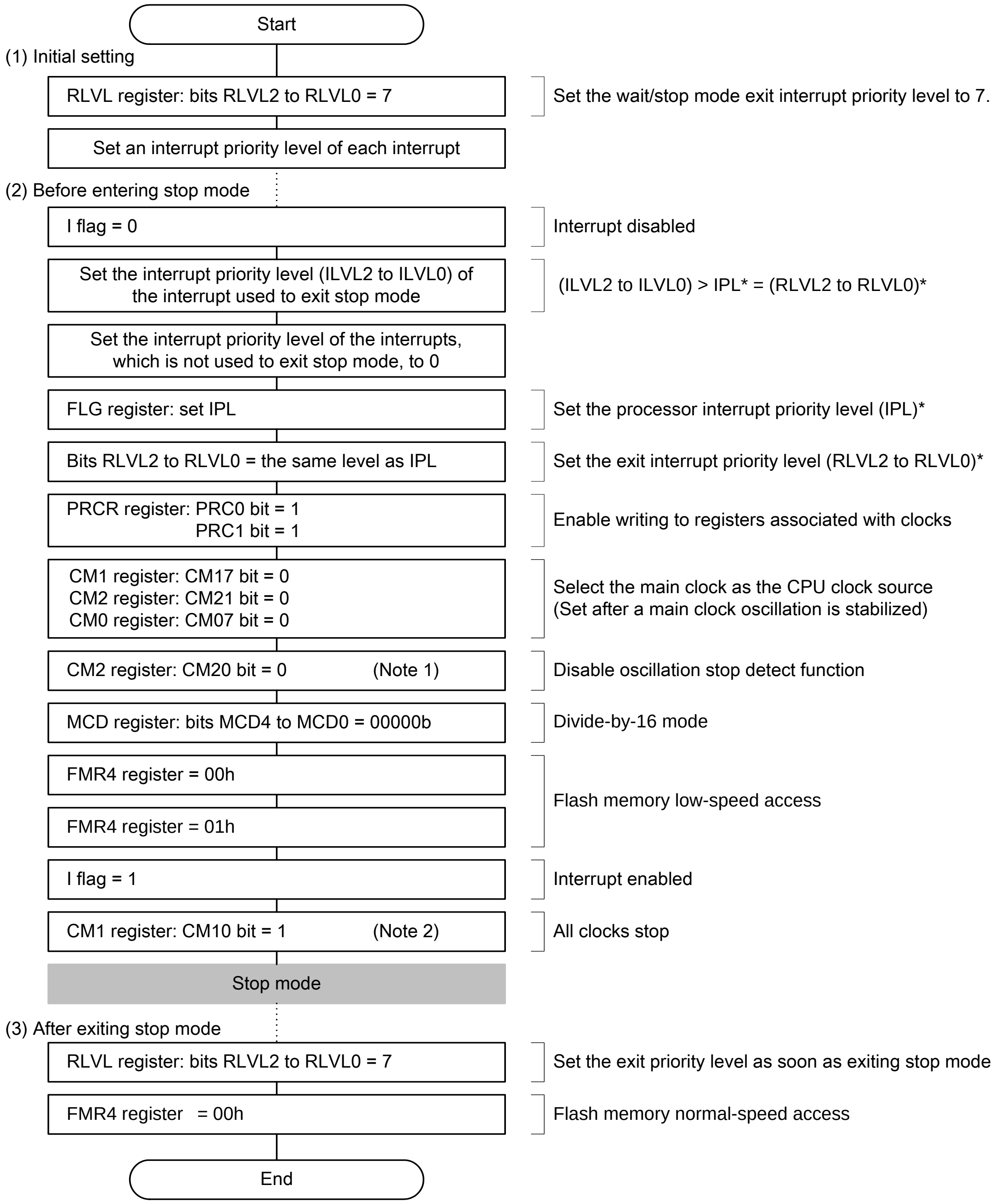
In stop mode, all clocks are stopped. Since the CPU clock and peripheral function clocks are stopped, the CPU and the peripheral functions which are operated by these clocks stop their operation. The least power is required to operate the MCU in stop mode. Enter stop mode from main clock mode.

9.5.3.1 Entering Stop Mode

Stop mode is entered by setting the CM10 bit in the CM1 register to 1 (all clocks stop) while the $\overline{\text{NMI}}$ pin is held "H". Also, bits MCD4 to MCD0 in the MCD register become 01000b (divide-by-8 mode) by setting the CM10 bit to 1. To enter stop mode, the MRS bit in the VRCR register is set to 0 (main voltage regulator operates).
Figure 9.18 shows a procedure to enter stop mode.

When entering stop mode, the instructions following CM10 = 1 instruction are stored into the instruction queue, and the program stops. When stop mode is exited, the instruction lined in the queue is executed before the exit interrupt routine is handled.
Insert the jmp.b instruction as follows after the instruction to set the CM10 bit to 1.

```
            fset I              ; I flag is set to 1
            bset 0,   cm1       ; all clocks stopped (stop mode)
            jmp.b LABEL_001     ; jmp.b instruction executed (no instruction between jmp.b and LABEL.)
    LABEL_001:
            nop                 ; nop(1)
            nop                 ; nop(2)
            nop                 ; nop(3)
            nop                 ; nop(4)
            mov.b #0, prcr      ; protection set
                    .
                    .
                    .
```


Start
(1) Initial setting
RLVL register: bits RLVL2 to RLVL0 = 7
Set the wait/stop mode exit interrupt priority level to 7.
Set an interrupt priority level of each interrupt
(2) Before entering stop mode
I flag = 0
Interrupt disabled
Set the interrupt priority level (ILVL2 to ILVL0) of the interrupt used to exit stop mode
(ILVL2 to ILVL0) > IPL* = (RLVL2 to RLVL0)*
Set the interrupt priority level of the interrupts, which is not used to exit stop mode, to 0
FLG register: set IPL
Set the processor interrupt priority level (IPL)*
Bits RLVL2 to RLVL0 = the same level as IPL
Set the exit interrupt priority level (RLVL2 to RLVL0)*
PRCR register: PRC0 bit = 1
PRC1 bit = 1
Enable writing to registers associated with clocks
CM1 register: CM17 bit = 0
CM2 register: CM21 bit = 0
CM0 register: CM07 bit = 0
Select the main clock as the CPU clock source
(Set after a main clock oscillation is stabilized)
CM2 register: CM20 bit = 0 (Note 1)
Disable oscillation stop detect function
MCD register: bits MCD4 to MCD0 = 00000b
Divide-by-16 mode
FMR4 register = 00h
FMR4 register = 01h
Flash memory low-speed access
I flag = 1
Interrupt enabled
CM1 register: CM10 bit = 1 (Note 2)
All clocks stop
Stop mode
(3) After exiting stop mode
RLVL register: bits RLVL2 to RLVL0 = 7
Set the exit priority level as soon as exiting stop mode
FMR4 register = 00h
Flash memory normal-speed access
End

NOTES:
1. This setting is required when the oscillation stop detect function is used.
2. Insert the jmp.b instruction as follows after the instruction to set the CM10 bit to 1.

```
          bset 0,     cm1         ; all clocks stopped (stop mode)
          jmp.b LABEL_001         ; jmp.b instruction executed (no instruction
     LABEL_001:                   ; between jmp.b and LABEL.)
          nop                     ; nop(1)
          nop                     ; nop(2)
          nop                     ; nop(3)
          nop                     ; nop(4)
          mov.b #0, prcr          ; protection set
                    .
                    .
                    .
```

Figure 9.18 Procedure to Enter Stop Mode

9.5.3.2 Pin States in Stop Mode

Table 9.9 lists pin states in stop mode.

Table 9.9 Pin States in Stop Mode

Pin		Memory Expansion Mode Microprocessor Mode	Single-Chip Mode
Address Bus, Data Bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BHE}}$		Maintain the state immediately before entering stop mode	
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$		“H”	
$\overline{\text{HLDA}}$, BCLK		“H”	
ALE		“H”	
Ports		Maintain the state immediately before entering stop mode	
CLKOUT	When fC is selected	“H”	
	When f8, f32 are selected	The clock is stopped and holds the level immediately before entering stop mode	
XIN		Placed in a high-impedance state	
XOUT		“H”	
XCIN, XCOUT		Placed in a high-impedance state	

9.5.3.3 Exiting Stop Mode

Stop mode is exited by the hardware reset, $\overline{\text{NMI}}$ interrupt, voltage monitor interrupt, or peripheral function interrupts. The following are the peripheral function interrupts that can be used to exit stop mode.

- Key input interrupt
- $\overline{\text{INT}}$ interrupt
- Timer A and timer B interrupts
  (Available when the timer counts external pulse having 100-Hz frequency or lower in event counter mode)

When only the hardware reset, $\overline{\text{NMI}}$ interrupt, or voltage monitor interrupt is used to exit stop mode, set bits ILVL2 to ILVL0 in the Interrupt Control Registers for all the peripheral function interrupts to 000b (interrupt disabled) before setting the CM10 bit in the CM1 register to 1 (all clocks stop).
If the voltage applied to pins VCC1 and VCC2 drops below 3.0 V in stop mode, exit stop mode by the hardware reset after the voltage has satisfied the recommended operating conditions.

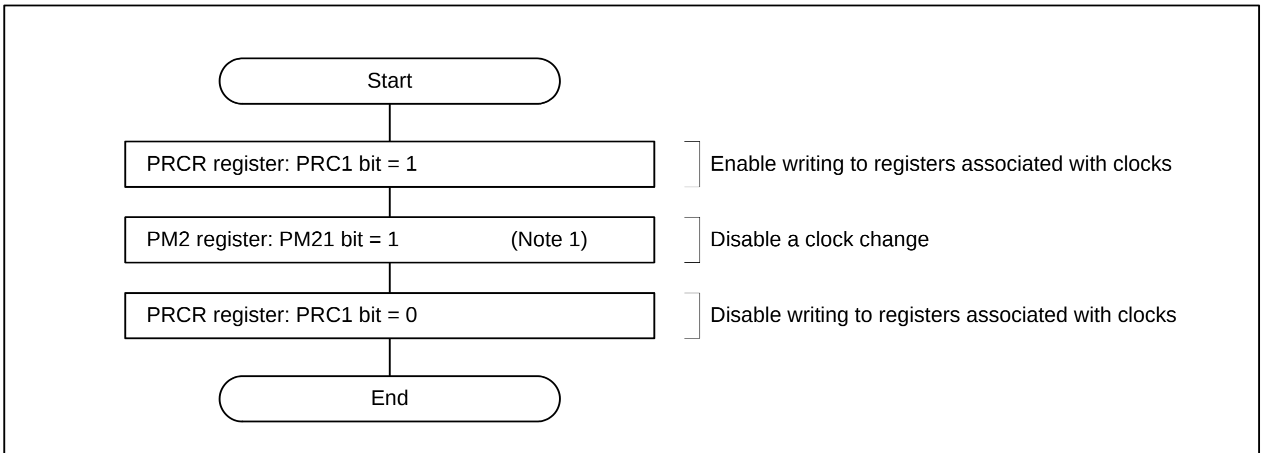
9.6 System Clock Protect Function

The system clock protect function prohibits the clock setting from being rewritten in order to prevent the CPU clock source from being changed when a program goes out of control.
When the PM21 bit in the PM2 register is set to 1 (disables a clock change), the following bits cannot be written:

- Bits CM02, CM05, and CM07 in the CM0 register
- Bits CM10 and CM17 in the CM1 register
- The CM20 bit in the CM2 register
- All bits in the PLC0 register

The CPU clock continues running when the WAIT instruction is executed.

Figure 9.19 shows a procedure to use the system clock protect function. Follow the procedure while the CM05 bit in the CM0 register is set to 0 (main clock oscillates) and the CM07 bit to 0 (main clock as CPU clock source).


Start
PRCR register: PRC1 bit = 1
Enable writing to registers associated with clocks
PM2 register: PM21 bit = 1 (Note 1)
Disable a clock change
PRCR register: PRC1 bit = 0
Disable writing to registers associated with clocks
End

NOTE:
1. When entering wait mode, execute the WAIT instruction while the PM21 bit in the PM2 register is set to 0.

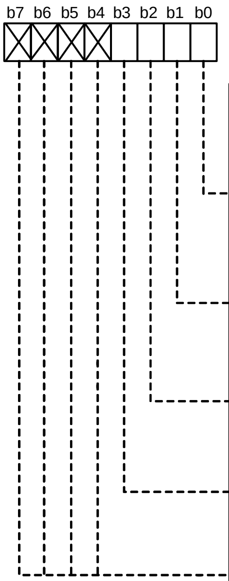
Figure 9.19 Procedure to Use System Clock Protect Function

10. Protection

The function protects important registers from being inadvertently overwritten in case of a program crash. Figure 10.1 shows the PRCR register.

The PRC2 bit in the PRCR register becomes 0 (write disable) by a write to the SFR area after the PRC2 bit is set to 1 (write enable). Set the PD9 or PS3 register immediately after the PRC2 bit is set to 1. Do not generate an interrupt or a DMA or DMACII transfer between these two instructions. Bits PRC0, PRC1, and PRC3 do not become 0 automatically even after a write to the SFR area. Set bits PRC0, PRC1, and PRC3 to 0 by a program.

Protect Register



Symbol	Address	After Reset
PRCR	000Ah	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
PRC0	Protect bit 0[1]	Writing to registers CM0, CM1, CM2, MCD, and PLC0 is enabled 0: Write disable 1: Write enable	RW
PRC1	Protect bit 1[1]	Writing to registers PM0, PM1, PM2, INVC0, and INVC1 is enabled 0: Write disable 1: Write enable	RW
PRC2	Protect bit 2[2]	Writing to registers PD9 and PS3 is enabled 0: Write disable 1: Write enable	RW
PRC3	Protect bit 3[1]	Writing to registers DVCR, LVDC, and VRCR is enabled 0: Write disable 1: Write enable	RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

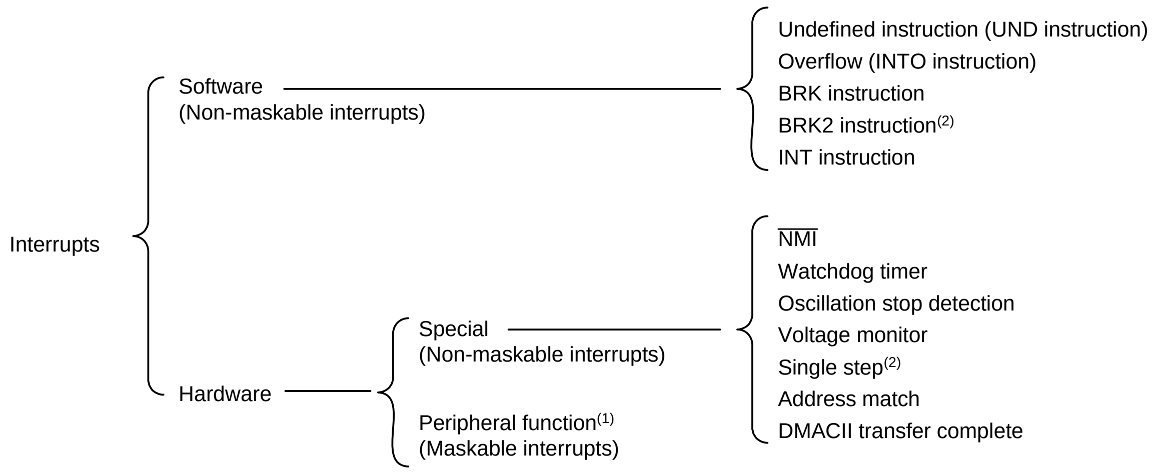
NOTES:
1. Bits PRC0, PRC1, and PRC3 do not become 0 automatically even after a write to the SFR area. Set bits PRC0, PRC1, and PRC3 to 0 by a program.
2. The PRC2 bit becomes 0 by a write to the SFR area after the PRC2 bit is set to 1.

Figure 10.1 PRCR Register

11. Interrupts

11.1 Types of Interrupts

Figure 11.1 shows the types of interrupts.



NOTES:
1. Peripheral function interrupts are generated by the on-chip peripheral functions in the MCU.
2. Do not use these interrupts. They are for use with development tool only.

Figure 11.1 Interrupts

- Maskable interrupts
  The I flag and IPL can enable and disable these interrupts.
  The interrupt priority order can be changed by using interrupt priority level settings.
- Non-maskable interrupt
  These interrupts cannot be disabled regardless of the I flag and IPL settings.

11.2 Software Interrupts

Software interrupts occur when particular instructions are executed. Software interrupts are non-maskable.

11.2.1 Undefined Instruction Interrupt

The undefined instruction interrupt occurs when the UND instruction is executed.

11.2.2 Overflow Interrupt

The overflow interrupt occurs when the INTO instruction is executed while the O flag in the FLG register is 1 (arithmetic operation overflow). Instructions that can set the O flag are: ABS, ADC, ADCF, ADD, ADDX, CMP, CMPX, DIV, DIVU, DIVX, NEG, RMPA, SBB, SCMPU, SHA, SUB, SUBX

11.2.3 BRK Interrupt

The BRK interrupt occurs when the BRK instruction is executed.

11.2.4 BRK2 Interrupt

The BRK2 interrupt occurs when the BRK2 instruction is executed.
Do not use this interrupt. This is for use with development support tool only.

11.2.5 INT Instruction Interrupt

The INT instruction interrupt occurs when the INT instruction is executed. The INT instruction can specify software interrupt numbers 0 to 63. Software interrupt numbers 8 to 43 are assigned to the vector table used for the peripheral function interrupt. This means that the MCU is able to execute the peripheral function interrupt routine by executing the INT instruction. When the INT instruction is executed, values in the FLG register and PC are saved to the stack. The relocatable vector of the specified software interrupt number is stored in PC.
The stack, where the data is saved, varies depending on a software interrupt number.
ISP is selected for software interrupt numbers 0 to 31. (The U flag in the FLG register becomes 0.) For software interrupt numbers 32 to 63, SP which is selected immediately before executing the INT instruction is used. (The U flag does not change.)
For the peripheral function interrupt, the FLG register value is saved and the U flag becomes 0 (ISP selected) when an interrupt request is acknowledged. Therefore, for software interrupt numbers 32 to 43, SP to be used can differ depending on whether an interrupt is generated by a peripheral function or by the INT instruction.

11.3 Hardware Interrupts

Special interrupts and peripheral function interrupts are available as hardware interrupts.

11.3.1 Special Interrupts

Special interrupts are non-maskable.

11.3.1.1 $\overline{\text{NMI}}$ Interrupt

The $\overline{\text{NMI}}$ interrupt occurs when a signal applied to the $\overline{\text{NMI}}$ pin changes from high level (“H”) to low level (“L”). Refer to **11.8 $\overline{\text{NMI}}$ Interrupt** for details.

11.3.1.2 Watchdog Timer Interrupt

The watchdog timer interrupt occurs when the watchdog timer counter underflows. Refer to **12. Watchdog Timer** for details.

11.3.1.3 Oscillation Stop Detection Interrupt

The oscillation stop detection interrupt occurs when the MCU detects a loss of the main clock. Refer to **9. Clock Generation Circuits** for details.

11.3.1.4 Voltage Monitor Interrupt

The voltage monitor interrupt occurs when voltage monitor function detects the changes in voltage. Refer to **6. Power Supply Voltage Monitor Function** for details.

11.3.1.5 Single-Step Interrupt

Do not use the single-step interrupt. This is for use with development support tool only.

11.3.1.6 Address Match Interrupt

When the AIERi bit in the AIER register is set to 1 (address match interrupt enabled), the address match interrupt occurs immediately before executing the instruction stored in the address indicated by the RMADi register (i = 0 to 7).
Set the starting address of the instruction in the RMADi register. The address match interrupt does not occur if a table data or any address other than the starting address of the instruction is set. Refer to **11.10 Address Match Interrupt** for details.

11.3.1.7 DMACII End-of-Transfer Complete Interrupt

The DMACII transfer complete interrupt is generated by the DMACII function. Refer to **14. DMACII** for details.

11.3.2 Peripheral Function Interrupt

The peripheral function interrupt is generated by the on-chip peripheral functions. The peripheral function interrupts and software interrupt numbers 8 to 43 for the INT instruction use the same interrupt vector table. The peripheral function interrupt is maskable.
See **Tables 11.2 and 11.3** for the peripheral function interrupt sources. Refer to the descriptions of individual peripheral functions for details.

11.4 High-Speed Interrupt

The high-speed interrupt executes an interrupt sequence in five cycles and returns from the interrupt routine in three cycles. When the FSIT bit in the RLVL register is set to 1 (interrupt priority level 7 is used for the high-speed interrupt), the interrupt that bits ILVL2 to ILVL0 in the Interrupt Control Register are set to 111b (level 7) becomes the high-speed interrupt.

Only one interrupt can be set as the high-speed interrupt. To use the high-speed interrupt, do not set multiple interrupts to interrupt priority level 7. Set the DMAII bit in the RLVL register to 0 (interrupt priority level 7 is used for interrupt) to use the high-speed interrupt.

Set the starting address of a high-speed interrupt routine in the VCT register.

When the high-speed interrupt is acknowledged, the FLG register value is saved into the SVF register and the PC value is saved into the SVP register. A program is executed from an address indicated by the VCT register.

Use the FREIT instruction to return from a high-speed interrupt routine. Values saved into registers SVF and SVP are restored to the FLG register and PC by executing the FREIT instruction.

The high-speed interrupt, and DMA2 and DMA3 share some of the registers. When using the high-speed interrupt, neither DMA2 nor DMA3 is available. DMA0 and DMA1 can still be used.

Figure 11.2 shows a procedure to use high-speed interrupt.

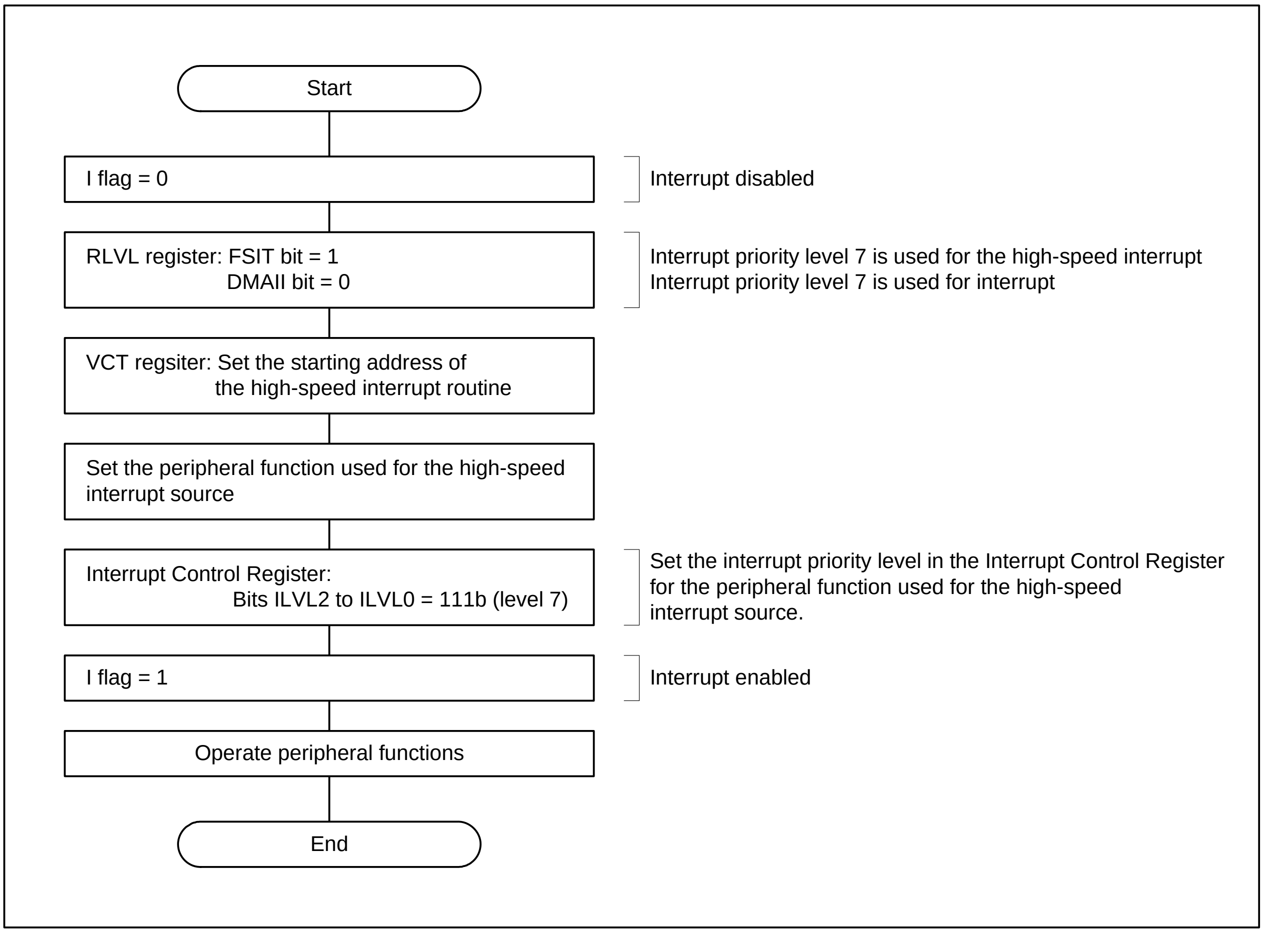

Start
I flag = 0
Interrupt disabled
RLVL register: FSIT bit = 1
DMAII bit = 0
Interrupt priority level 7 is used for the high-speed interrupt
Interrupt priority level 7 is used for interrupt
VCT regsiter: Set the starting address of the high-speed interrupt routine
Set the peripheral function used for the high-speed interrupt source
Interrupt Control Register: Bits ILVL2 to ILVL0 = 111b (level 7)
Set the interrupt priority level in the Interrupt Control Register for the peripheral function used for the high-speed interrupt source.
I flag = 1
Interrupt enabled
Operate peripheral functions
End

Figure 11.2 Procedure to Use High-Speed Interrupt

11.5 Interrupts and Interrupt Vectors

There are four bytes in each interrupt vector. Set the starting address of an interrupt routine in each interrupt vector. When an interrupt request is acknowledged, an interrupt routine is executed from the address set in its interrupt vector. Figure 11.3 shows an interrupt vector.

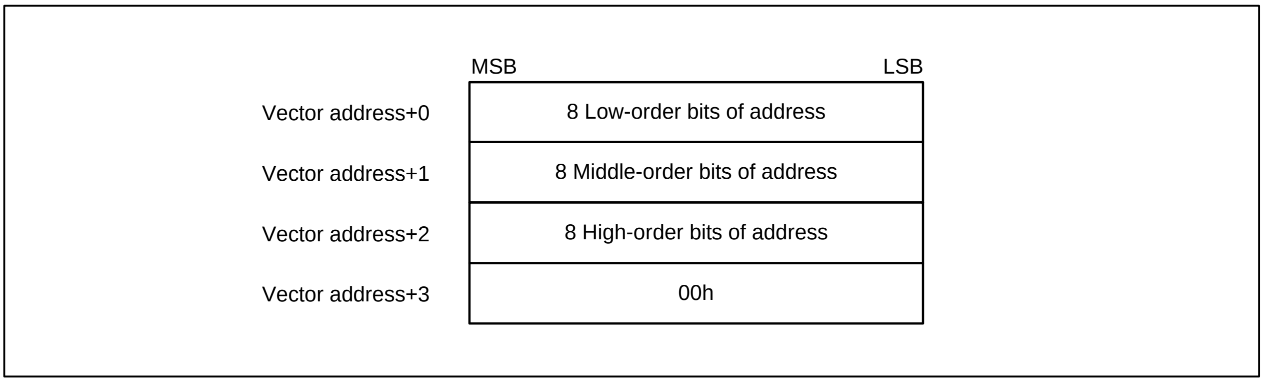

MSB
LSB
Vector address+0
8 Low-order bits of address
Vector address+1
8 Middle-order bits of address
Vector address+2
8 High-order bits of address
Vector address+3
00h

Figure 11.3 Interrupt Vector

11.5.1 Fixed Vector Table

The fixed vector table is allocated in addresses FFFFDCh to FFFFFFh. Table 11.1 lists the fixed vector table. The ID code which is used for the ID code check function of the flash memory is stored to the part of the fixed vector table. Refer to **23.3.3 ID Code Check Function** for details.

Table 11.1 Fixed Vector Table

Interrupt Source	Vector Addresses Address (L) to Address (H)	Remarks	Reference
Undefined instruction	FFFFDCh to FFFFDFh		M32C/80 series software manual
Overflow	FFFFE0h to FFFFE3h		
BRK instruction	FFFFE4h to FFFFE7h	If the content of the address FFFFE7h is FFh, the CPU executes from the address stored in the software interrupt number 0 in the relocatable vector table.	
Address match	FFFFE8h to FFFFEBh		
–	FFFFECh to FFFFEFh	Reserved space	
Watchdog timer	FFFFF0h to FFFFF3h	These addresses are used for the watchdog timer interrupt, oscillation stop detection interrupt, and voltage monitor interrupt.	Voltage monitor function, Clock generation circuit, Watchdog timer
–	FFFFF4h to FFFFF7h	Reserved space	
$\overline{\text{NMI}}$	FFFFF8h to FFFFFBh		
Reset	FFFFFCh to FFFFFFh		Reset

11.5.2 Relocatable Vector Table

The relocatable vector table occupies 256 bytes beginning from the address set in the INTB register. Tables 11.2 and 11.3 list the relocatable vector table.

Set an even address to the starting address of the vector set in the INTB register to increase the interrupt sequence execution rate.

Table 11.2 Relocatable Vector Tables (1/2)

Interrupt Source	Vector Table Address Address (L) to Address (H)[1]	Software Interrupt Number	Reference
BRK instruction[2]	+0 to +3 (0000h to 0003h)	0	M32C/80 Series Software Manual
Reserved space	+4 to +31 (0004h to 001Fh)	1 to 7	
DMA0	+32 to +35 (0020h to 0023h)	8	DMAC
DMA1	+36 to +39 (0024h to 0027h)	9	
DMA2	+40 to +43 (0028h to 002Bh)	10	
DMA3	+44 to +47 (002Ch to 002Fh)	11	
Timer A0	+48 to +51 (0030h to 0033h)	12	Timer A
Timer A1	+52 to +55 (0034h to 0037h)	13	
Timer A2	+56 to +59 (0038h to 003Bh)	14	
Timer A3	+60 to +63 (003Ch to 003Fh)	15	
Timer A4	+64 to +67 (0040h to 0043h)	16	
UART0 transmission, NACK[3]	+68 to +71 (0044h to 0047h)	17	Serial interfaces
UART0 reception, ACK[3]	+72 to +75 (0048h to 004Bh)	18	
UART1 transmission, NACK[3]	+76 to +79 (004Ch to 004Fh)	19	
UART1 reception, ACK[3]	+80 to +83 (0050h to 0053h)	20	
Timer B0	+84 to +87 (0054h to 0057h)	21	Timer B
Timer B1	+88 to +91 (0058h to 005Bh)	22	
Timer B2	+92 to +95 (005Ch to 005Fh)	23	
Timer B3	+96 to +99 (0060h to 0063h)	24	
Timer B4	+100 to +103 (0064h to 0067h)	25	
$\overline{\text{INT5}}$	+104 to +107 (0068h to 006Bh)	26	Interrupts
$\overline{\text{INT4}}$	+108 to +111 (006Ch to 006Fh)	27	
$\overline{\text{INT3}}$	+112 to +115 (0070h to 0073h)	28	
$\overline{\text{INT2}}$	+116 to +119 (0074h to 0077h)	29	
$\overline{\text{INT1}}$	+120 to +123 (0078h to 007Bh)	30	
$\overline{\text{INT0}}$	+124 to +127 (007Ch to 007Fh)	31	
Timer B5	+128 to +131 (0080h to 0083h)	32	Timer B
UART2 transmission, NACK[3]	+132 to +135 (0084h to 0087h)	33	Serial interfaces
UART2 reception, ACK[3]	+136 to +139 (0088h to 008Bh)	34	
UART3 transmission, NACK[3]	+140 to +143 (008Ch to 008Fh)	35	
UART3 reception, ACK[3]	+144 to +147 (0090h to 0093h)	36	
UART4 transmission, NACK[3]	+148 to +151 (0094h to 0097h)	37	
UART4 reception, ACK[3]	+152 to +155 (0098h to 009Bh)	38	

NOTES:

1. The numbers represent address offsets from the base address set in the INTB register.
2. The I flag can not disable this interrupt.
3. In I^2C mode, NACK, ACK, or start/stop condition detection can be the interrupt sources.

Table 11.3 Relocatable Vector Tables (2/2)

Interrupt Source	Vector Table Address Address (L) to Address (H)(1)	Software Interrupt Number	Reference
Bus conflict detection, Start condition detection/ Stop condition detection (UART2)(3)	+156 to +159 (009Ch to 009Fh)	39	Serial interfaces
Bus conflict detection, Start condition detection/ Stop condition detection (UART3 or UART0)(4)	+160 to +163 (00A0h to 00A3h)	40	
Bus conflict detection, Start condition detection/ Stop condition detection (UART4 or UART1)(4)	+164 to +167 (00A4h to 00A7h)	41	
A/D0	+168 to +171 (00A8h to 00ABh)	42	A/D converter
Key input	+172 to +175 (00ACh to 00AFh)	43	Interrupts
Reserved space	+176 to +255 (00B0h to 00FFh)	44 to 63	–
INT instruction(2)	+0 to +3 (0000h to 0003h) to +252 to +255 (00FCh to 00FFh)	0 to 63	Interrupts

NOTES:
1. The numbers represent address offsets from the base address set in the INTB register.
2. The I flag can not disable this interrupt.
3. In I^2C mode, NACK, ACK, or start/stop condition detection can be the interrupt sources.
4. The IFSR6 bit in the IFSR register selects either UART0 or UART3. The IFSR7 bit selects either UART1 or UART4.

11.6 Interrupt Request Acknowledgement

Software interrupts occur when their corresponding instructions are executed. The INTO instruction, however, requires the O flag in the FLG register to be 1. Special interrupts occur when their corresponding interrupt requests are generated.

For the peripheral function interrupts to be acknowledged, the following conditions must be met:

- I flag = 1
- IR bit = 1
- Bits ILVL2 to ILVL0 > IPL

The I flag, IPL, IR bit, and bits ILVL2 to ILVL0 are independent of each other. The I flag and IPL are in the FLG register. The IR bit and bits ILVL2 to ILVL0 are in the Interrupt Control Register.

11.6.1 I Flag and IPL

The I flag enables and disables maskable interrupts. When the I flag is set to 1 (enable), all maskable interrupts are enabled; when the I flag is set to 0 (disable), they are disabled. The I flag automatically becomes 0 after reset.

IPL is 3 bits wide and indicates the Interrupt Priority Level (IPL) from level 0 to level 7. If a requested interrupt has higher priority level than IPL, the interrupt is acknowledged.

Table 11.4 lists interrupt priority levels associated with IPL.

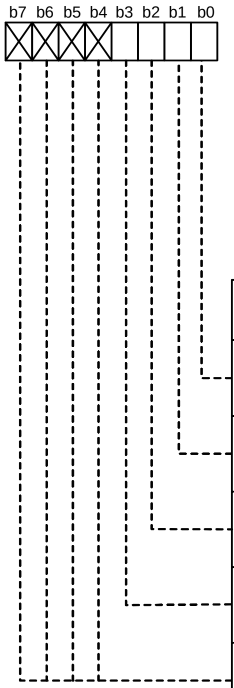
Table 11.4 Interrupt Priority Levels

IPL2 to IPL0	Required Interrupt Priority Levels to Be Acknowledged for Maskable Interrupts
0	Level 1 and above
1	Level 2 and above
2	Level 3 and above
3	Level 4 and above
4	Level 5 and above
5	Level 6 and above
6	Level 7 and above
7	All maskable interrupts are disabled

11.6.2 Interrupt Control Registers and RLVL Register

The Interrupt Control Registers are used to control the peripheral function interrupts. Figures 11.4 and 11.5 show the Interrupt Control Registers. Figure 11.6 shows the RLVL register.

Interrupt Control Register



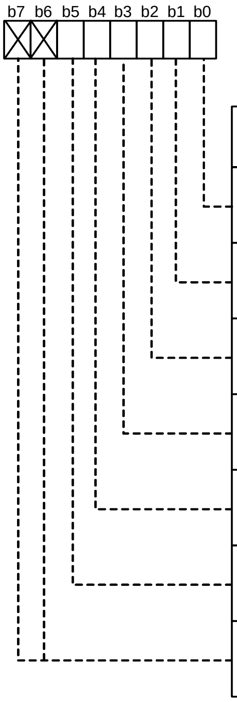
Symbol	Address	After Reset
TA0IC to TA4IC	006Ch, 008Ch, 006Eh, 008Eh, 0070h	XXXX X000b
TB0IC to TB5IC	0094h, 0076h, 0096h, 0078h, 0098h, 0069h	XXXX X000b
S0TIC to S4TIC	0090h, 0092h, 0089h, 008Bh, 008Dh	XXXX X000b
S0RIC to S4RIC	0072h, 0074h, 006Bh, 006Dh, 006Fh	XXXX X000b
BCN0IC to BCN4IC	0071h, 0091h, 008Fh, 0071h(1), 0091h(2)	XXXX X000b
DM0IC to DM3IC	0068h, 0088h, 006Ah, 008Ah	XXXX X000b
AD0IC	0073h	XXXX X000b
KUPIC	0093h	XXXX X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bits	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(3)	0: Interrupt not requested 1: Interrupt requested	RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. The BCN0IC register shares the address with the BCN3IC register.
2. The BCN1IC register shares the address with the BCN4IC register.
3. The IR bit can be set to 0 only. Do not set to 1.

Figure 11.4 Interrupt Control Register (1/2)

Interrupt Control Register



Symbol	Address	After Reset
INT0IC to INT2IC	009Eh, 007Eh, 009Ch	XX00 X000b
INT3IC to INT5IC[1]	007Ch, 009Ah, 007Ah	XX00 X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bits	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[2]	0: Interrupt not requested 1: Interrupt requested	RW
POL	Polarity switch bit[3]	0: Falling edge / "L" level selected 1: Rising edge / "H" level selected	RW
LVS	Level sensitive/ edge sensitive switch bit[4]	0 : Edge sensitive 1 : Level sensitive	RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:

1. When a 16-bit data bus is used in microprocessor mode and memory expansion mode, pins $\overline{\text{INT3}}$ to $\overline{\text{INT5}}$ are used as data bus. In this case, set bits ILVL2 to ILVL0 in registers INT3IC to INT5IC to 000b.
2. The IR bit can be set to 0 only. Do not set to 1.
3. Set the POL bit to 0 when its corresponding bit in the IFSR register is set to 1 (both edges).
4. When the LVS bit is set to 1, set its corresponding bit in the IFSR register to 0 (one edge).

Figure 11.5 Interrupt Control Register (2/2)

11.6.2.1 Bits ILVL2 to ILVL0

Bits ILVL2 to ILVL0 determine an interrupt priority level. The higher the interrupt priority level is, the higher priority the interrupt has.

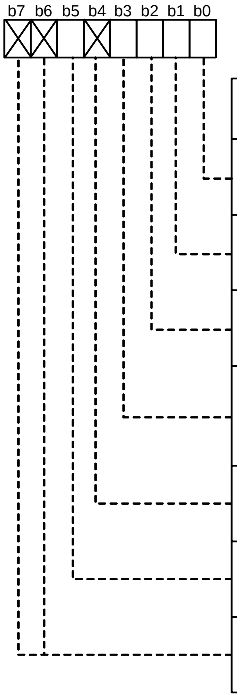
When an interrupt request is generated, its interrupt priority level is compared to IPL. This interrupt is enabled only when its interrupt priority level is higher than IPL. When bits ILVL2 to ILVL0 are set to 000b (level 0), the interrupt is disabled.

11.6.2.2 IR Bit

The IR bit is automatically set to 1 (interrupt requested) by hardware when an interrupt request is generated. After an interrupt request is acknowledged and an interrupt sequence in the corresponding interrupt vector is executed, the IR bit is automatically set to 0 (interrupt not requested) by hardware.

The IR bit can be set to 0 by a program. Do not set it to 1.

Exit Priority Register



Symbol	Address	After Reset
RLVL	009Fh	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
RLVL0	Exit wait mode/stop mode interrupt priority level control bits[1]	b2 b1 b0 0 0 0: Level 0 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
RLVL1			RW
RLVL2			RW
FSIT	High-speed interrupt select bit	0: Interrupt priority level 7 is used for normal interrupt 1: Interrupt priority level 7 is used for high-speed interrupt[2][3]	RW
– (b4)	Unimplemented. Write 0. Read as undefined value.		–
DMAII	DMACII select bit[4]	0: Interrupt priority level 7 is used for interrupt 1: Interrupt priority level 7 is used for DMACII transfer[2]	RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:

1. The MCU exits stop or wait mode when an interrupt priority level of a requested interrupt is higher than a level set using bits RLVL2 to RLVL0. Set bits RLVL2 to RLVL0 to the same value as IPL in the FLG register.
2. Do not set both the FSIT and DMAII bits to 1.
   Set either the FSIT bit or the DMAII bit to 1 before setting bits ILVL2 to ILVL0 in the Interrupt Control Register to 111b.
3. Only one interrupt can have the interrupt priority level 7 when selecting the high-speed interrupt.
4. The DMAII bit is undefined after reset. To use interrupt priority level 7 for an interrupt, set it to 0 before setting the Interrupt Control Register.

Figure 11.6 RLVL Register

11.6.2.3 Bits RLVL2 to RLVL0

When using an interrupt to exit wait mode or stop mode, refer to **9.5.2 Wait Mode** and **9.5.3 Stop Mode** for details.

11.6.3 Interrupt Sequence

The interrupt sequence is performed between an interrupt request acknowledgment and interrupt routine execution.

When an interrupt request is generated while an instruction is being executed, the CPU determines its interrupt priority after the instruction in progress is completed. Then, the CPU starts the interrupt sequence from the following cycle. However, for the SCMPU, SIN, SMOVB, SMOVF, SMOVU, SSTR, SOUT, and RMPA instructions, if an interrupt request is generated while one of these instructions is being executed, the MCU suspends the instruction execution to start the interrupt sequence.

The interrupt sequence is performed as indicated below:

(1) The CPU obtains the interrupt number by reading the address 000000h (address 000002h for the high-speed interrupt). Then, the corresponding IR bit to the interrupt becomes 0 (interrupt not requested).

(2) The FLG register value, immediately before the interrupt sequence, is saved to a temporary register[1] in the CPU.

(3) Each bit in the FLG register becomes as follows:
The I flag becomes 0 (interrupt disabled)
The D flag becomes 0 (single-step interrupt disabled)
The U flag becomes 0 (ISP selected)

(4) The internal register value (the FLG register value saved in (2)) in the CPU is saved to the stack; or to the SVF register for the high-speed interrupt.

(5) The PC value is saved to the stack; or to the SVP register for the high-speed interrupt.

(6) The interrupt priority level of the acknowledged interrupt becomes the IPL level.

(7) An interrupt vector corresponding to the acknowledged interrupt is stored into PC.

After the interrupt sequence is completed, the CPU executes the instruction from the starting address of the interrupt routine.

NOTE:

1. Temporary register cannot be accessed by users.

11.6.4 Interrupt Response Time

Figure 11.7 shows the interrupt response time. Interrupt response time is the period between an interrupt request generation and the end of an interrupt sequence. Interrupt response time is divided into two phases: the period between an interrupt request generation and the end of the ongoing instruction execution ((a) in Figure 11.7), and the period required to perform the interrupt sequence ((b) in Figure 11.7).

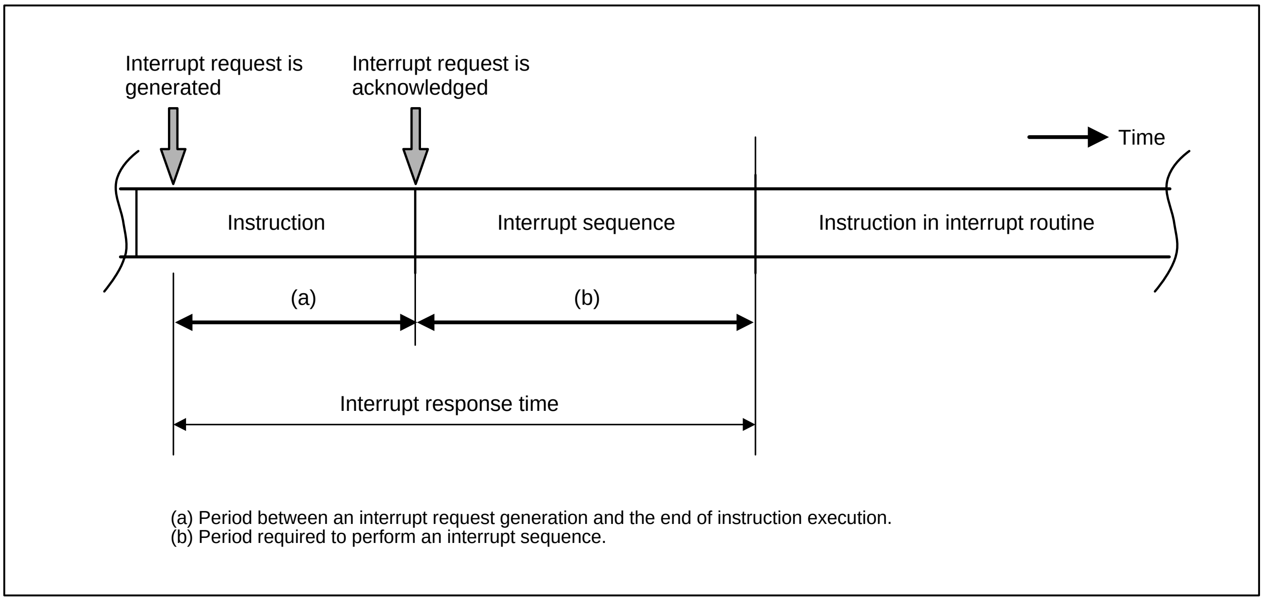

Interrupt request is generated
Interrupt request is acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
(b)
Interrupt response time
(a) Period between an interrupt request generation and the end of instruction execution.
(b) Period required to perform an interrupt sequence.

Figure 11.7 Interrupt Response Time

Time (a) varies depending on an instruction being executed. The DIV, DIVX, and DIVU instructions require the longest time (a), which is at the maximum of 42 cycles.
Table 11.5 lists time (b).

Table 11.5 Interrupt Sequence Execution Time[1]

Interrupts	Execution Time (in terms of CPU clock)
Peripheral function	14 cycles
INT instruction	12 cycles
$\overline{\text{NMI}}$ Watchdog timer Undefined instruction Address match	13 cycles
Overflow	14 cycles
BRK instruction (relocatable vector table)	17 cycles
BRK instruction (fixed vector table)	19 cycles
High-speed interrupt	5 cycles

NOTE:

1. The values when interrupt vectors are allocated in even addresses in the internal ROM, except for the high-speed interrupt.

11.6.5 IPL Change when Interrupt Request is Acknowledged

When a peripheral function interrupt request is acknowledged, the priority level for the acknowledged interrupt becomes the IPL level in the flag register.
Software interrupts and special interrupts have no interrupt priority level. If an interrupt that has no interrupt priority level occurs, the value shown in Table 11.6 becomes the IPL level.

Table 11.6 Interrupts without Interrupt Priority Levels and IPL

Interrupt Source	IPL level
Watchdog timer, $\overline{\text{NMI}}$, oscillation stop detection, voltage monitor, DMACII end-of-transfer interrupt	7
Software, address match	Not changed

11.6.6 Saving a Register

In the interrupt sequence, values of the FLG register and PC are saved to the stack.
Figure 11.8 shows the stack states before and after an interrupt request is acknowledged.
The other necessary registers are saved by a program at the beginning of the interrupt routine. The PUSHM instruction can save multiple registers(1) in the register bank currently used.
Refer to **11.4 High-Speed Interrupt** for the high-speed interrupt.

NOTE:
1. Selectable from registers R0, R1, R2, R3, A0, A1, SB, and FB.

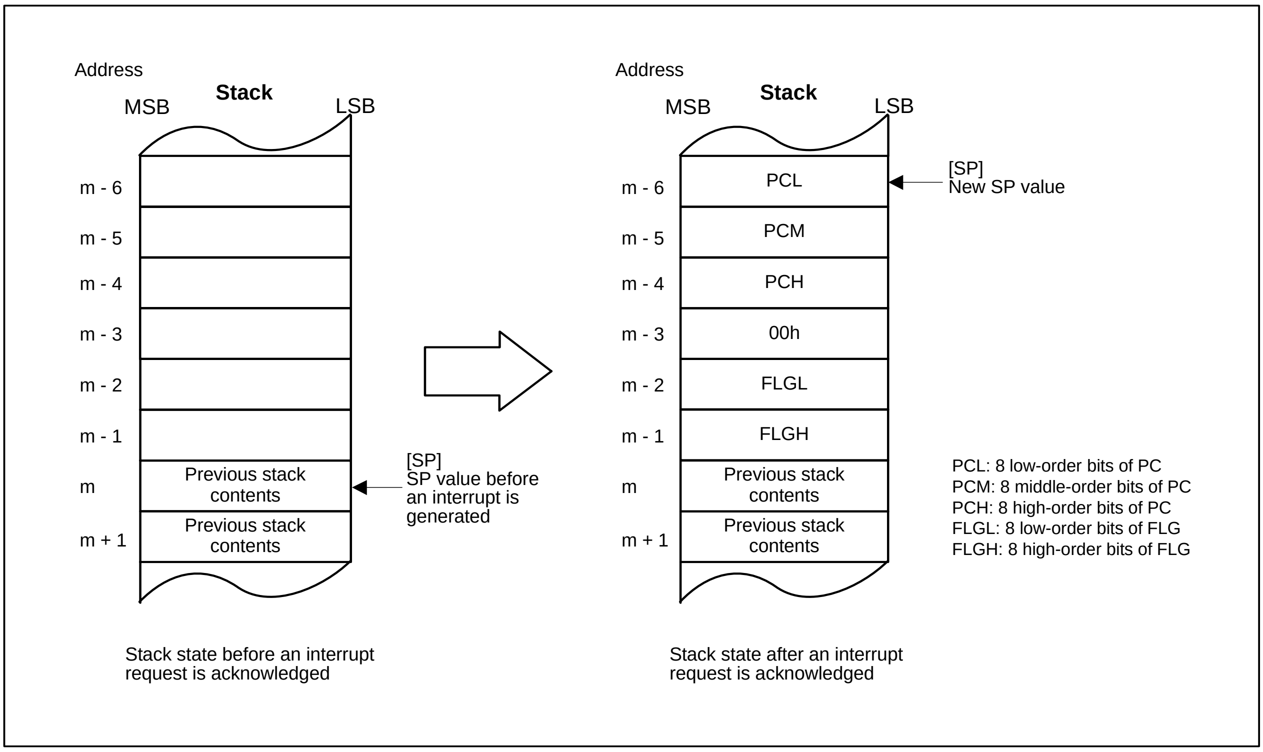

Address
Stack
MSB
LSB
m - 6
m - 5
m - 4
m - 3
m - 2
m - 1
m
m + 1
Previous stack contents
Previous stack contents
[SP]
SP value before an interrupt is generated
Stack state before an interrupt request is acknowledged
Address
Stack
MSB
LSB
m - 6
m - 5
m - 4
m - 3
m - 2
m - 1
m
m + 1
PCL
PCM
PCH
00h
FLGL
FLGH
Previous stack contents
Previous stack contents
[SP]
New SP value
PCL: 8 low-order bits of PC
PCM: 8 middle-order bits of PC
PCH: 8 high-order bits of PC
FLGL: 8 low-order bits of FLG
FLGH: 8 high-order bits of FLG
Stack state after an interrupt request is acknowledged

Figure 11.8 Stack States Before and After Acknowledgement of Interrupt Request

11.6.7 Returning from Interrupt Routine

When the REIT instruction is executed at the end of an interrupt routine, the values of the FLG register and PC, which have been saved to the stack before the interrupt sequence is performed, are automatically restored. And then, the program that was running before an interrupt request was acknowledged, resumes its process. The high-speed interrupt uses the FREIT instruction instead. Refer to **11.4 High-Speed Interrupt** for details.

Before executing the REIT or FREIT instruction, use the POPM instruction or the like to restore registers saved by a program in the interrupt routine. By executing the REIT or FREIT instruction, register bank is switched back to the bank used immediately before the interrupt sequence.

11.6.8 Interrupt Priority

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

Set bits ILVL2 to ILVL0 in the Interrupt Control Register to select the given priority level for maskable interrupts (peripheral function interrupts).

Priority levels of special interrupts, such as $\overline{\text{NMI}}$ and watchdog timer interrupt are fixed by hardware. Figure 11.9 shows the priority of hardware interrupts.

The interrupt priority does not affect software interrupts. Executing an instruction for a software interrupt causes the MCU to execute an interrupt routine.

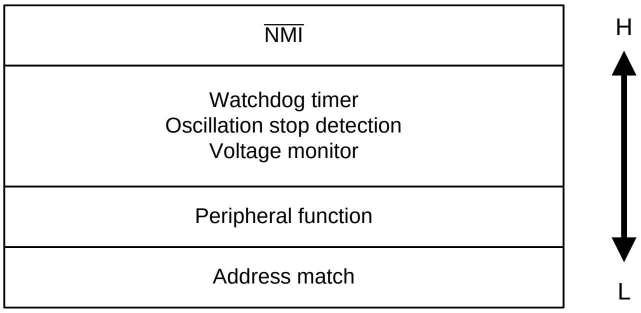


Figure 11.9 Interrupt Priority of Hardware Interrupts

11.6.9 Interrupt Priority Level Decision Circuit

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

Figure 11.10 shows the interrupt priority level decision circuit.

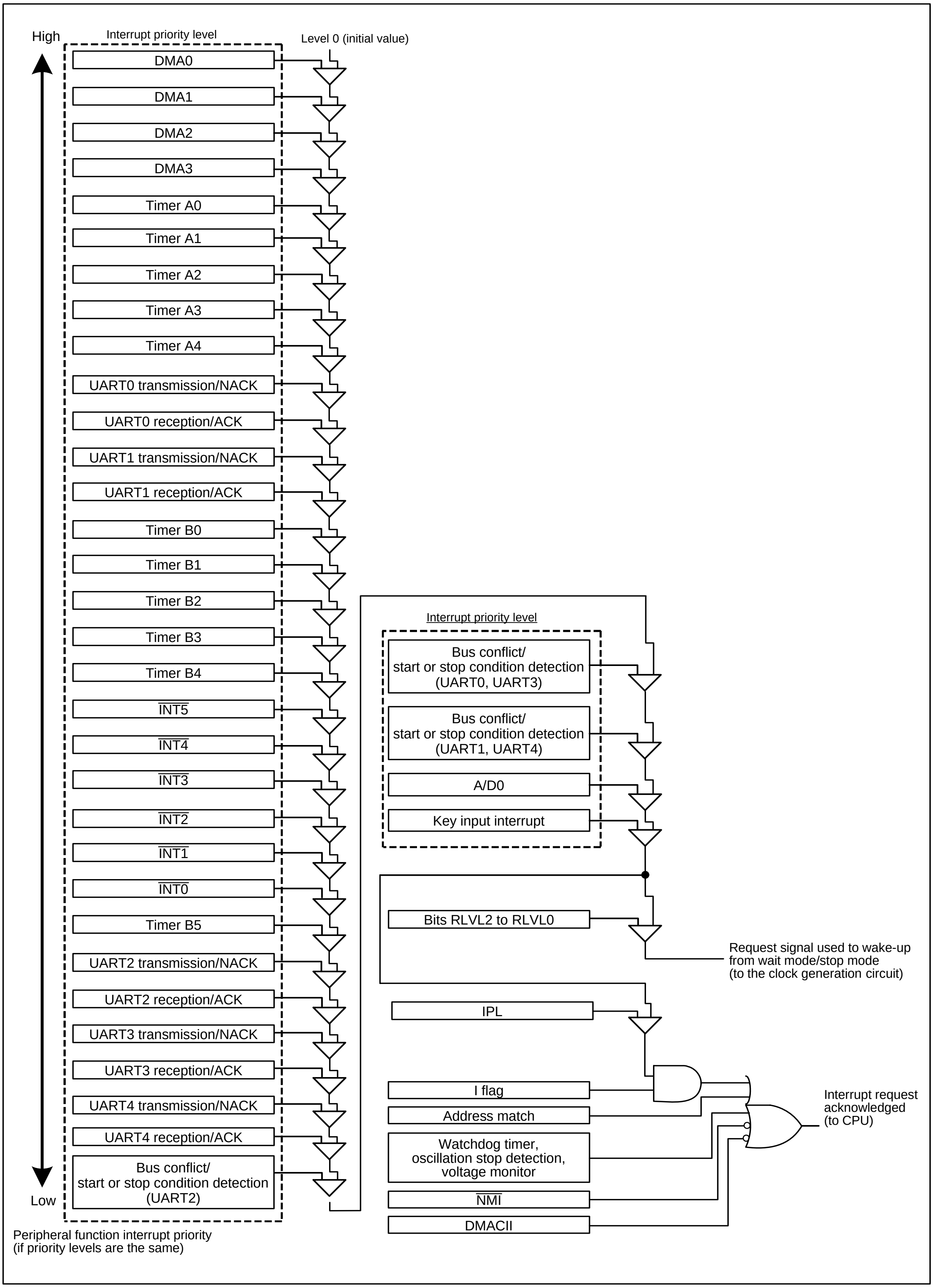


Figure 11.10 Interrupt Priority Level Decision Circuit

11.7 INT Interrupt

External input to pins INT0 to INT5 generates the INT0 to INT5 interrupt. INT0 to INT5 interrupts can select either edge sensitive, which the rising/falling edge triggers an interrupt request, or level sensitive, which an input signal level to the INTi pin (i = 0 to 5) triggers an interrupt request.

To use INT0 to INT5 interrupts with edge sensitive, set the LVS bit in the INTiIC register to 0 (edge sensitive), and select a rising edge, falling edge, or both edges using the POL bit in the INTiIC register and the IFSRi bit in the IFSR register. When the IFSRi bit is set to 1 (both edges), set the corresponding POL bit to 0 (falling edge). When the selected edge is detected at the INTi pin, the corresponding IR bit becomes 1.

To use INT0 to INT5 interrupts with level sensitive, set the LVS bit to 1 (level sensitive) and select either “L” level or “H” level using the POL bit. Also, set the IFSRi bit to 0 (one edge). While the selected level is detected at the INTi pin, the IR bit becomes 1 and remains 1. Therefore, the interrupt requests are generated repeatedly as long as the selected level is detected at the INTi pin. When the input signal is changed to the inactive level, the IR bit becomes 0 by the interrupt request acknowledgement or writing a 0 by a program.

Interrupts can be enabled or disabled using bits ILVL2 to ILVL0 in the INTiIC register.

Figure 11.11 shows INTi interrupt setting procedures (i = 0 to 5). Figure 11.12 shows the IFSR register.

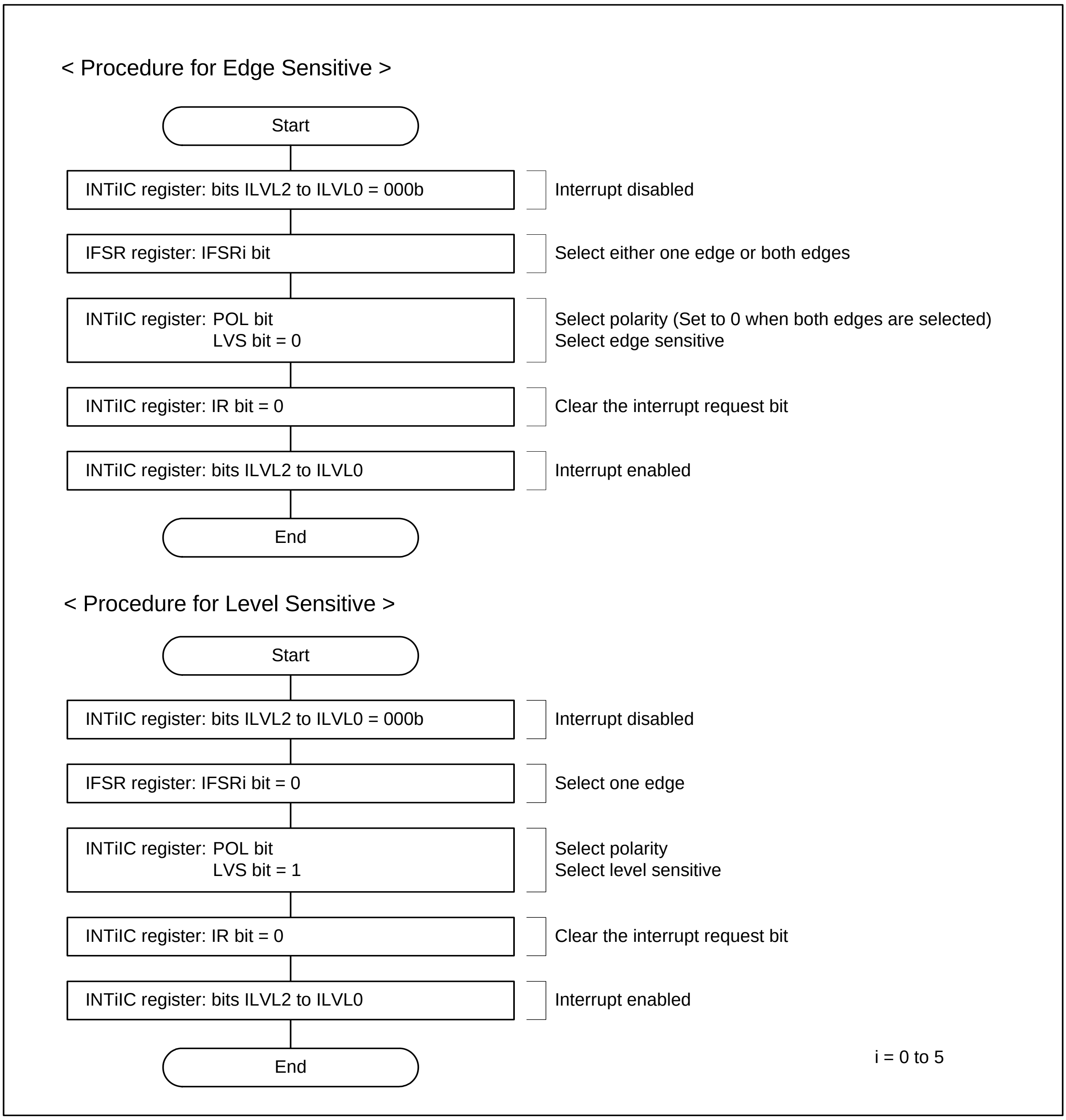
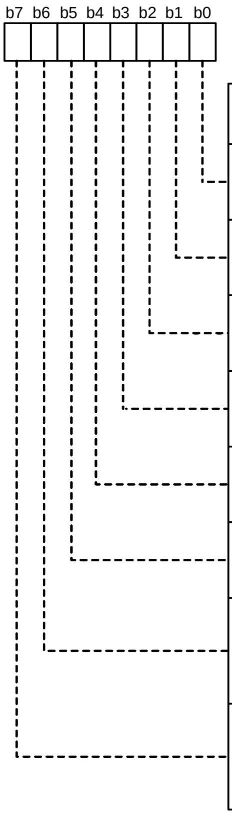

< Procedure for Edge Sensitive >
Start
INTiIC register: bits ILVL2 to ILVL0 = 000b
Interrupt disabled
IFSR register: IFSRi bit
Select either one edge or both edges
INTiIC register: POL bit
LVS bit = 0
Select polarity (Set to 0 when both edges are selected)
Select edge sensitive
INTiIC register: IR bit = 0
Clear the interrupt request bit
INTiIC register: bits ILVL2 to ILVL0
Interrupt enabled
End
< Procedure for Level Sensitive >
Start
INTiIC register: bits ILVL2 to ILVL0 = 000b
Interrupt disabled
IFSR register: IFSRi bit = 0
Select one edge
INTiIC register: POL bit
LVS bit = 1
Select polarity
Select level sensitive
INTiIC register: IR bit = 0
Clear the interrupt request bit
INTiIC register: bits ILVL2 to ILVL0
Interrupt enabled
End
i = 0 to 5

Figure 11.11 $\overline{\text{INTi}}$ Interrupt Setting Procedures (i = 0 to 5)

External Interrupt Source Select Register



Symbol	Address	After Reset
IFSR	031Fh	00h

Bit Symbol	Bit Name	Function	RW
IFSR0	$\overline{\text{INT0}}$ interrupt polarity select bit(1)	0: One edge 1: Both edges	RW
IFSR1	$\overline{\text{INT1}}$ interrupt polarity select bit(1)	0: One edge 1: Both edges	RW
IFSR2	$\overline{\text{INT2}}$ interrupt polarity select bit(1)	0: One edge 1: Both edges	RW
IFSR3	$\overline{\text{INT3}}$ interrupt polarity select bit(1)	0: One edge 1: Both edges	RW
IFSR4	$\overline{\text{INT4}}$ interrupt polarity select bit(1)	0: One edge 1: Both edges	RW
IFSR5	$\overline{\text{INT5}}$ interrupt polarity select bit(1)	0: One edge 1: Both edges	RW
IFSR6	UART0, UART3 interrupt source select bit	0: UART3 bus conflict, start condition detection, stop condition detection 1: UART0 bus conflict, start condition detection, stop condition detection	RW
IFSR7	UART1, UART4 interrupt source select bit	0: UART4 bus conflict, start condition detection, stop condition detection 1: UART1 bus conflict, start condition detection, stop condition detection	RW

NOTE:

1. Set the IFSRi bit (i = 0 to 5) to 0 to select a level-sensitive triggering. When selecting both edges, set the POL bit in the corresponding INTiIC register to 0 (falling edge).

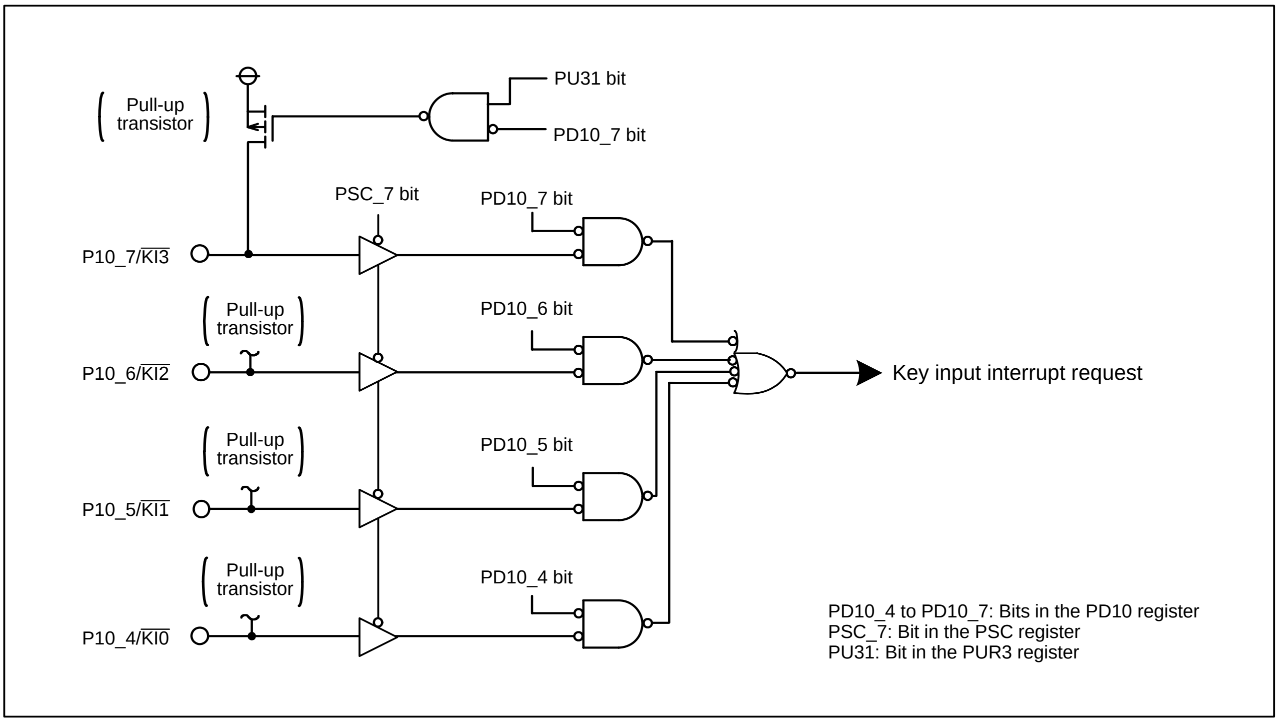
Figure 11.12 IFSR Register

11.8 NMI Interrupt

The NMI interrupt is non-maskable. The NMI interrupt occurs when a signal applied to the P8_5/NMI pin changes from “H” level to “L” level. A read from the P8_5 bit in the P8 register returns the input level of the NMI pin. When the NMI interrupt is not used, connect the NMI pin to VCC1 via a resistor (pull-up). Each “H” or “L” width of the signal applied to the NMI pin must be 2 CPU clock cycles + 300 ns or more.

11.9 Key Input Interrupt

The IR bit in the KUPIC register becomes 1 when an falling edge is detected at any of the pins P10_4 to P10_7 set to input mode. The key input interrupt can also be used as key-on wake-up function to exit wait mode or stop mode. To use the key input interrupt, do not use pins P10_4 to P10_7 as A/D input. Figure 11.13 shows a block diagram of the key input interrupt. When an “L” signal is applied to one of the pins P10_4 to P10_7 in input mode, a falling edge detected at the other pins is not recognized as an interrupt request signal.
When the PSC_7 bit in the PSC register is set to 1 (AN_4 to AN_7), the input buffer for the port and the key input interrupt is disconnected. Therefore, the pin level cannot be obtained by reading the Port P10 register in input mode. Also, the IR bit in the KUPIC register does not become 1 even if a falling edge is detected at pins KI0 to KI3.



PD10_4 to PD10_7: Bits in the PD10 register
PSC_7: Bit in the PSC register
PU31: Bit in the PUR3 register

Figure 11.13 Key Input Interrupt Block Diagram

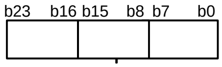
11.10 Address Match Interrupt

The address match interrupt is non-maskable. This interrupt occurs immediately before executing the instruction stored in the address specified by the RMADi register (i = 0 to 7). Eight addresses can be set for the address match interrupt. The AIERi bit in the AIER register determines whether the interrupt is enabled or disabled.

Figure 11.14 shows registers associated with the address match interrupt.

Set the starting address of the instruction in the RMADi register. The address match interrupt does not occur if a table data or any address other than the starting address of the instruction is set.

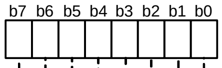
Address Match Interrupt Register i (i = 0 to 7)



Symbol	Address	After Reset
RMAD0	0012h to 0010h	000000h
RMAD1	0016h to 0014h	000000h
RMAD2	001Ah to 0018h	000000h
RMAD3	001Eh to 001Ch	000000h
RMAD4	002Ah to 0028h	000000h
RMAD5	002Eh to 002Ch	000000h
RMAD6	003Ah to 0038h	000000h
RMAD7	003Eh to 003Ch	000000h

Function	Setting Range	RW
Address register for the address match interrupt	000000h to FFFFFFh	RW

Address Match Interrupt Enable Register



Symbol	Address	After Reset
AIER	0009h	00h

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
AIER2	Address match interrupt 2 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER3	Address match interrupt 3 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER4	Address match interrupt 4 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER5	Address match interrupt 5 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER6	Address match interrupt 6 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER7	Address match interrupt 7 enable bit	0: interrupt disabled 1: interrupt enabled	RW

Figure 11.14 RMAD0 to RMAD7 Registers, AIER Register

12. Watchdog Timer

The watchdog timer is used to detect the program running improperly. The watchdog timer contains a 15-bit free-running counter. If a write to the WDTS register is not performed due to a program running out of control, the free-running counter underflows, which results in the watchdog timer interrupt generation or the MCU reset. When operating the watchdog timer, write to the WDTS register in a shorter cycle than the watchdog timer cycle in such as the main routine.

Tables 12.1 and 12.2 list specifications of the watchdog timer. Figure 12.1 shows a block diagram of the watchdog timer. Figures 12.2 and 12.3 show registers associated with the watchdog timer.

Table 12.1 Watchdog Timer Specifications (1/2)

Item	Specification
Count operation	The free-running counter decrements
Count start condition	Writing to the WDTS register: A write to the WDTS register initializes a free-running counter and the counter decrements from 7FFFh
When underflows	One of the following occurs (selectable using the CM06 bit in the CM0 register): • Watchdog timer interrupt generation(1) • MCU reset
After underflows	The counter continues decrementing (when the watchdog timer interrupt is selected)
Read from watchdog timer	A read from bit 4 to bit 0 in the WDC register returns bit 14 to bit 10 of the free-running counter

NOTE:

1. The watchdog timer shares the same vector with the oscillation stop detection interrupt and voltage monitor interrupt.

Table 12.2 Watchdog Timer Specifications (2/2)

Item	Bit Setting and Specification			
PM22 bit in PM2 register(1)	0	0	0	1
CM07 bit in CM0 register	0	0	1	0 or 1
WDC7 bit in WDC register	0	1	0 or 1	0 or 1
Clock source	CPU clock			On-chip oscillator
	Clock divided by MCD register		Sub clock	
Prescaler	Divide-by-16	Divide-by-128	Divide-by-2	not available
Count source for counter	$\frac{1}{fCPU} \times 16$	$\frac{1}{fCPU} \times 128$	$\frac{1}{fCPU} \times 2$	$\frac{1}{fROC}$
Time-out period (formula)(2)	$\frac{1}{fCPU} \times 524288$	$\frac{1}{fCPU} \times 4194304$	$\frac{1}{fCPU} \times 65536$	$\frac{1}{fROC} \times 32768$
Time-out period (reference)	Approx. 16.4 ms fCPU = 32 MHz	Approx. 131.1 ms fCPU = 32 MHz	Approx. 2 s fCPU = 32 kHz	Approx. 32.8 ms fROC = 1 MHz
Operation in wait mode, stop mode, and hold state	Stops			Operates(3)

fCPU: CPU clock frequency
fROC: On-chip oscillator clock frequency

NOTES:

1. Once the PM22 bit is set to 1, it cannot be set to 0 by a program.
2. Difference between the calculation result and actual period can be one count source cycle of the counter.
3. A write to the CM10 bit in the CM1 register is disabled. Writing a 1 has no effect and the MCU does not enter stop mode. The watchdog timer interrupt cannot be used to exit wait mode.

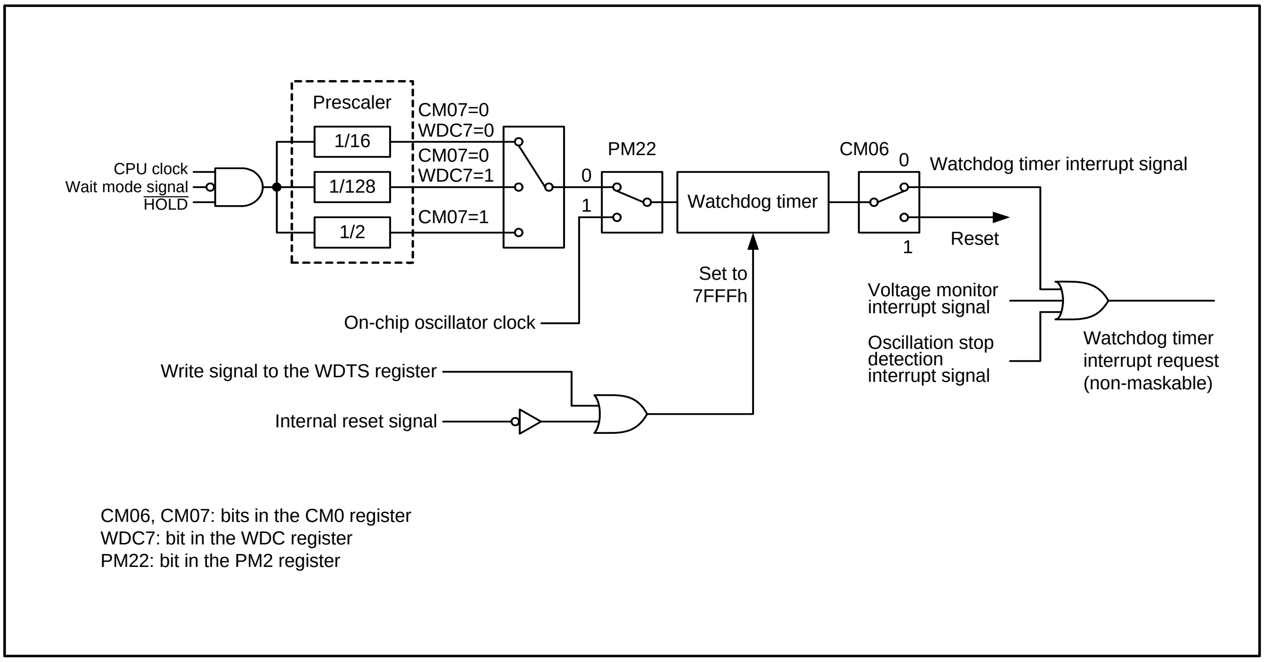
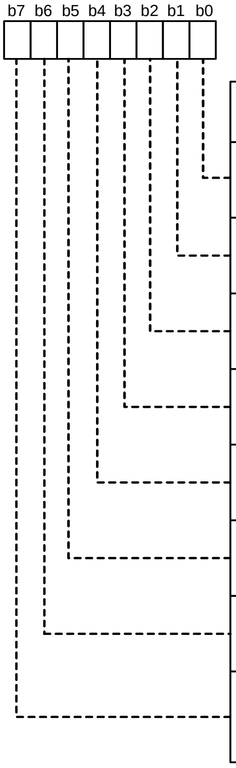


Figure 12.1 Watchdog Timer Block Diagram

System Clock Control Register 0[1]



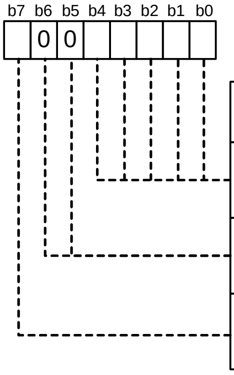
Symbol	Address	After Reset
CM0	0006h	0000 1000b

Bit Symbol	Bit Name	Function	RW
CM00	Clock output function select bits[2]	b1 b0 0 0: I/O port P5_3[2] 0 1: Outputs fC 1 0: Outputs f8 1 1: Outputs f32	RW
CM01			RW
CM02	Peripheral function clock stop in wait mode bit[9]	0: Peripheral clocks do not stop in wait mode 1: Peripheral clocks stop in wait mode[3]	RW
CM03	XCIN-XCOUT drive capability select bit[10]	0: Low 1: High	RW
CM04	Port XC switch bit	0: I/O port function 1: XCIN-XCOUT oscillation function[4]	RW
CM05	Main clock (XIN-XOUT) stop bit[5, 9]	0: Main clock oscillates 1: Main clock stops[6]	RW
CM06	Watchdog timer function select bit	0: Watchdog timer interrupt 1: Reset[7]	RW
CM07	CPU clock select bit 0[8, 9]	0: Clock selected by the CM21 bit divided by the MCD register 1: Sub clock	RW

NOTES:
1. Set the CM0 register after the PRC0 bit in the PRCR register is set to 1 (write enable).
2. The BCLK, ALE, or "L" signal is output from the P5_3 pin in memory expansion mode or microprocessor mode. The P5_3 does not function as an I/O port.
3. fC32 does not stop running.
4. To set the CM04 bit to 1, set bits PD8_7 and PD8_6 in the PD8 register to 00b (ports P8_6 and P8_7 in input mode) and the PU25 bit in the PUR2 register to 0 (not pulled up).
5. The CM05 bit stops the main clock oscillation when entering low-power consumption mode or on-chip oscillator low-power consumption mode. The CM05 bit cannot be used to determine whether the main clock stops or not. To stop the main clock oscillation, set the PLC07 bit in the PLC0 register to 0 and the CM05 bit to 1 after setting the CM07 bit to 1 or setting the CM21 bit in the CM2 register to 1 (on-chip oscillator clock).
   When the CM05 bit is set to 1, the XOUT pin outputs "H". Since an on-chip feedback resistor remains ON, the XIN pin is pulled up to the XOUT pin via the feedback resistor.
6. When the CM05 bit is set to 1, bits MCD4 to MCD0 in the MCD register become 01000b (divide-by-8 mode). In on-chip oscillator mode, bits MCD4 to MCD0 do not become 01000b even if the CM05 bit is set to 1.
7. Once the CM06 bit is set to 1, it cannot be set to 0 by a program.
8. Change the CM07 bit setting from 0 to 1, after the CM04 bit is set to 1 and the sub clock oscillation stabilizes.
   Change the CM07 bit setting from 1 to 0, after the CM05 bit is set to 0 and the main clock oscillation stabilizes.
   Do not change the CM07 bit simultaneously with the CM04 or CM05 bit.
9. If the PM21 bit in the PM2 register is set to 1 (disables a clock change), a write to bits CM02, CM05, and CM07 has no effect.
10. When stop mode is entered, the CM03 bit becomes 1.

Figure 12.2 CM0 Register

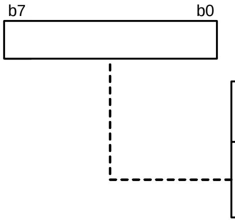
Watchdog Timer Control Register



Symbol	Address	After Reset
WDC	000Fh	000X XXXXb

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



Symbol	Address	Address
WDTS	000Eh	Undefined

Function	RW
The counter is initialized and starts decrementing by a write instruction to the WDTS register. 7FFFh is the default value after initialization no matter what value is written.	WO

Figure 12.3 WDC Register, WDTS Register

13. DMAC

DMAC allows data to be sent to and from memory without involving the CPU. The M32C/8B Group has four DMAC channels. DMAC transfers an 8- or 16-bit data from a source address to a destination address for each transfer request. DMA0 and DMA1 must be prioritized when using DMAC. DMA2 and DMA3 share the registers with the high-speed interrupts. The high-speed interrupts cannot be used when three or more DMAC channels are used.

The CPU and DMAC use the same data bus, but DMAC has a higher bus access privilege than the CPU. DMAC employing the cycle-steal method enables a high-speed operation from a transfer request to a completion of 16-bit (word) or 8-bit (byte) data transfer.

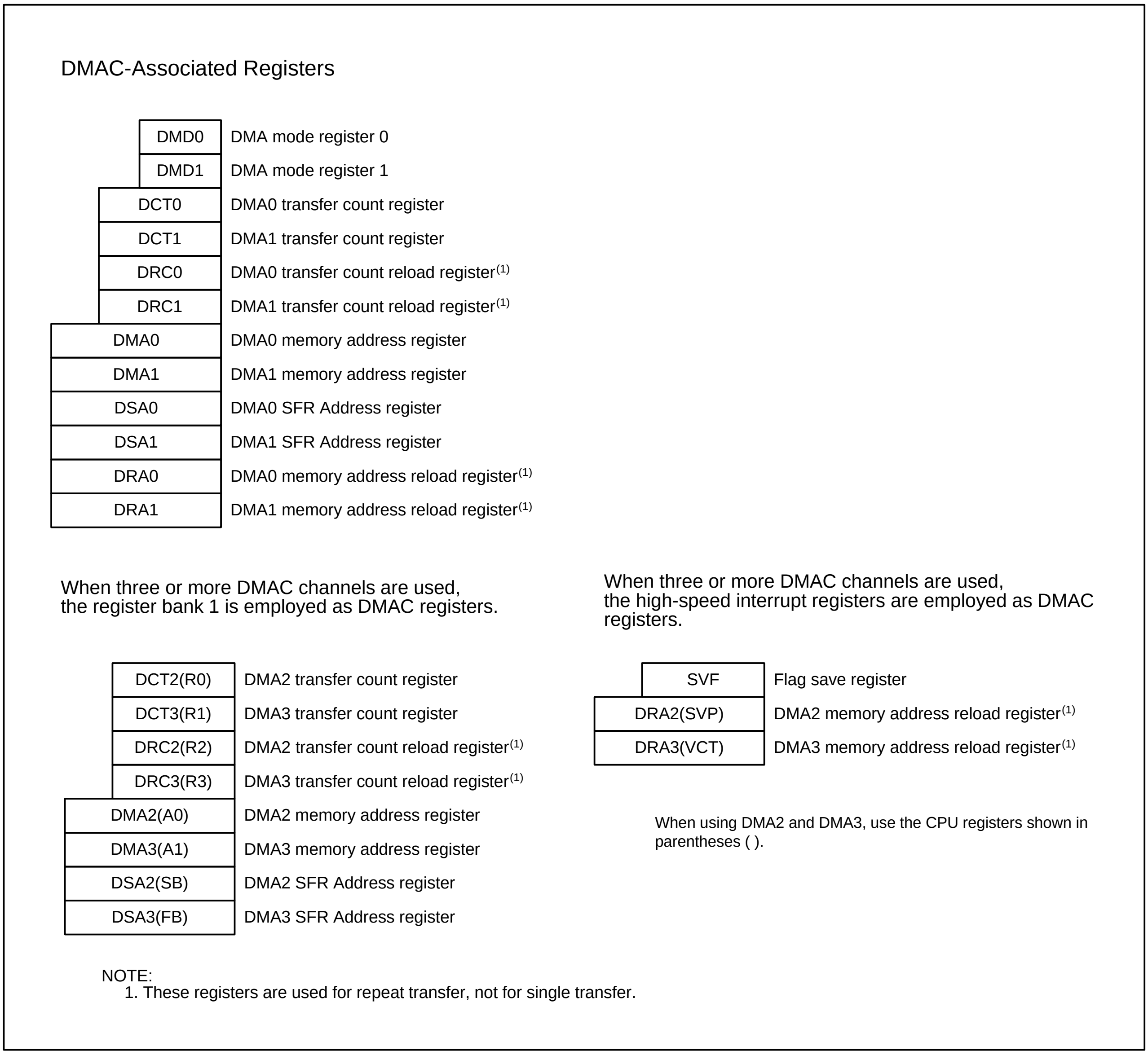
Figure 13.1 shows a mapping of DMAC-associated registers. Table 13.1 lists specifications of DMAC. Figures 13.2 to 13.6 show DMAC-associated registers. Figures 13.7 and 13.8 show register settings.

Because the registers shown in Figure 13.1 are allocated in the CPU, use the LDC instruction to set the registers.

To set registers DCT2, DCT3, DRC2, DRC3, DMA2, and DMA3, set the B flag to 1 (register bank 1) and write to registers R0 to R3, A0, and A1 with the MOV instruction.

To set registers DSA2 and DSA3, set the B flag to 1 and write to registers SB and FB with the LDC instruction.

To set registers DRA2 and DRA3, write to registers SVP and VCT with the LDC instruction.



DMAC-Associated Registers

DMD0	DMA mode register 0
DMD1	DMA mode register 1
DCT0	DMA0 transfer count register
DCT1	DMA1 transfer count register
DRC0	DMA0 transfer count reload register[1]
DRC1	DMA1 transfer count reload register[1]
DMA0	DMA0 memory address register
DMA1	DMA1 memory address register
DSA0	DMA0 SFR Address register
DSA1	DMA1 SFR Address register
DRA0	DMA0 memory address reload register[1]
DRA1	DMA1 memory address reload register[1]

When three or more DMAC channels are used, the register bank 1 is employed as DMAC registers.

DCT2(R0)	DMA2 transfer count register
DCT3(R1)	DMA3 transfer count register
DRC2(R2)	DMA2 transfer count reload register[1]
DRC3(R3)	DMA3 transfer count reload register[1]
DMA2(A0)	DMA2 memory address register
DMA3(A1)	DMA3 memory address register
DSA2(SB)	DMA2 SFR Address register
DSA3(FB)	DMA3 SFR Address register

When three or more DMAC channels are used, the high-speed interrupt registers are employed as DMAC registers.

SVF	Flag save register
DRA2(SVP)	DMA2 memory address reload register[1]
DRA3(VCT)	DMA3 memory address reload register[1]

When using DMA2 and DMA3, use the CPU registers shown in parentheses ( ).

NOTE:
1. These registers are used for repeat transfer, not for single transfer.

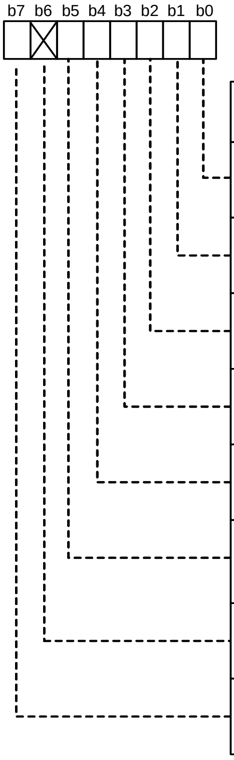
Figure 13.1 Register Mapping for DMAC

A software trigger or an interrupt request generated by individual peripheral functions can be the DMA transfer request source. Bits DSEL 4 to DSEL0 in the DMiSL register determine which source is selected. When a software trigger is selected, a DMA transfer is started by setting the DSR bit in the DMiSL register to 1. When a peripheral function interrupt request is selected, a DMA transfer is started by an interrupt request generation. The DMA transfer is performed even if interrupts are disabled by the I flag, IPL, or Interrupt Control Register, since DMAC is free from these affects. When an interrupt request (DMA request) is generated, the IR bit in the Interrupt Control Register becomes 1. The IR bit, however, does not become 0 even if the DMA transfer is performed.

Table 13.1 DMAC Specifications

Item		Specification
Number of Channels		4 channels (cycle-steal method)
Transfer memory space		• From a given address in a 16-Mbyte space to a fixed address in a 16-Mbyte space • From a fixed address in a 16-Mbyte space to a given address in a 16-Mbyte space
Maximum bytes transferred		128 Kbytes (when a 16-bit data is transferred) 64 Kbytes (when an 8-bit data is transferred)
DMA request source		• Falling edge or both edges of signals applied to pins $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ • Timer A0 to A4 interrupt requests • Timer B0 to B5 interrupt requests • UART0 to UART4 transmit/receive interrupt requests • A/D0 interrupt request • Software trigger
Channel priority		DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority)
Transfer unit		8 bits, 16 bits
Transfer address		Fixed address: one specified address Incremented address: address which is incremented by a transfer unit on each successive access. (Source address and destination address cannot be both fixed nor both incremented.)
Transfer mode	Single transfer	Transfer is completed when the DCTi register (i = 0 to 3) becomes 0000h.
	Repeat transfer	When the DCTi register becomes 0000h, values of the DRCi register are reloaded into the DCTi register and the DMA transfer continues.
DMA interrupt request generation timing		When the DCTi register becomes from 0001h to 0000h, a DMA interrupt request is generated.
DMA start	Single transfer	DMAC starts a data transfer when a DMA request is generated after bits MDi1 and MDi0 in the DMDj register (j = 0 to 1) are set to 01b (single transfer), while the DCTi register is set to 0001h or higher value.
	Repeat transfer	DMAC starts a data transfer when a DMA request is generated after bits MDi1 and MDi0 are set to 11b (repeat transfer), while the DCTi register is set to 0001h or higher value.
DMA stop	Single transfer	• When bits MDi1 and MDi0 are set to 00b (DMA disabled). • When the DCTi register becomes 0000h (no DMA transfer) at completion of DMA transfer, or is set to 0000h by a program.
	Repeat transfer	• When bits MDi1 and MDi0 are set to 00b (DMA disabled). • When the DCTi register becomes 0000h (no DMA transfer) at completion of DMA transfer, or is set to 0000h by a program while the DRCi register is 0000h.
Reload timing to registers DCTi and DMAi		Values are reloaded when the DCTi register becomes from 0001h to 0000h in repeat transfer mode.
DMA transfer time		Between SFR area and internal RAM transfer: minimum 3 bus clock cycles

DMAi Request Source Select Register (i=0 to 3)



Symbol	Address	After Reset
DM0SL to DM3SL	0378h, 0379h, 037Ah, 037Bh	0X00 0000b

Bit Symbol	Bit Name	Function	RW
DSEL0	DMA request source select bits[1]	See **Table "DMiSL register function (i = 0 to 3)"** Do not set to values other than specified in the Table.	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
DSEL4			RW
DSR	Software DMA request bit[2]	When a software trigger is selected, a DMA request is generated by setting this bit to 1 (Read as 0)	RW
– (b6)	Reserved bit	Read as undefined value	–
DRQ	DMA request bit[2, 3]	0: Not requested 1: Requested	RW

NOTES:

1. Change settings of bits DSEL4 to DSEL0 while bits MDi1 and MDi0 in the DMD0 or DMD1 register are set to 00b (DMA disabled). Also, when bits DSEL4 to DSEL0 are changed, set the DRQ bit to 1 at the same time.
   e.g., MOV.B #083h, DMiSL ; Select timer A0
2. When the DSR bit is set to 1, set the DRQ bit to 1 at the same time.
   e.g., OR.B #0A0h, DMiSL
3. Do not write a 0 to the DRQ bit.

Figure 13.2 DM0SL to DM3SL Registers

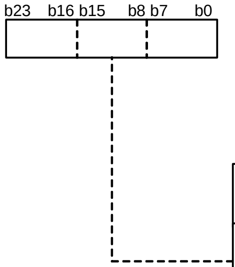
Table 13.2 DMiSL Register (i = 0 to 3) Function

Setting Value					DMA Request Source				
b4	b3	b2	b1	b0	DMA0	DMA1	DMA2	DMA3	
0	0	0	0	0	Software trigger				
0	0	0	0	1	Falling edge of $\overline{\mathrm{INT0}}$	Falling edge of $\overline{\mathrm{INT1}}$	Falling edge of $\overline{\mathrm{INT2}}$	Falling edge of $\overline{\mathrm{INT3}}$[1]	(Note 2)
0	0	0	1	0	Both edges of $\overline{\mathrm{INT0}}$	Both edges of $\overline{\mathrm{INT1}}$	Both edges of $\overline{\mathrm{INT2}}$	Both edges of $\overline{\mathrm{INT3}}$[1]	(Note 2)
0	0	0	1	1	Timer A0 interrupt request				
0	0	1	0	0	Timer A1 interrupt request				
0	0	1	0	1	Timer A2 interrupt request				
0	0	1	1	0	Timer A3 interrupt request				
0	0	1	1	1	Timer A4 interrupt request				
0	1	0	0	0	Timer B0 interrupt request				
0	1	0	0	1	Timer B1 interrupt request				
0	1	0	1	0	Timer B2 interrupt request				
0	1	0	1	1	Timer B3 interrupt request				
0	1	1	0	0	Timer B4 interrupt request				
0	1	1	0	1	Timer B5 interrupt request				
0	1	1	1	0	UART0 transmit interrupt request				
0	1	1	1	1	UART0 receive interrupt or ACK interrupt request[3]				
1	0	0	0	0	UART1 transmit interrupt request				
1	0	0	0	1	UART1 receive interrupt or ACK interrupt request[3]				
1	0	0	1	0	UART2 transmit interrupt request				
1	0	0	1	1	UART2 receive interrupt or ACK interrupt request[3]				
1	0	1	0	0	UART3 transmit interrupt request				
1	0	1	0	1	UART3 receive interrupt or ACK interrupt request[3]				
1	0	1	1	0	UART4 transmit interrupt request				
1	0	1	1	1	UART4 receive interrupt or ACK interrupt request[3]				
1	1	0	0	0	A/D0 interrupt request				

NOTES:

1. When the $\overline{\mathrm{INT3}}$ pin is used for data bus in memory expansion mode or microprocessor mode, a DMA3 interrupt request cannot be generated by an input signal to the $\overline{\mathrm{INT3}}$ pin.
2. The falling edge or both edges of input signal to the $\overline{\mathrm{INTi}}$ pin can be a DMA request source. It is not affected by the $\overline{\mathrm{INT}}$ interrupts (bits POL and LVS in the INTiIC register, the IFSR register) and vice versa.
3. To switch between the UARTj receive interrupt and ACK interrupt (j = 0 to 4), use the IICM bit in the UiSMR register and IICM2 bit on the UiSMR2 register. To use the ACK interrupt, set the IICM bit to 1 (I^{2}C mode) and the IICM2 bit to 0 (NACK/ACK interrupt).

DMAi Memory Address Register (i = 0 to 3)

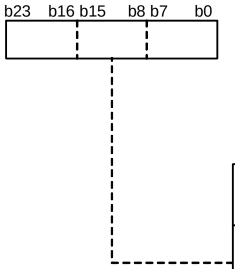


Symbol	Address	After Reset
DMA0[2]	(CPU internal register)	XXXXXXh
DMA1[2]	(CPU internal register)	XXXXXXh
DMA2 (bank1:A0)[3]	(CPU internal register)	000000h
DMA3 (bank1:A1)[4]	(CPU internal register)	000000h

Function	Setting Range	RW
Set an incremented source address or incremented destination address[1]	000000h to FFFFFFh (16 Mbytes)	RW

NOTES:
1. When the RWk bit (k = 0 to 3) in the DMDj register (j = 0, 1) is set to 0 (fixed address to incremented address), a destination address is selected. When the RWk bit is set to 1 (incremented address to fixed address), a source address is selected.
2. Use the LDC instruction to set registers DMA0 and DMA1.
3. To set the DMA2 register, set the B flag in the FLG register to 1 (register bank 1) and write to the A0 register.
4. To set the DMA3 register, set the B flag to 1 and write to the A1 register.

DMAi SFR Address Register (i = 0 to 3)



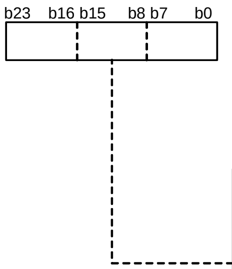
Symbol	Address	After Reset
DSA0[2]	(CPU internal register)	XXXXXXh
DSA1[2]	(CPU internal register)	XXXXXXh
DSA2 (bank1:SB)[3]	(CPU internal register)	000000h
DSA3 (bank1:FB)[4]	(CPU internal register)	000000h

Function	Setting Range	RW
Set a fixed source address or fixed destination address[1]	000000h to FFFFFFh (16 Mbytes)	RW

NOTES:
1. When the RWk bit (k = 0 to 3) in the DMDj register (j = 0, 1) is set to 0 (fixed address to incremented address), a source address is selected. When the RWk bit is set to 1 (incremented address to fixed address), a destination address is selected.
2. Use the LDC instruction to set registers DSA0 and DSA1.
3. To set the DSA2 register, set the B flag in the FLG register to 1 (register bank 1) and write to the SB register using the LDC instruction.
4. To set the DSA3 register, set the B flag to 1 and write to the FB register using the LDC instruction.

Figure 13.3 DMA0 to DMA3 Registers, DSA0 to DSA3 Registers

DMAi Memory Address Reload Register[1] (i = 0 to 3)

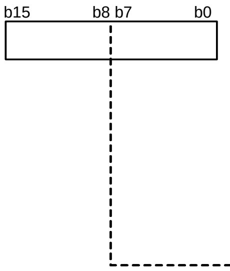


Symbol	Address	After Reset
DRA0	(CPU internal register)	XXXXXXh
DRA1	(CPU internal register)	XXXXXXh
DRA2 (SVP)[2]	(CPU internal register)	XXXXXXh
DRA3 (VCT)[3]	(CPU internal register)	XXXXXXh

Function	Setting Range	RW
Set an incremented source address or incremented destination address	000000h to FFFFFFh (16 Mbytes)	RW

NOTES:
1. Use the LDC instruction to set registers DRA0 to DRA3.
2. To set the DRA2 register, write to the SVP register.
3. To set the DRA3 register, write to the VCT register.

DMAi Transfer Count Register (i = 0 to 3)

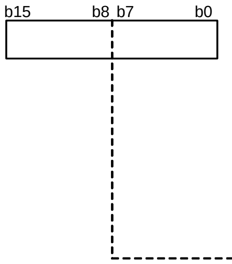


Symbol	Address	After Reset
DCT0[2]	(CPU internal register)	XXXXh
DCT1[2]	(CPU internal register)	XXXXh
DCT2 (bank1:R0)[3]	(CPU internal register)	0000h
DCT3 (bank1:R1)[4]	(CPU internal register)	0000h

Function	Setting Range	RW
Set the number of transfers	0000h to FFFFh[1]	RW

NOTES:
1. When the DCTi register is set to 0000h, no data transfer occurs regardless of a DMA request generation.
2. Use the LDC instruction to set registers DCT0 and DCT1.
3. To set the DCT2 register, set the B flag in the FLG register to 1 (register bank 1) and write to the R0 register.
4. To set the DCT3 register, set the B flag to 1 and write to the R1 register.

DMAi Transfer Count Reload Register (i = 0 to 3)



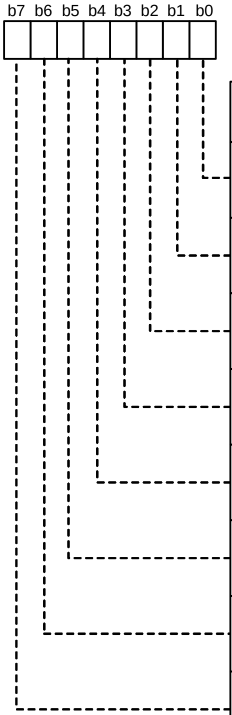
Symbol	Address	After Reset
DRC0[1]	(CPU internal register)	XXXXh
DRC1[1]	(CPU internal register)	XXXXh
DRC2 (bank1:R2)[2]	(CPU internal register)	0000h
DRC3 (bank1:R3)[3]	(CPU internal register)	0000h

Function	Setting Range	RW
Set the number of transfers	0000h to FFFFh	RW

NOTES:
1. Use the LDC instruction to set registers DRC0 and DRC1.
2. To set the DRC2 register, set the B flag in the FLG register to 1 (register bank 1) and write to the R2 register.
3. To set the DRC3 register, set the B flag to 1 and write to the R3 register.

Figure 13.4 DRA0 to DRA3 Registers, DCT0 to DCT3 Registers, DRC0 to DRC3 Registers

DMA Mode Register 0[1]



Symbol	Address	After Reset
DMD0	(CPU internal register)	00h

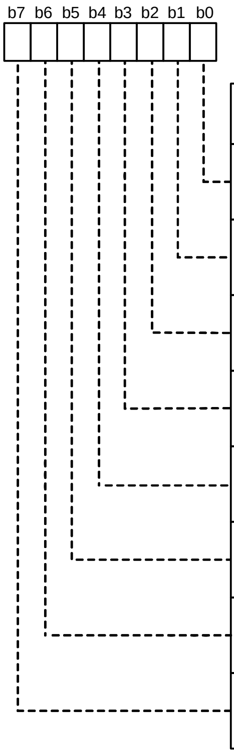
Bit Symbol	Bit Name	Function	RW
MD00	Channel 0 transfer mode select bits	b1 b0 0 0: DMA disabled 0 1: Single transfer 1 0: Do not set to this value 1 1: Repeat transfer	RW
MD01			RW
BW0	Channel 0 transfer unit select bit	0: 8 bits 1: 16 bits	RW
RW0	Channel 0 transfer direction select bit	0: Fixed address to incremented address 1: Incremented address to fixed address	RW
MD10	Channel 1 transfer mode select bits	b5 b4 0 0: DMA disabled 0 1: Single transfer 1 0: Do not set to this value 1 1: Repeat transfer	RW
MD11			RW
BW1	Channel 1 transfer unit select bit	0: 8 bits 1: 16 bits	RW
RW1	Channel 1 transfer direction select bit	0: Fixed address to incremented address 1: Incremented address to fixed address	RW

NOTE:

1. Use the LDC instruction to set the DMD0 register.

Figure 13.5 DMD0 Register

DMA Mode Register 1[(1)]



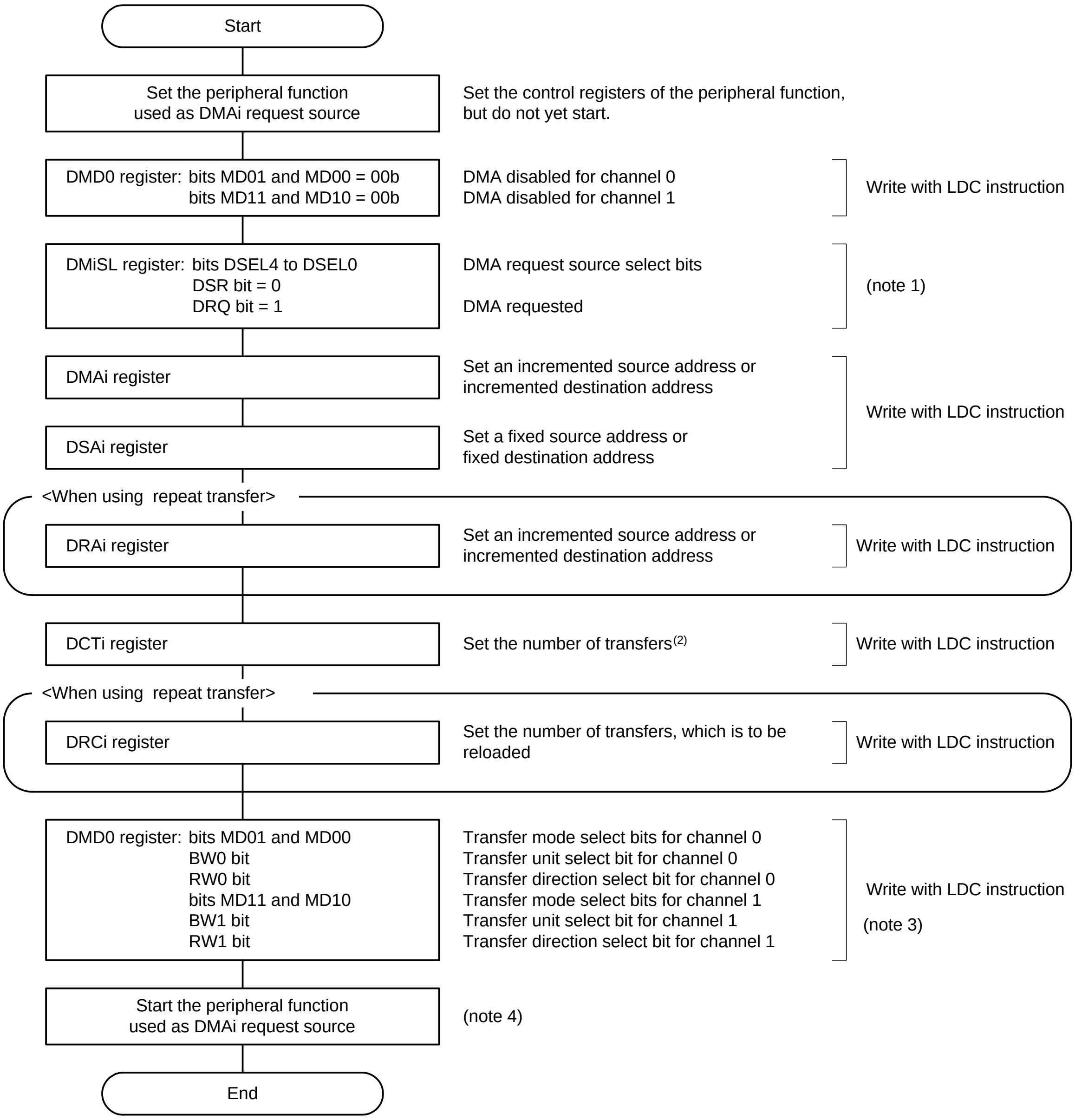
Symbol	Address	After Reset
DMD1	(CPU internal register)	00h

Bit Symbol	Bit Name	Function	RW
MD20	Channel 2 transfer mode select bits	b1 b0 0 0: DMA disabled 0 1: Single transfer 1 0: Do not set to this value 1 1: Repeat transfer	RW
MD21			RW
BW2	Channel 2 transfer unit select bit	0: 8 bits 1: 16 bits	RW
RW2	Channel 2 transfer direction select bit	0: Fixed address to incremented address 1: Incremented address to fixed address	RW
MD30	Channel 3 transfer mode select bits	b5 b4 0 0: DMA disabled 0 1: Single transfer 1 0: Do not set to this value 1 1: Repeat transfer	RW
MD31			RW
BW3	Channel 3 transfer unit select bit	0: 8 bits 1: 16 bits	RW
RW3	Channel 3 transfer direction select bit	0: Fixed address to incremented address 1: Incremented address to fixed address	RW

NOTE:

1. Use the LDC instruction to set the DMD1 register.

Figure 13.6 DMD1 Register

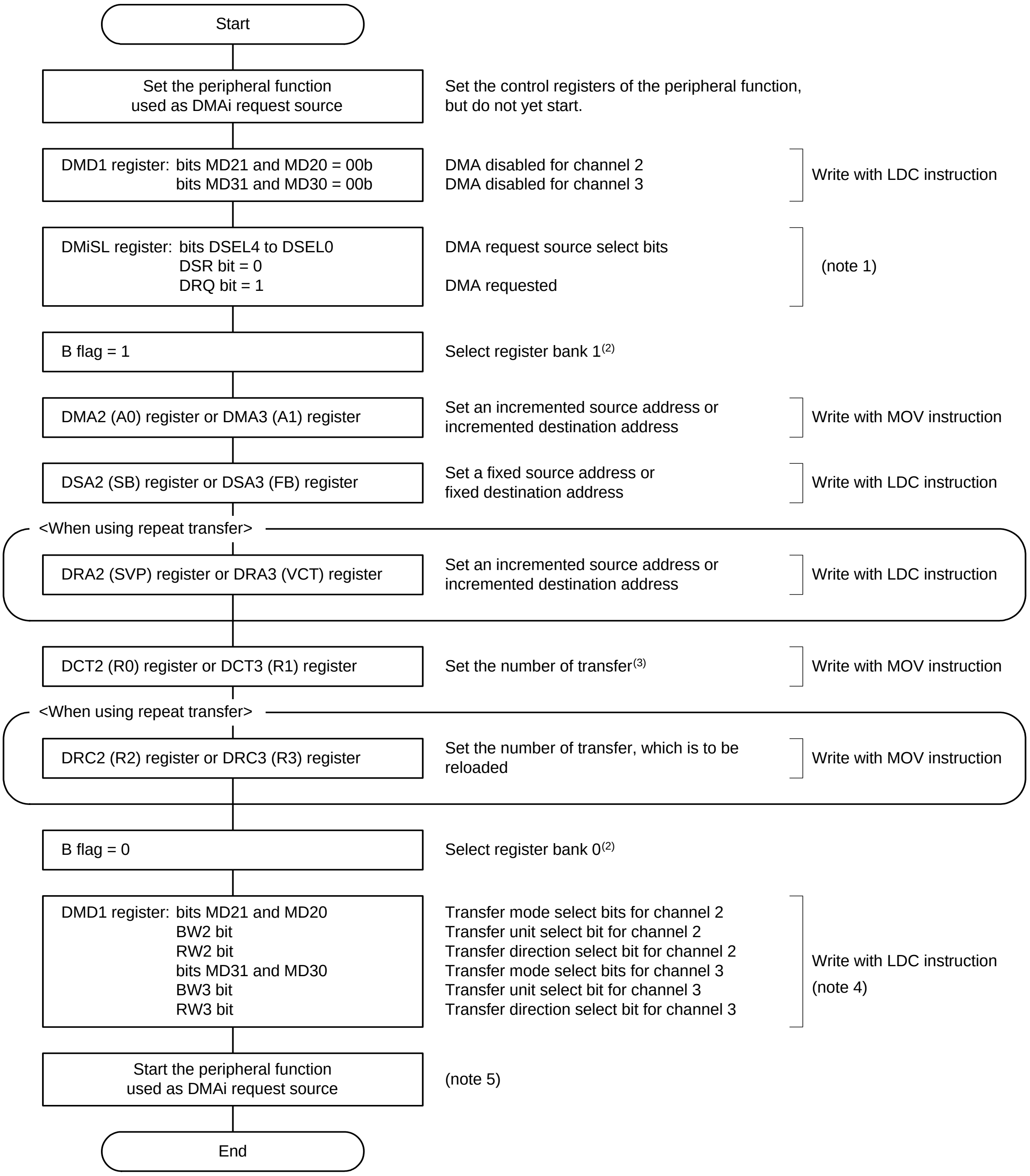

Start
Set the peripheral function used as DMAi request source
Set the control registers of the peripheral function, but do not yet start.
DMD0 register: bits MD01 and MD00 = 00b
bits MD11 and MD10 = 00b
DMA disabled for channel 0
DMA disabled for channel 1
Write with LDC instruction
DMiSL register: bits DSEL4 to DSEL0
DSR bit = 0
DRQ bit = 1
DMA request source select bits
DMA requested
(note 1)
DMAi register
Set an incremented source address or incremented destination address
DSAi register
Set a fixed source address or fixed destination address
Write with LDC instruction
<When using repeat transfer>
DRAi register
Set an incremented source address or incremented destination address
Write with LDC instruction
DCTi register
Set the number of transfers(2)
Write with LDC instruction
<When using repeat transfer>
DRCi register
Set the number of transfers, which is to be reloaded
Write with LDC instruction
DMD0 register: bits MD01 and MD00
BW0 bit
RW0 bit
bits MD11 and MD10
BW1 bit
RW1 bit
Transfer mode select bits for channel 0
Transfer unit select bit for channel 0
Transfer direction select bit for channel 0
Transfer mode select bits for channel 1
Transfer unit select bit for channel 1
Transfer direction select bit for channel 1
Write with LDC instruction
(note 3)
Start the peripheral function used as DMAi request source
(note 4)
End

i = 0, 1

NOTES:

1. When setting the DMiSL register, write a 1 to the DRQ bit.
2. When the $\overline{\text{INT}}$ interrupts are selected as a DMA request source, do not write a 1 to the DCTi register. If the DCTi register is 1, do not generate a DMA request when writing 01b or 11b to bits MDi1 and MDi0.
3. Wait six CPU clock cycles or more by a program to set bits MDi1 and MDi0 to 01b or 11b after setting the DMiSL register.
4. When a DMA transfer is started by the software trigger, set both the DSR and DRQ bit in the DMiSL register to 1 at the same time.

Figure 13.7 Register Settings When Using DMA0 or DMA1


Start
Set the peripheral function used as DMAi request source
Set the control registers of the peripheral function, but do not yet start.
DMD1 register: bits MD21 and MD20 = 00b
bits MD31 and MD30 = 00b
DMA disabled for channel 2
DMA disabled for channel 3
Write with LDC instruction
DMiSL register: bits DSEL4 to DSEL0
DSR bit = 0
DRQ bit = 1
DMA request source select bits
DMA requested
(note 1)
B flag = 1
Select register bank 1(2)
DMA2 (A0) register or DMA3 (A1) register
Set an incremented source address or incremented destination address
Write with MOV instruction
DSA2 (SB) register or DSA3 (FB) register
Set a fixed source address or fixed destination address
Write with LDC instruction
<When using repeat transfer>
DRA2 (SVP) register or DRA3 (VCT) register
Set an incremented source address or incremented destination address
Write with LDC instruction
DCT2 (R0) register or DCT3 (R1) register
Set the number of transfer(3)
Write with MOV instruction
<When using repeat transfer>
DRC2 (R2) register or DRC3 (R3) register
Set the number of transfer, which is to be reloaded
Write with MOV instruction
B flag = 0
Select register bank 0(2)
DMD1 register: bits MD21 and MD20
BW2 bit
RW2 bit
bits MD31 and MD30
BW3 bit
RW3 bit
Transfer mode select bits for channel 2
Transfer unit select bit for channel 2
Transfer direction select bit for channel 2
Transfer mode select bits for channel 3
Transfer unit select bit for channel 3
Transfer direction select bit for channel 3
Write with LDC instruction
(note 4)
Start the peripheral function used as DMAi request source
(note 5)
End

i = 2, 3

NOTES:

1. When setting the DMiSL register, write a 1 to the DRQ bit.
2. The register bank 1 and high-speed interrupt cannot be used when using DMA2 and DMA3.
3. When the $\overline{\text{INT}}$ interrupts are selected as a DMA request source, do not write a 1 to the DCTi register. If the DCTi register is 1, do not generate a DMA request when writing 01b or 11b to bits MDi1 and MDi0.
4. Wait six CPU clock cycles or more by a program to set bits MDi1 and MDi0 to 01b or 11b after setting the DMiSL register.
5. When a DMA transfer is started by the software trigger, set both the DSR and DRQ bit in the DMiSL register to 1 at the same time.

Figure 13.8 Register Settings When Using DMA2 or DMA3

13.1 Transfer Cycles

The transfer cycle is composed of bus cycles to read data from source address (source read) and bus cycles to write data to destination address (destination write). The number of read and write bus cycles depends on the locations of source and destination addresses. In memory expansion mode and microprocessor mode, the number of read and write bus cycles also depends on DS register setting. Software wait state insertion and the $\overline{\mathrm{RDY}}$ signal can extend the number of the bus cycles.

13.1.1 Effect of Source and Destination Addresses

When a 16-bit data is transferred with a 16-bit data bus and a source address starts with an odd address, the source-read cycle is added by one bus cycle, compared to a source address starting with an even address.
When a 16-bit data is transferred with a 16-bit data bus and a destination address starts with an odd address, the destination-write cycle is added by one bus cycle, compared to a destination address starting with an even address.

13.1.2 Effect of the DS Register

In an external space in memory expansion mode and microprocessor mode, the transfer cycle varies depending on the data bus width of the source and destination addresses. See **Figure 8.1** for details about the DS register.

- When a 16-bit data is transferred accessing both source address and destination address with an 8-bit data bus (the DSi bit in the DS register is set to 0 (i = 0 to 3)), an 8-bit data will be transferred twice. Therefore, two bus cycles are required for reading and another two bus cycles for writing.
- When a 16-bit data is transferred accessing a source address with an 8-bit data bus (the DSi bit is set to 0) and a destination address with a 16-bit data bus, an 8-bit data will be read twice but be written once as 16-bit data. Therefore, two bus cycles are required for reading and one bus cycle for writing.
- When a 16-bit data is transferred accessing a source address with a 16-bit data bus (the DSi bit is set to 1) and a destination address with an 8-bit data bus, a 16-bit data will be read once and an 8-bit data will be written twice. Therefore, one bus cycle is required for reading and two bus cycles for writing.

13.1.3 Effect of Software Wait State

When accessing the SFR area or memory space that requires wait states, the number of bus clocks (BCLK) is increased by software wait states.

13.1.4 Effect of the $\overline{\mathrm{RDY}}$ Signal

In memory expansion mode and microprocessor mode, the $\overline{\mathrm{RDY}}$ signal affects the number of the bus cycles if a source address or destination address is in an external space. Refer to **8.2.6 $\overline{\mathrm{RDY}}$ Signal** for details.

13.2 DMA Transfer Time

The DMA transfer time can be calculated as follows. (in terms of bus clock)
Table 13.3 lists the number of the source read cycle and destination write cycle. Table 13.4 lists coefficient j, k (the number of bus clock).

Transfer time = source read bus cycle × j + destination write bus cycle × k

Table 13.3 Source Read Cycle and Destination Write Cycle

Transfer Unit	Bus Width	Access Address	Accessing Internal Space		Accessing External Space	
			Read Cycle	Write Cycle	Read Cycle	Write Cycle
8-bit transfer (BWi bit in the DMDp register = 0)	16 bits	Even	1	1	1	1
		Odd	1	1	1	1
	8 bits	Even	–	–	1	1
		Odd	–	–	1	1
16-bit transfer (BWi bit = 1)	16 bits	Even	1	1	1	1
		Odd	2	2	2	2
	8 bits	Even	–	–	2	2
		Odd	–	–	2	2

i=0 to 3, p=0 and 1

Table 13.4 Coefficient j, k

Internal Space			External Space
Internal ROM or internal RAM	Internal ROM or internal RAM	SFR area	j and k BCLK cycles shown in **Table 8.6** (j, k = 2 to 9). Add one cycle to j or k cycles when inserting a recovery cycle

with no wait state
j=1
k=1
	with wait state
	j=2
	k=2
		j=2
		k=2	

13.3 Channel Priority and DMA Transfer Timing

When multiple DMA requests are generated in the same sampling period (between a falling edge of the BCLK and the next falling edge), the corresponding DRQ bits in the DMiSL register (i = 0 to 3) are set to 1 (requested) simultaneously. Channel priority in this case is: DMA0 > DMA1 > DMA2 > DMA3. Leave the following period between each DMA transfer request generation on the same channel.

DMA request interval ≥ (number of channels set for DMA transfer - 1) × 5 BCLK cycles

Described in the following is the operation when DMA0 and DMA1 requests are generated in the same sampling period. Figure 13.9 shows an example of DMA transfers triggered by the $\overline{\text{INT}}$ interrupts.
In Figure 13.9, DMA0 and DMA1 requests are generated simultaneously. A DMA0 request having higher priority is acknowledged first to start a transfer. After one DMA0 transfer is completed, the DMAC returns ownership of the bus to the CPU. When the CPU has completed one bus access, a DMA1 transfer starts. After one DMA1 transfer is completed, bus ownership is again returned to the CPU.
DMA requests cannot be counted up since each channel has one DRQ bit. Even if multiple DMA1 requests are generated before receiving bus ownership as shown in Figure 13.9, the DRQ bit is set to 0 as soon as bus ownership is acquired. Bus ownership is returned to the CPU after one transfer is completed.

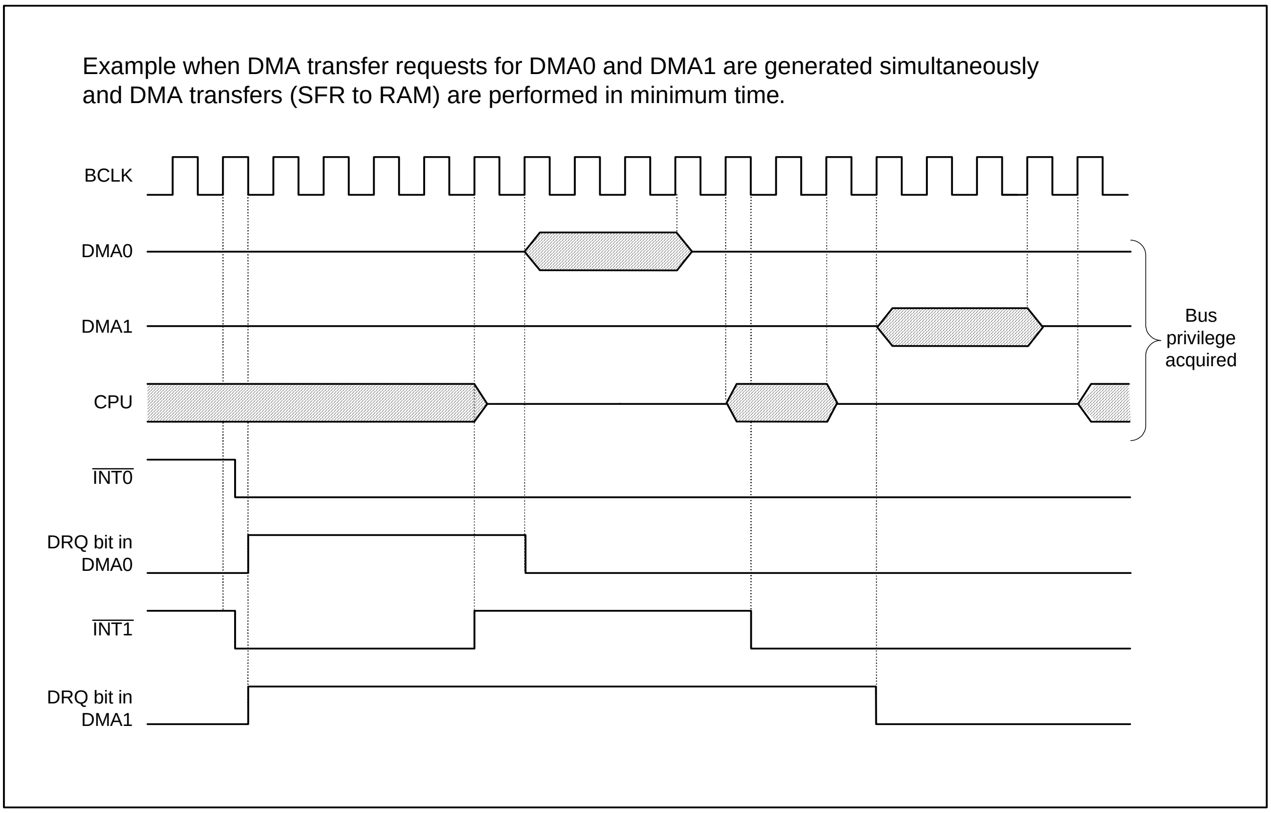

Example when DMA transfer requests for DMA0 and DMA1 are generated simultaneously
and DMA transfers (SFR to RAM) are performed in minimum time.
BCLK
DMA0
DMA1
CPU
$\overline{\text{INT0}}$
DRQ bit in DMA0
$\overline{\text{INT1}}$
DRQ bit in DMA1
Bus privilege acquired

Figure 13.9 DMA Transfers Triggered by $\overline{\text{INT}}$ Interrupt Requests

14. DMACII

DMACII performs memory-to-memory transfer, immediate data transfer, and calculation transfer which transfers a result of the addition of two data. DMACII transfer occurs in response to interrupt requests from the peripheral functions.
Table 14.1 lists specifications of DMACII.

Table 14.1 DMACII Specifications

Item	Specification
DMACII request source	Interrupt requests generated by any peripheral functions with bits ILVL2 to ILVL0 in the Interrupt Control Register set to 111b (level 7)
Transfer data	- Data in a memory location is transferred to another memory location (memory-to-memory transfer) - Immediate data is transferred to a memory location (immediate data transfer) - Data in a memory location (or immediate data) + data in another memory location is transferred to the other memory location (calculation transfer)
Transfer unit	8 bits or 16 bits
Transfer space	64-Kbyte space in addresses 00000h to 0FFFFh[1][2]
Transfer address	Fixed address: one specified address Incremented address: address which is incremented by the transfer unit on each successive access. (Selectable for source address and destination address individually)
Transfer mode	Single transfer, burst transfer, multiple transfer
Chain transfer function	Address indicated by an interrupt vector for DMACII index is replaced when a transfer counter reaches zero
End-of-transfer interrupt	Interrupt occurs when a transfer counter reaches zero

NOTES:

1. When a destination address is 0FFFFh and a 16-bit data is transferred, it is transferred to addresses 0FFFFh and 10000h. Likewise, when a source address is 0FFFFh, a 16-bit data in addresses 0FFFFh and 10000h is transferred to a given destination address.
2. The actual transferable space varies depending on internal RAM capacity.

14.1 DMACII Settings

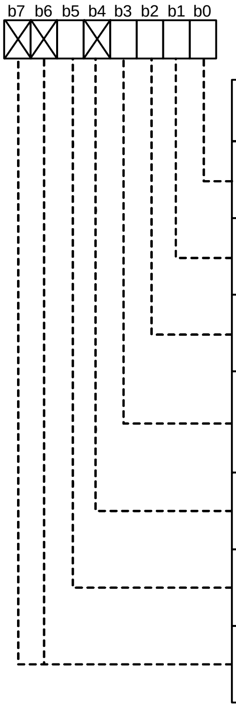
Set up the following registers and tables to activate DMACII.

- RLVL register
- DMACII Index
- Interrupt Control Register of the peripheral functions triggering DMACII requests
- The relocatable vector table of the peripheral functions triggering DMACII requests

14.1.1 RLVL Register

When the DMAII bit is set to 1 (interrupt priority level 7 is used for DMACII transfer) and the FSIT bit to 0 (interrupt priority level 7 is used for normal interrupt), DMACII is activated by an interrupt request from any peripheral functions with bits ILVL2 to ILVL0 in the Interrupt Control Register set to 111b (level 7).
Figure 14.1 shows the RLVL register.

Exit Priority Register



Symbol	Address	After Reset
RLVL	009Fh	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
RLVL0	Exit wait mode/stop mode interrupt priority level control bits[1]	b2 b1 b0 0 0 0: Level 0 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
RLVL1			RW
RLVL2			RW
FSIT	High-speed interrupt select bit	0: Interrupt priority level 7 is used for normal interrupt 1: Interrupt priority level 7 is used for high-speed interrupt[2][3]	RW
– (b4)	Unimplemented. Write 0. Read as undefined value.		–
DMAII	DMACII select bit[4]	0: Interrupt priority level 7 is used for interrupt 1: Interrupt priority level 7 is used for DMACII transfer[2]	RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:

1. The MCU exits stop or wait mode when an interrupt priority level of a requested interrupt is higher than a level set using bits RLVL2 to RLVL0. Set bits RLVL2 to RLVL0 to the same value as IPL in the FLG register.
2. Do not set both the FSIT and DMAII bits to 1.
   Set either the FSIT bit or the DMAII bit to 1 before setting bits ILVL2 to ILVL0 in the Interrupt Control Register to 111b.
3. Only one interrupt can have the interrupt priority level 7 when selecting the high-speed interrupt.
4. The DMAII bit is undefined after reset. To use interrupt priority level 7 for an interrupt, set it to 0 before setting the Interrupt Control Register.

Figure 14.1 RLVL Register

14.1.2 DMACII Index

The DMACII index is an 8- to 32-byte data table, which stores parameters for transfer mode, transfer counter, source address (or immediate data), operation address as an address to be calculated, destination address, chain transfer address, and end-of-transfer interrupt address.

The DMACII index must be located on the RAM area.

Figure 14.2 shows a configuration of the DMACII index. Table 14.2 lists an example configuration of the DMACII index.

Memory-to-Memory Transfer, Immediate Transfer, Calculation Transfer

DMACII Index	16 bits
Starting Address (BASE)	Transfer mode (MOD)
BASE+2	Transfer counter (COUNT)
BASE+4	Transfer source address (or immediate data) (SADR)
BASE+6	Operation address[1] (OADR)
BASE+8	Transfer destination address (DADR)
BASE+10	Chain Transfer Address (lower byte)[2] (CADR0)
BASE+12	Chain Transfer Address (higher byte)[2] (CADR1)
BASE+14	End-of-Transfer Interrupt Address (lower byte)[3] (IADR0)
BASE+16	End-of-Transfer Interrupt Address (higher byte)[3] (IADR1)

Multiple Transfer

	16 bits
BASE	Transfer mode (MOD)
BASE+2	Transfer counter (COUNT)
BASE+4	Transfer source address (SADR1)
BASE+6	Transfer destination address (DADR1)
BASE+8	Transfer source address (SADR2)
BASE+10	Transfer destination address (DADR2)
	to
BASE+28	Transfer source address (SADR7)
BASE+30	Transfer destination address (DADR7)

NOTES:

1. This data is not needed unless using the calculation transfer function.
2. This data is not needed unless using the chain transfer function.
3. This data is not needed unless using the end-of-transfer interrupt.

Place the DMACII index in the RAM. Necessary data must be set top-aligned without any space. For example, if not using the calculation transfer function, assign a transfer destination address to BASE+6.
The starting address of the DMACII index must be assigned to the interrupt vector of the peripheral function interrupt triggering a DMACII request.

Figure 14.2 DMACII Index

Details of the DMACII index are described below. Set these parameters in the specified order listed in Table 14.2, depending on DMACII transfer mode.

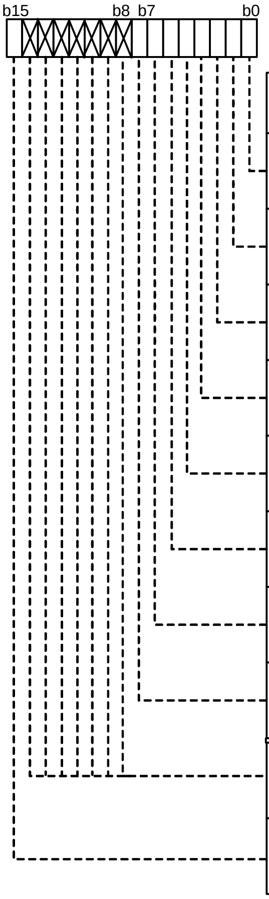
- Transfer mode (MOD)
  MOD is two-byte data and required to set transfer mode. Figure 14.3 shows a configuration for transfer mode.
- Transfer counter (COUNT)
  COUNT is two-byte data and required to set the number of transfer.
- Transfer source address (SADR)
  SADR is two-byte data and required to set a source memory address or immediate data.
- Operation address (OADR)
  OADR is two-byte data and required to set a memory address to be calculated. Set this data only when using the calculation transfer function.
- Transfer destination address (DADR)
  DADR is two-byte data and required to set a destination memory address.
- Chain transfer address (CADR)
  CADR is four-byte data and required to set the starting address of the DMACII index for the next transfer. Set this data only when using the chain transfer function.
- End-of-transfer interrupt address (IADR)
  IADR is four-byte data and required to set a jump address for end-of-transfer interrupt processing. Set this data only when using the end-of-transfer interrupt.

The abbreviations shown in parentheses( ) for each parameter are used in this section.

Table 14.2 DMACII Index Configuration in Transfer Mode

Transfer data	Memory-to-Memory Transfer/ Immediate Data Transfer				Calculation Transfer				Multiple Transfer
Chain transfer	Not used	Used	Not used	Used	Not used	Used	Not used	Used	Cannot used
End-of-Transfer Interrupt	Not used	Not used	Used	Used	Not used	Not used	Used	Used	Cannot used
DMAC II index	MOD COUNT SADR DADR 8 bytes	MOD COUNT SADR DADR CADR0 CADR1 12 bytes	MOD COUNT SADR DADR IADR0 IADR1 12 bytes	MOD COUNT SADR DADR CADR0 CADR1 IADR0 IADR1 16 bytes	MOD COUNT SADR OADR DADR 10 bytes	MOD COUNT SADR OADR DADR CADR0 CADR1 14 bytes	MOD COUNT SADR OADR DADR IADR0 IADR1 14 bytes	MOD COUNT SADR OADR DADR CADR0 CADR1 IADR0 IADR1 18 bytes	MOD COUNT SADR1 DADR1 SADRi DADRi i = 1 to 7 max. 32 bytes (when i = 7)

Transfer Mode (MOD)[1]



Bit Symbol	Bit Name	Function (MULT = 0)	Function (MULT = 1)	RW
SIZE	Transfer unit select bit	0: 8 bits 1: 16 bits		RW
IMM	Transfer data select bit	0: Immediate data 1: Memory	Set to 1	RW
UPDS	Transfer source direction select bit	0: Fixed address 1: Incremented address		RW
UPDD	Transfer destination direction select bit	0: Fixed address 1: Incremented address		RW
OPER/ CNT0[2]	Calculation transfer function select bit	0: Not used 1: Used	b6 b5 b4 0 0 0: Do not set to this value 0 0 1: Once 0 1 0: Twice : : 1 1 0: 6 times 1 1 1: 7 times	RW
BRST/ CNT1[2]	Burst transfer select bit	0: Single transfer 1: Burst transfer		RW
INTE/ CNT2[2]	End-of-transfer interrupt select bit	0: Interrupt not used 1: Interrupt used		RW
CHAIN	Chain transfer select bit	0: Chain transfer not used 1: Chain transfer used	Set to 0	RW
– (b14-b8)	Unimplemented. Write 0. Read as undefined value.			–
MULT	Multiple transfer select bit	0: Multiple transfer not used	1: Multiple transfer used	RW

NOTES:
1. MOD must be located in the RAM.
2. When the MULT bit is set to 0, bits 6 to 4 function as bits OPER, BRST, and INTE. When the MULT bit is set to 1, bits 6 to 4 function as bits CNT2 to CNT0.

Figure 14.3 MOD

14.1.3 Interrupt Control Register for the Peripheral Function

To use the peripheral function interrupt as a DMACII request source, set bits ILVL2 to ILVL0 to 111b (level 7).

14.1.4 Relocatable Vector Table for the Peripheral Function

Set the starting address of the DMACII index in an interrupt vector for the peripheral function interrupt used as a DMACII request source. When using the chain transfer, the relocatable vector table must be located in the RAM.

14.2 DMACII Performance

The DMACII function is selected by setting the DMAII bit to 1 (interrupt priority level 7 is used for DMACII transfer). DMACII transfer request is generated by interrupt requests from any peripheral function with bits ILVL2 to ILVL0 set to 111b (level 7). These peripheral function interrupt requests are used as DMACII transfer requests and the peripheral function interrupts cannot be used.
When an interrupt request with bits ILVL2 to ILVL0 set to 111b (level 7) is generated, DMACII is activated regardless of the I flag and IPL settings.

14.3 Transfer Data

DMACII transfers data in 8-bit units or 16-bit units.

- Memory-to-memory transfer: data is transferred from a given memory location in the 64-Kbyte space (addresses 00000h to 0FFFFh) to another given memory location in the same space.
- Immediate data transfer: immediate data is transferred to a given memory location in the 64-Kbyte space.
- Calculation transfer: two 8-bit or two 16-bit data are added together and the result is transferred to a given memory location in the 64-Kbyte space.

When a 16-bit data is transferred to a destination address 0FFFFh, it is transferred to addresses 0FFFFh and 10000h. Likewise, when a source address is 0FFFFh, a 16-bit data in addresses 0FFFFh and 10000h is transferred to a given destination address.
The actual transferable space varies depending on internal RAM capacity. Refer to **Figure 3.1** for the internal memory.

14.3.1 Memory-to-memory Transfer

Data transfer between any two memory locations in the 64-Kbyte space can be:

- a transfer from a fixed address to another fixed address;
- a transfer from a fixed address to an incremented address;
- a transfer from an incremented address to a fixed address;
- a transfer from an incremented address to another incremented address.

When an incremented address is selected, DMACII increments an address after every transfer for the following transfer. In a 8-bit data transfer, a transfer address is incremented by one. In a 16-bit data transfer, a transfer address is incremented by two.
When a source or destination address exceeds 0FFFFh as a result of address incrementation, the source or destination address returns to 00000h and continues incrementation. Maintain source and destination address at 0FFFFh or below.

14.3.2 Immediate Data Transfer

DMACII transfers immediate data to a given memory location. A fixed or incremented address can be selected as a destination address. Store immediate data into SADR. To transfer an 8-bit immediate data, write data in the low-order byte of SADR. (The high-order byte is ignored.)

14.3.3 Calculation Transfer

After two memory data, or an immediate data and a memory data, are added together, DMACII transfers the calculated result to a given memory location. Set a memory address or immediate data to be calculated in SADR. Set another memory address to be calculated in OADR. To use a “memory + memory” calculation transfer, a fixed or incremented address can be selected as a source or destination address. If a source address is incremented, an operation address also becomes incremented. To use an “immediate data + memory” calculation transfer, a fixed or incremented address can be selected as a destination address.

14.4 Transfer Modes

In DMACII, a single transfer, burst transfer, and multiple transfer are available. The BRST bit in MOD selects either a single transfer or burst transfer, and the MULT bit in MOD selects a multiple transfer. COUNT determines how many transfers occur. No transfer occurs when COUNT is set to 0000h.

14.4.1 Single Transfer

For one transfer request, DMACII transfers an 8-bit or 16-bit data once. When an incremented address is selected for a source or destination address, DMACII increments the address after every transfer for the following transfer.

COUNT is decremented every time a transfer occurs. If using the end-of-transfer interrupt, an interrupt occurs when COUNT reaches zero.

14.4.2 Burst Transfer

For one transfer request, DMACII continuously transfers data the number of times determined by COUNT. COUNT is decremented every time DMACII transfers one transfer unit, and when it reaches zero, a burst transfer is completed. If using the end-of-transfer interrupt, an interrupt occurs at the end of the burst transfer. While the burst transfer is taking place, no interrupt can be acknowledged.

14.4.3 Multiple Transfer

When using the multiple transfer, select the memory-to-memory transfer. For one transfer request, DMACII transfers data multiple times. Bits CNT2 to CNT0 in MOD selects the number of transfers from 001b (once) to 111b (7 times). Do not set bits CNT2 to CNT0 to 000b.

Source and destination addresses enough for all transfers must be allocated alternately in addresses following MOD and COUNT in DMACII index.

While the transfers are taking place the number of times set using bits CNT2 to CNT0, no interrupt can be acknowledged. When the multiple transfer is selected, a calculation transfer, burst transfer, chain transfer, and end-of-transfer interrupt cannot be used.

14.5 Chain Transfer

The chain transfer can be selected with the CHAIN bit in MOD.
The chain transfer is performed as follows.

(1) Transfer occurs in response to an interrupt request from a peripheral function and is performed according to the contents of the DMACII index at the address specified by the interrupt vector. For one transfer request, either a single transfer or burst transfer selected by the BRST bit in MOD occurs.

(2) When COUNT reaches zero, the interrupt vector in (1) is replaced with the address written in CADR1 and CADR0. The end-of-transfer interrupt occurs after the replacement, if the INTE bit in MOD is set to 1.

(3) When the next DMACII transfer request is generated, the transfer is performed according to the contents of the DMACII index specified by the interrupt vector which has been replaced in (2).

Figure 14.4 shows the relocatable vector and DMACII index when using the chain transfer.
For the chain transfer, the relocatable vector table must be located in the RAM.

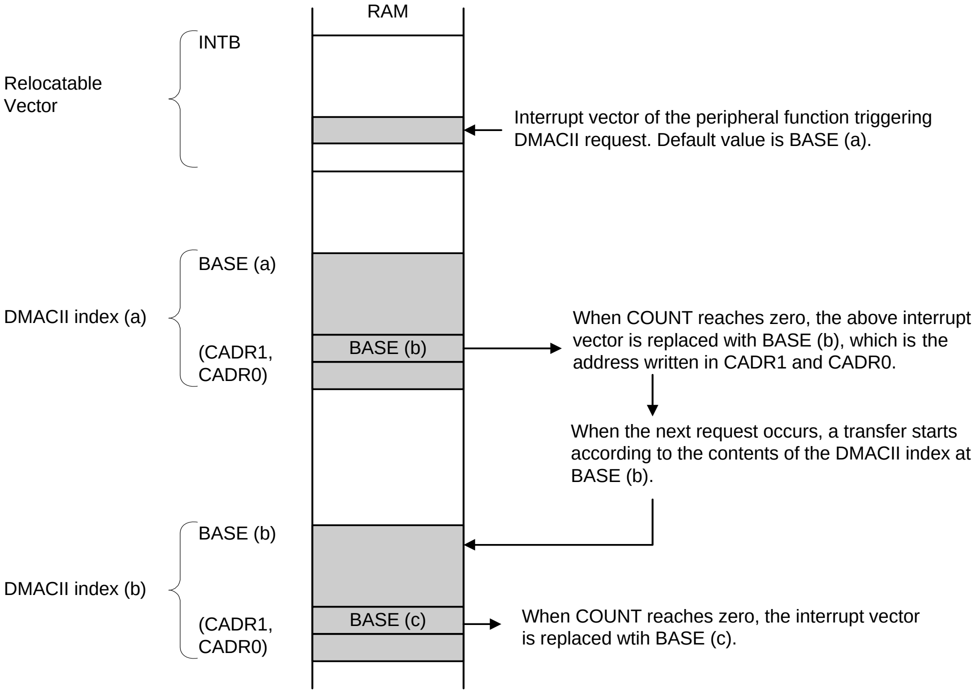

RAM
INTB
Relocatable Vector
Interrupt vector of the peripheral function triggering DMACII request. Default value is BASE (a).
BASE (a)
DMACII index (a)
(CADR1, CADR0)
BASE (b)
When COUNT reaches zero, the above interrupt vector is replaced with BASE (b), which is the address written in CADR1 and CADR0.
When the next request occurs, a transfer starts according to the contents of the DMACII index at BASE (b).
BASE (b)
DMACII index (b)
(CADR1, CADR0)
BASE (c)
When COUNT reaches zero, the interrupt vector is replaced wtih BASE (c).

Figure 14.4 Relocatable Vector and DMACII Index When using the Chain Transfer

14.6 End-of-Transfer Interrupt

The end-of-transfer interrupt can be selected with the INTE bit in MOD. Set the starting address of the end-of-transfer interrupt routine in IADR1 and IADR0. The end-of-transfer interrupt occurs when COUNT reaches zero.

14.7 Execution Time

DMACII execution time is calculated by the following equations (single-speed mode):

Multiple transfers: t [bus clock] = 21+ (11 + b + c) × k
Other than multiple transfers: t [bus clock] = 6 + (26 + a + b + c + d) × m + (4 + e) × n

a: If IMM = 0 (source is immediate data), a = 0; if IMM = 1 (source is data in memory location), a = -1.
b: If UPDS = 1 (source address is incremented), b = 0; if UPDS = 0 (source address is fixed), b = 1.
c: If UPDD = 1 (destination address is incremented), c = 0; if UPDD = 0 (destination address is fixed), c = 1.
d: If OPER = 0 (calculation function is not selected), d = 0;
if OPER = 1 (calculation function is selected) and UPDS = 0 (source is immediate data or fixed address in memory location), d = 7;
if OPER = 1 (calculation function is selected) and UPDS = 1 (source is incremented address in memory location), d = 8.
e: If CHAIN = 0 (chain transfer is not selected), e = 0; if CHAIN = 1 (chain transfer is selected), e = 4.
m: If BRST = 0 (single transfer), m = 1; if BRST = 1 (burst transfer), m = a value set in COUNT.
n: If COUNT = 1, n = 0; if COUNT = 2 or more, n = 1.
k: The number of transfers set in bits CNT2 to CNT0 in MOD.

The above equations are approximations. The execution time varies depending on CPU state, bus wait states, and DMACII index allocation.
The first instruction of the end-of-transfer interrupt routine is executed in the eighth bus clock after the DMACII transfer is completed.

Conditions of the example below:
-memory-to-memory transfer (a = -1)
-incremented source address (b = 0)
-fixed destination address (c = 1)
-no calculation function (d = 0)
-no chain transfer (e = 0)
-single transfer (m = 1)
-the end-of-transfer interrupt (transfer counter = 2) occurs

First DMACII transfer t = 6 + 26 x 1 + 4 x 1 = 36 bus clocks
Second DMACII transfer t = 6 + 26 x 1 + 4 x 0 = 32 bus clocks

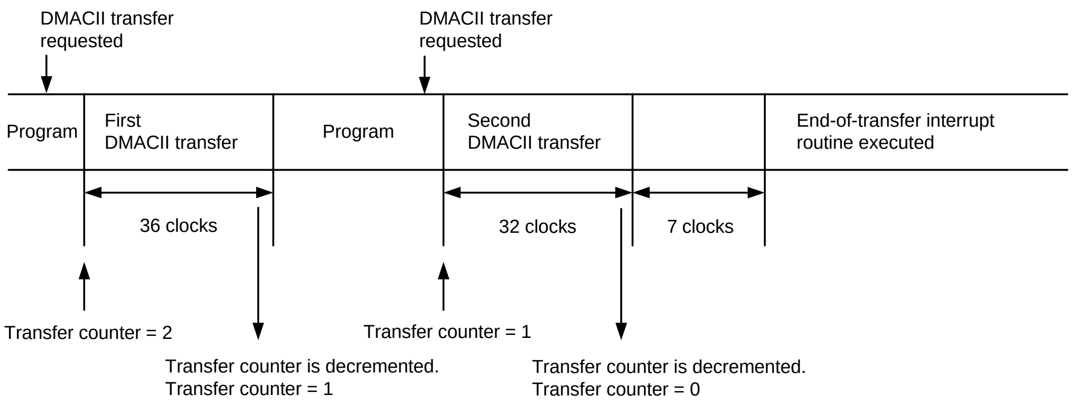


Figure 14.5 Transfer Time

When a DMACII transfer request is generated simultaneously with another request having a higher priority (e.g., $\overline{\text{NMI}}$ or watchdog timer), the interrupt with higher priority is acknowledged first, and the pending DMACII transfer starts after the interrupt sequence of the higher priority interrupt has been completed.

15. Timers

The M32C/8B Group has eleven 16-bit timers, and they are separated into five timer A and six timer B based on their functions. Individual timers function independently. The count source for each timer is used to operate the timer for counting and reloading, etc.
Figures 15.1 and 15.2 show block diagrams of timer A and timer B configurations.

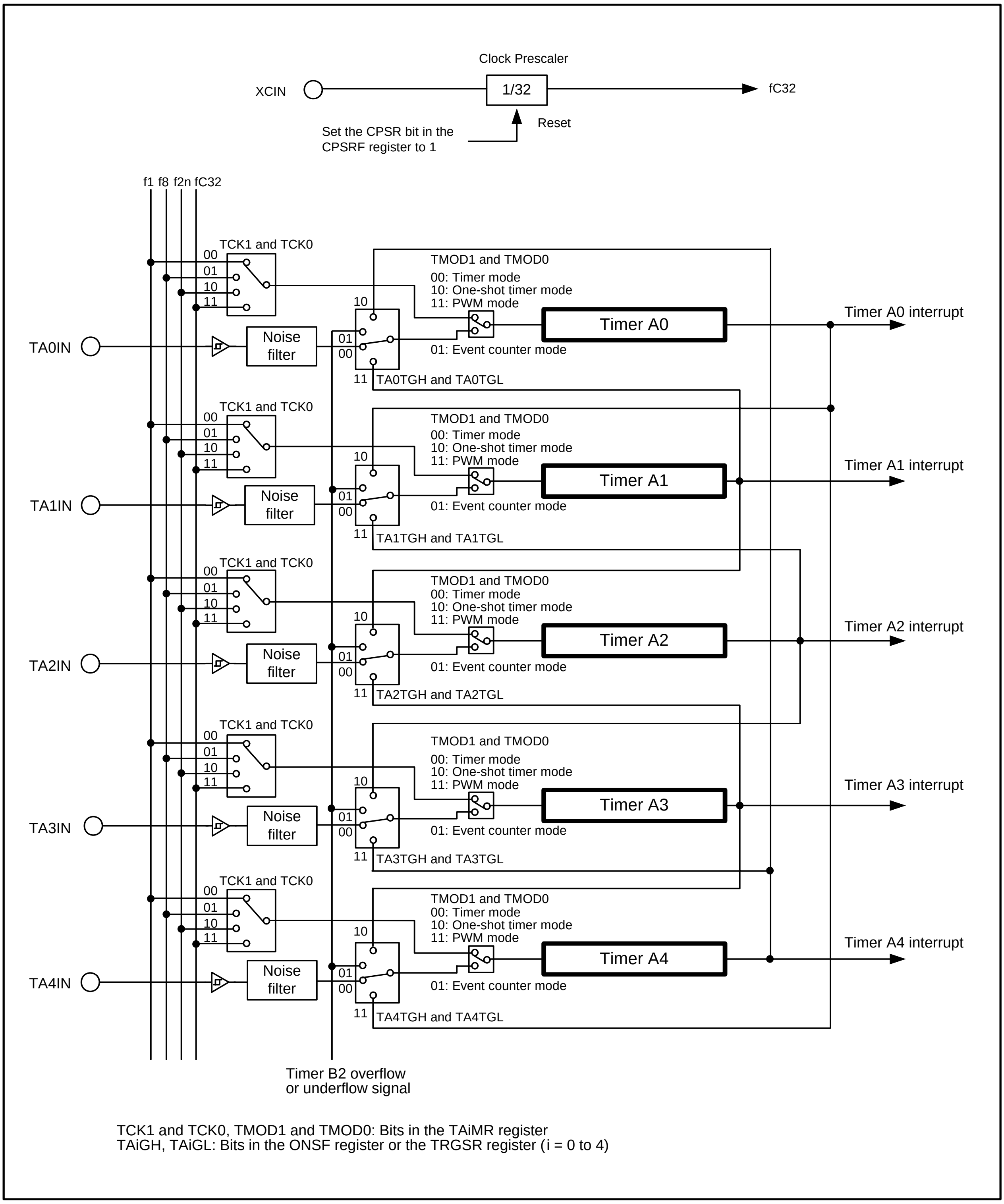
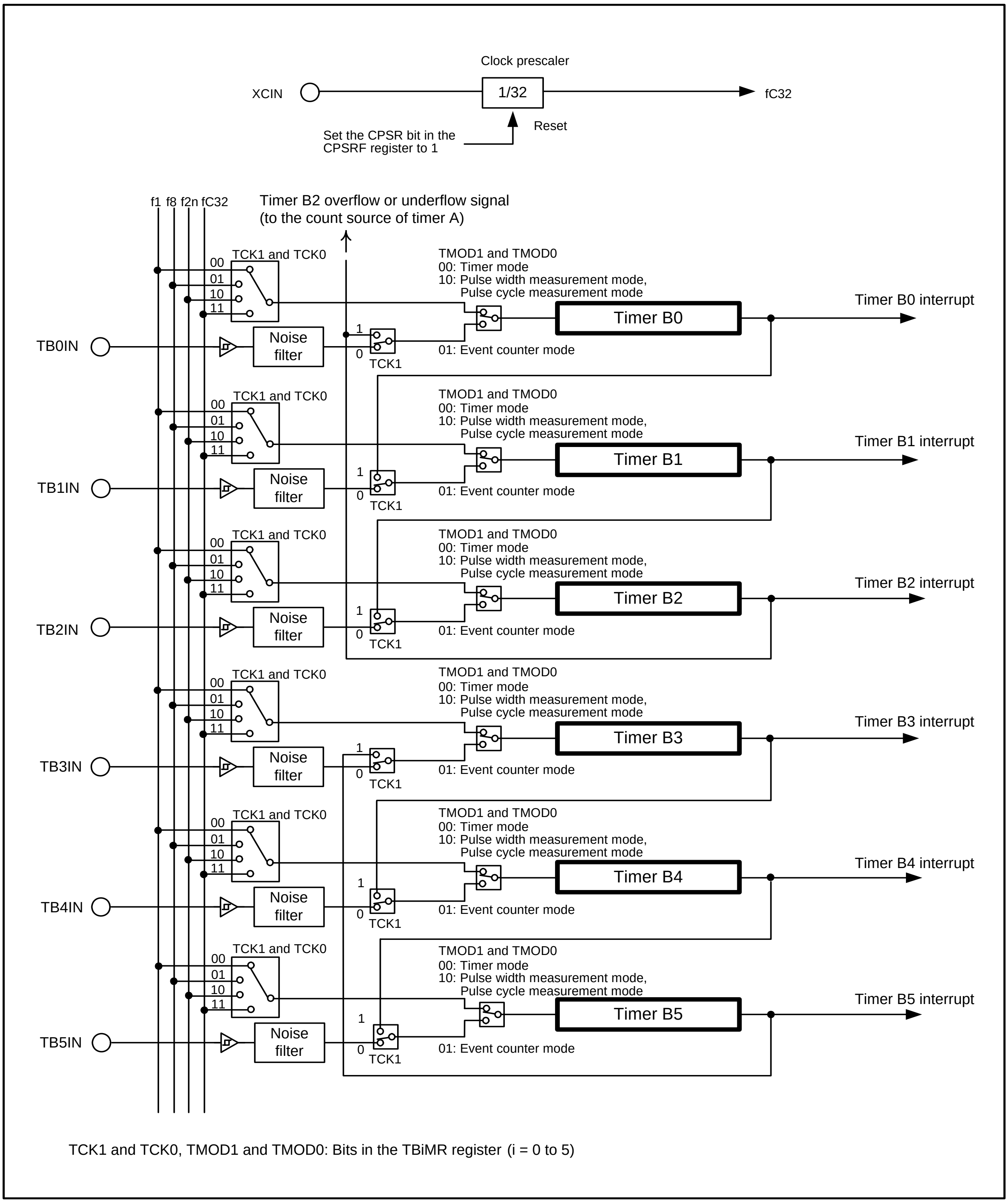


Figure 15.1 Timer A Configuration



TCK1 and TCK0, TMOD1 and TMOD0: Bits in the TBiMR register (i = 0 to 5)

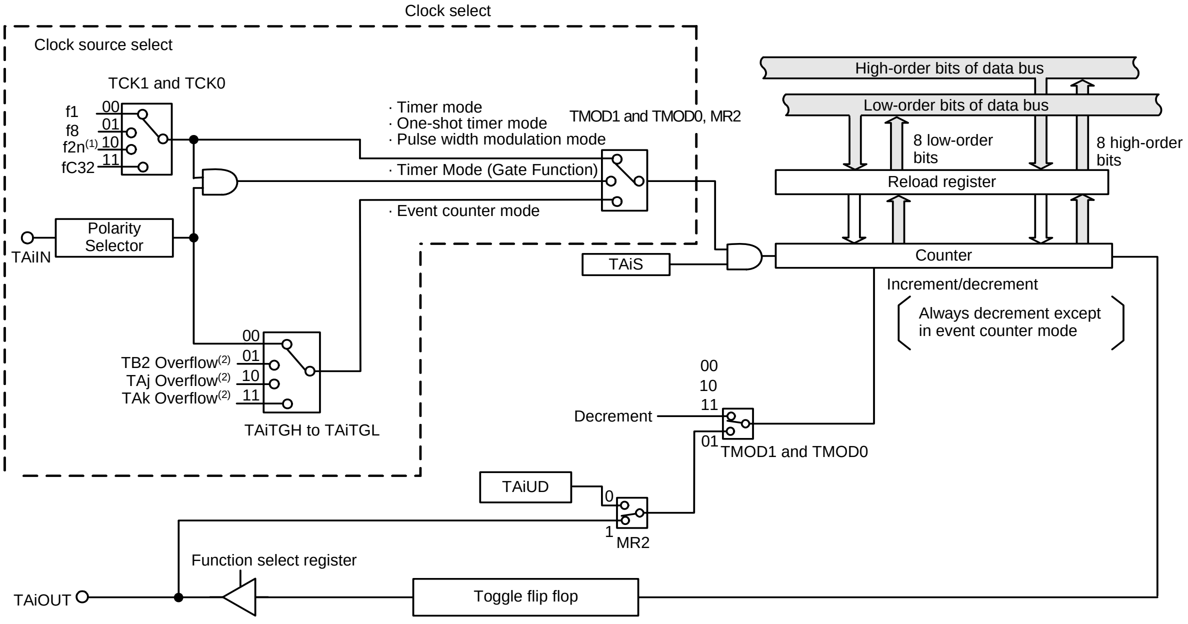
Figure 15.2 Timer B Configuration

15.1 Timer A

Timer A contains the following four modes. Except in event counter mode, all timers A0 to A4 have the same functionality. Bits TMOD1 and TMOD0 in the TAiMR register (i = 0 to 4) determine which mode is used.

- Timer mode: The timer counts the internal count source.
- Event counter mode: The timer counts overflow/underflow signal of another timer or the external pulses.
- One-shot timer mode: The timer operates only once for one trigger.
- Pulse width modulation mode: The timer continuously outputs given pulse widths.

Figure 15.3 shows a block diagram of timer A. Figures 15.4 to 15.13 show the registers associated with timer A. Table 15.1 lists TAiOUT pin settings to use in output mode. Table 15.2 lists TAiIN and TAiOUT pin settings to use in input mode.



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

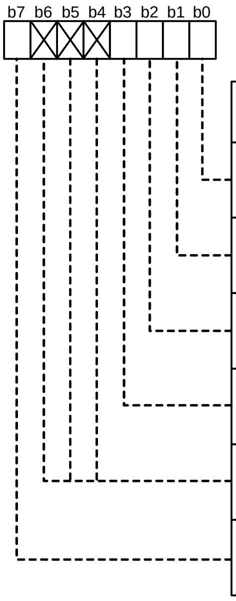
NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Overflow signal or underflow signal.

TAi	Addresses	TAj	TAk
Timer A0	0347h 0346h	Timer A4	Timer A1
Timer A1	0349h 0348h	Timer A0	Timer A2
Timer A2	034Bh 034Ah	Timer A1	Timer A3
Timer A3	034Dh 034Ch	Timer A2	Timer A4
Timer A4	034Fh 034Eh	Timer A3	Timer A0

TCK1 and TCK0, TMOD1 and TMOC0, MR2 and MR1: Bits in the TAiMR register
TAiTGH to TAiTGL: Bits in the ONSF register if i = 0 or bits in the TRGSR register if i = 1 to 4
TAiS: Bit in the TABSR register
TAiUD: Bit in the UDF register

Figure 15.3 Timer A Block Diagram

Count Source Prescaler Register



Symbol	Address	After Reset[2]
TCSPR	035Fh	0XXX 0000b

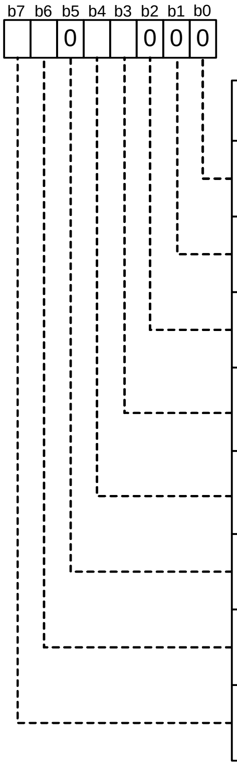
Bit Symbol	Bit Name	Function	RW
CNT0	Divide ratio select bits[1]	If the setting value is *n*, *f2n* is the main clock, on-chip oscillator, or PLL clock divided by *2n*. No division if *n* = 0	RW
CNT1			RW
CNT2			RW
CNT3			RW
– (b6-b4)	Reserved bits	Read as undefined value	–
CST	Operation enable bit	0: Divider stops 1: Divider operates	RW

NOTES:

1. Set the CST bit to 0 before bits CNT3 to CNT0 are rewritten.
2. The TCSPR register maintains values set before reset, even after software reset or watchdog timer reset has been performed.

Figure 15.4 TCSPR Register

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



Symbol	Address	After Reset
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 0: Timer mode	RW
TMOD1			RW
– (b2)	Reserved bit	Set to 0	RW
MR1	Gate function select bits	b4 b3 0 0: } Gate function disabled 0 1: } (TAiIN pin is a programmable I/O port) 1 0: Timer counts only while an "L" signal is input to the TAiIN pin 1 1: Timer counts only while an "H" signal is input to the TAiIN pin	RW
MR2			RW
MR3	Set to 0 in timer mode		RW
TCK0	Count source select bits	b7 b6 0 0: f1 0 1: f8 1 0: f2n(1) 1 1: fC32	RW
TCK1			RW

NOTE:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divided-by-2n (n = 1 to 15). To select f2n, set the CST bit in the TCSPR register to 1 before setting bits TCK1 and TCK0 to 10b.

Figure 15.5 TA0MR to TA4MR Registers in Timer Mode

Timer Ai Mode Register (i = 0 to 4)(Event Counter Mode)

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

Symbol	Address	After Reset
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	00h

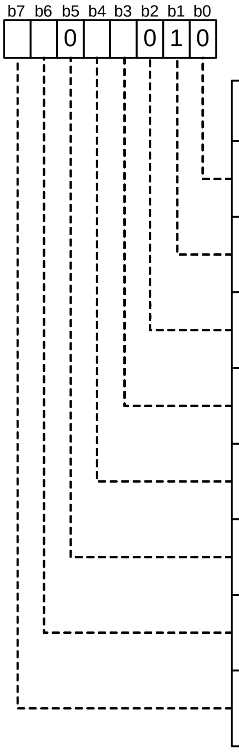
Bit Symbol	Bit Name	Function (When not processing two-phase pulse signals)	Function (When processing two-phase pulse signals)	RW
TMOD0	Operating mode select bits	b1 b0 0 1: Event counter mode[1]		RW
TMOD1				RW
– (b2)	Reserved bit	Set to 0		RW
MR1	Count polarity select bit[2]	0: Falling edges of an external signal counted 1: Rising edges of an external signal counted	Set to 0	RW
MR2	Increment/decrement switching source select bit	0: UDF regisstser setting 1: Signal applied to the TAiOUT pin[3]	Set to 1	RW
MR3	Set to 0 in event counter mode			RW
TCK0	Count operation type select bit	0: Reload 1: Free running		RW
TCK1	Two-phase pulse signal processing operation select bit[4,5]	Set to 0	0: Normal processing operation 1: Multiply-by-4 processing operation	RW

NOTES:

1. Bits TAiTGH and TAiTGL in the ONSF or TRGSR register determine a count source in event counter mode.
2. The MR1 bit is enabled only when counting external signals.
3. The counter decrements when an “L” signal is applied to the TAiOUT pin. The counter increments when an “H” signal is applied to the TAiOUT pin.
4. The TCK1 bit is enabled only in the TA3MR register. The TCK1 bit in registers TA0MR to TA2MR and TA4MR are disabled.
5. For two-phase pulse signal processing, set the TAjP bit in the UDF register (j = 2 to 4) to 1 (two-phase pulse signal processing function enabled). Also, set bits TAjTGH and TAjTGL in the TRGSR register to 00b (input to the TAjIN pin).

Figure 15.6 TA0MR to TA4MR Registers in Event Counter Mode

Timer Ai Mode Register (i = 0 to 4)(One-Shot Timer Mode)



Symbol	Address	After Reset
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	00h

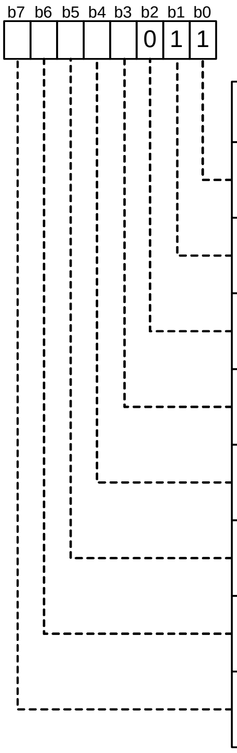
Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 1 0: One-shot timer mode	RW
TMOD1			RW
– (b2)	Reserved bit	Set to 0	RW
MR1	External trigger select bit[1]	0: Falling edge of signal applied to the TAiIN pin 1: Rising edge of signal applied to the TAiIN pin	RW
MR2	Trigger select bit	0: The TAiOS bit enabled 1: Selected by bits TAiTGH and TAiTGL	RW
MR3	Set to 0 in one-shot timer mode		RW
TCK0	Count source select bits	b7 b6 0 0: f1 0 1: f8 1 0: f2n[2] 1 1: fC32	RW
TCK1			RW

NOTES:

1. The MR1 bit is enabled only when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are set to 00b (input to the TAiIN pin).
   The MR1 bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (TB2 overflow or underflow), 10b (TAj (j = i - 1, except j = 4 if i = 0) overflow or underflow), or 11b (TAk (k = i + 1, except i = 4 if k = 0) overflow or underflow).
2. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15). To select f2n, set the CST bit in the TCSPR register to 1 before setting bits TCK1 and TCK0 to 10b.

Figure 15.7 TA0MR to TA4MR Registers in One-Shot Timer Mode

Timer Ai Mode Register (i = 0 to 4)(Pulse Width Modulation Mode)



Symbol	Address	After Reset
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	00h

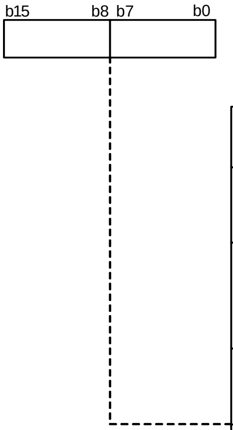
Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 1 1: Pulse width modulation (PWM) mode	RW
TMOD1			RW
– (b2)	Reserved bit	Set to 0	RW
MR1	External trigger select bit[1]	0: Falling edge of signal applied to the TAiIN pin 1: Rising edge of signal applied to the TAiIN pin	RW
MR2	Trigger select bit	0: The TAiS bit is enabled 1: Selected by bits TAiTGH and TAiTGL	RW
MR3	16/8-bit PWM mode select bit	0: Functions as 16-bit pulse width modulator 1: Functions as 8-bit pulse width modulator	RW
TCK0	Count source select bits	b7 b6 0 0: f1 0 1: f8 1 0: f2n [2] 1 1: fC32	RW
TCK1			RW

NOTES:

1. The MR1 bit is enabled only when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are set to 00b (input to the TAiIN pin).
   The MR1 bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (TB2 overflow or underflow), 10b (TAj (j = i - 1, except j = 4 if i = 0) overflow or underflow), or 11b (TAk (k = i + 1, except i = 4 if k = 0) overflow or underflow).
2. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15). To select f2n, set the CST bit in the TCSPR register to 1 before setting bits TCK1 and TCK0 to 10b.

Figure 15.8 TA0MR to TA4MR Registers in Pulse Width Modulation Mode

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



Symbol	Address	After Reset
TA0 to TA2	0347h - 0346h, 0349h - 0348h, 034Bh - 034Ah	Undefined
TA3, TA4	034Dh - 034Ch, 034Fh - 034Eh	Undefined

Mode	Function	Setting Range	RW
Timer mode	If a count source frequency is fj and the setting value of TAi register is n, the counter cycle is (n + 1) / fj	0000h to FFFFh	RW
Event counter mode	If the setting value is n, the count times are (FFFFh - n+1) when the counter increments, and (n+1) when the counter decrements(2)	0000h to FFFFh	RW
One-shot timer mode	If the setting value is n, the counter counts n times and stops.	0000h to FFFFh(3, 4)	WO
Pulse width modulation mode (16-bit PWM)	If a count source frequency is fj and the setting value of the TAi register is n, PWM cycle: (2^{16} - 1) / fj "H" width of PWM pulse: n / fj	0000h to FFFEh(3, 5)	WO
Pulse width modulation mode (8-bit PWM)	If a count source frequency is fj, the setting value of high-order bits in the TAi register is n, and the setting value of low-order bits in the TAi register is m, PWM cycle: (2^8 -1) x (m+1) / fj "H" width of PWM pulse: (m+1) n / fj	00h to FEh(3, 6) (High-order address bits) 00h to FFh(3, 6) (Low-order address bits)	WO

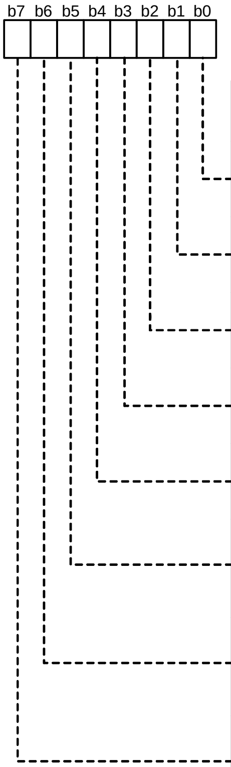
fj: f1, f8, f2n, fC32

NOTES:

1. Read and write this register in 16-bit units.
2. The TAi register counts external pulses or another timer overflows or underflows.
3. Read-modify-write instructions cannot be used to set the TAi register. Refer to **Usage Notes** for details.
4. When the TAi register is set to 0000h, the counter does not start and a timer Ai interrupt request is not generated.
5. When the TAi register is set to 0000h, the pulse width modulator does not operate and the TAiOUT pin output is held "L". A timer Ai interrupt request is not generated. When the TAi register is set to FFFFh, the pulse width modulator does not operate and the TAiOUT pin output is held "H". A timer Ai interrupt request is not generated.
6. When 8 high-order bits are set to 00h, the pulse width modulator does not operate and the TAiOUT pin output is held "L". A timer Ai interrupt request is not generated. When 8 high-order bits are set to FFh, the pulse width modulator does not operate and the TAiOUT pin output is held "H". A timer Ai interrupt request is not generated.

Figure 15.9 TA0 to TA4 Registers

Up/Down Select Register[1]



Symbol
UDF

Address
0344h

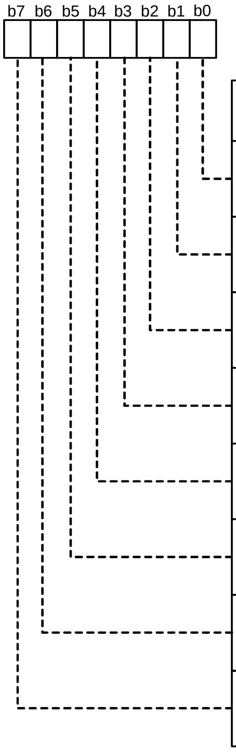
After Reset
00h

Bit Symbol	Bit Name	Function	RW
TA0UD	Timer A0 up/down select bit[2]	0: Decrement 1: Increment	RW
TA1UD	Timer A1 up/down select bit[2]	0: Decrement 1: Increment	RW
TA2UD	Timer A2 up/down select bit[2]	0: Decrement 1: Increment	RW
TA3UD	Timer A3 up/down select bit[2]	0: Decrement 1: Increment	RW
TA4UD	Timer A4 up/down select bit[2]	0: Decrement 1: Increment	RW
TA2P	Timer A2 two-phase pulse signal processing function select bit[3]	0: Two-phase pulse signal processing function disabled 1: Two-phase pulse signal processing function enabled	WO
TA3P	Timer A3 two-phase pulse signal processing function select bit[3]	0: Two-phase pulse signal processing function disabled 1: Two-phase pulse signal processing function enabled	WO
TA4P	Timer A4 two-phase pulse signal processing function select bit[3]	0: Two-phase pulse signal processing function disabled 1: Two-phase pulse signal processing function enabled	WO

NOTES:
1. Read-modify-write instructions cannot be used to set the UDF register. Refer to **Usage Notes** for details.
2. This bit is enabled when the MR2 bit in the TAiMR register (i = 0 to 4) is set to 0 (the UDF register causes increment/decrement switching) in event counter mode.
3. Set these bits to 0 when not using the two-phase pulse signal processing function.

Figure 15.10 UDF Register

Trigger Select Register



Symbol	Address	After Reset
TRGSR	0343h	00h

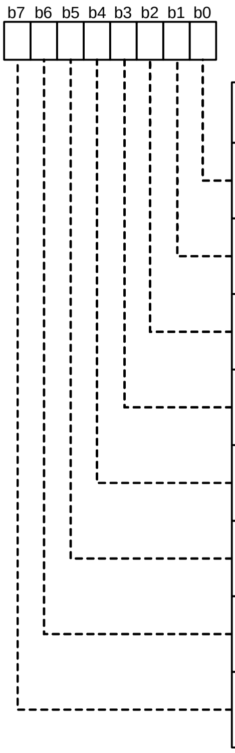
Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 trigger select bits	b1 b0 0 0: Input to the TA1IN pin selected 0 1: TB2 overflows selected(1) 1 0: TA0 overflows selected(1) 1 1: TA2 overflows selected(1)	RW
TA1TGH			RW
TA2TGL	Timer A2 trigger select bits	b3 b2 0 0: Input to the TA2IN pin selected 0 1: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

NOTE:
1. Overflow or underflow.

Figure 15.11 TRGSR Register

Count Start Register

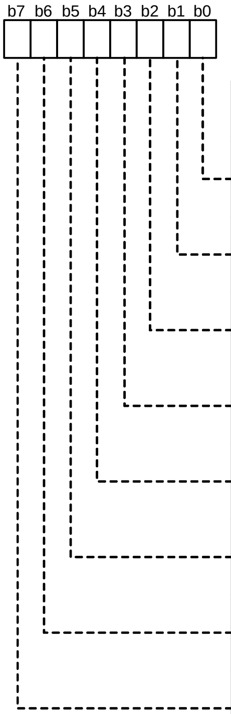
Symbol	Address	After Reset
TABSR	0340h	00h



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

Figure 15.12 TABSR Register

One-Shot Start Register



Symbol	Address	After Reset
ONSF	0342h	00h

Bit Symbol	Bit Name	Function	RW
TA0OS	Timer A0 one-shot start bit[1]	0: In an idle state 1: Timer starts	RW
TA1OS	Timer A1 one-shot start bit[1]	0: In an idle state 1: Timer starts	RW
TA2OS	Timer A2 one-shot start bit[1]	0: In an idle state 1: Timer starts	RW
TA3OS	Timer A3 one-shot start bit[1]	0: In an idle state 1: Timer starts	RW
TA4OS	Timer A4 one-shot start bit[1]	0: In an idle state 1: Timer starts	RW
TAZIE	Z-phase input enable bit	0: Z-phase input disabled 1: Z-phase input enabled	RW
TA0TGL	Timer A0 trigger select bits	b7 b6 0 0: Input to the TA0IN pin selected 0 1: TB2 overflows selected[2] 1 0: TA4 overflows selected[2] 1 1: TA1 overflows selected[2]	RW
TA0TGH			RW

NOTES:

1. Read as 0.
2. Overflow or underflow.

Figure 15.13 ONSF Register

Table 15.1 TAiOUT Pin Settings in Output Mode (i = 0 to 4)

Port	Function	Bit Setting		
		PSC Register	PSL1, PSL2 Registers	PS1, PS2 Registers[1]
P7_0[2]	TA0OUT	_	PSL1_0 = 1	PS1_0 = 1
P7_2	TA1OUT	–	PSL1_2 = 1	PS1_2 = 1
P7_4	TA2OUT	PSC_4 = 0	PSL1_4 = 0	PS1_4 = 1
P7_6	TA3OUT	–	PSL1_6 = 1	PS1_6 = 1
P8_0	TA4OUT	–	PSL2_0 = 0	PS2_0 = 1

NOTES:

1. Set registers PS1and PS2 after setting registers PSC, PSL1, and PSL2.
2. P7_0 is an N-channel open drain output port.

Table 15.2 TAiIN and TAiOUT Pin Settings in Input Mode (i = 0 to 4)

Port	Function	Bit Setting	
		PD7, PD8 Registers	PS1, PS2 Registers
P7_0	TA0OUT	PD7_0 = 0	PS1_0 = 0
P7_1	TA0IN	PD7_1 = 0	PS1_1 = 0
P7_2	TA1OUT	PD7_2 = 0	PS1_2 = 0
P7_3	TA1IN	PD7_3 = 0	PS1_3 = 0
P7_4	TA2OUT	PD7_4 = 0	PS1_4 = 0
P7_5	TA2IN	PD7_5 = 0	PS1_5 = 0
P7_6	TA3OUT	PD7_6 = 0	PS1_6 = 0
P7_7	TA3IN	PD7_7 = 0	PS1_7 = 0
P8_0	TA4OUT	PD8_0 = 0	PS2_0 = 0
P8_1	TA4IN	PD8_1 = 0	PS2_1 = 0

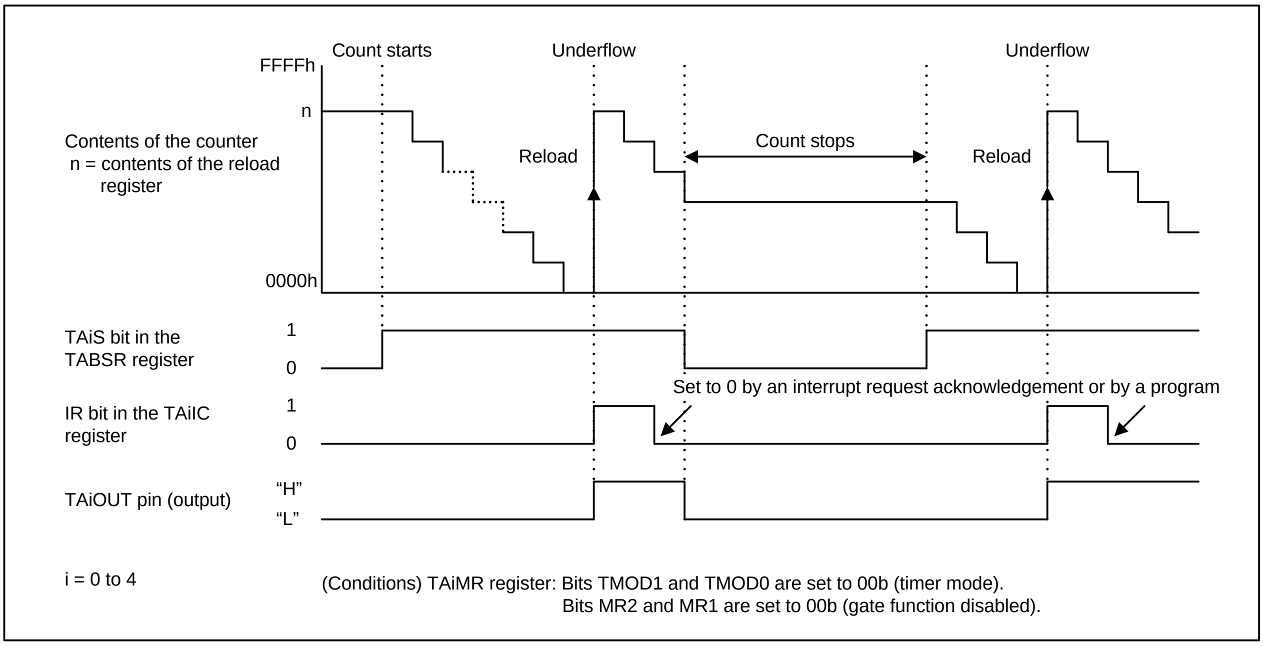
15.1.1 Timer Mode

In timer mode, the timer counts an internally generated count source.
Table 15.3 lists specifications of timer mode. Figure 15.14 shows a timer mode operation (Timer A).

Table 15.3 Specifications of Timer Mode

Item	Specification
Count source	f1, f8, f2n(1), fC32
Count operation	• Counter decrements When the timer underflows, the contents of the reload register are reloaded into the counter and the count continues.
Counter cycle	$\frac{n+1}{fj}$ fj: count source frequency n: setting value of the TAi register (i = 0 to 4), 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is set to 1 (count starts)
Count stop condition	The TAiS bit is set to 0 (count stops)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Input for gate function
TAiOUT pin function	Pulse output
Read from timer	A read from the TAi register returns a counter value
Write to timer	• A write to the TAi register while the count is stopped: The value is written to both the reload register and the counter. • A write to the TAi register while counting: The value is written to the reload register (It is transferred to the counter at the next reload timing).(2)
Selectable function	• Gate function A signal applied to the TAiIN pin determines whether the count starts or stops. • Pulse output function The polarity of the TAiOUT pin is inverted whenever the timer underflows. The TAiOUT pin outputs an “L” signal while the TAiS bit is 0 (count stops).

NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Wait for one or more count source cycles to write after the count starts.


Count starts
Underflow
Underflow
FFFFh
n
Contents of the counter
n = contents of the reload register
Reload
Count stops
Reload
0000h
TAiS bit in the TABSR register
1
0
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TAiIC register
1
0
TAiOUT pin (output)
“H”
“L”
i = 0 to 4

(Conditions) TAiMR register: Bits TMOD1 and TMOD0 are set to 00b (timer mode).
Bits MR2 and MR1 are set to 00b (gate function disabled).

Figure 15.14 Operation in Timer Mode (Timer A)

15.1.2 Event Counter Mode

In event counter mode, the timer counts overflows/underflows of another timer, or the external pulse input. Timers A2, A3, and A4 can count externally generated two-phase signals.

Table 15.4 lists specifications of event counter mode when not handling two-phase pulse signals.

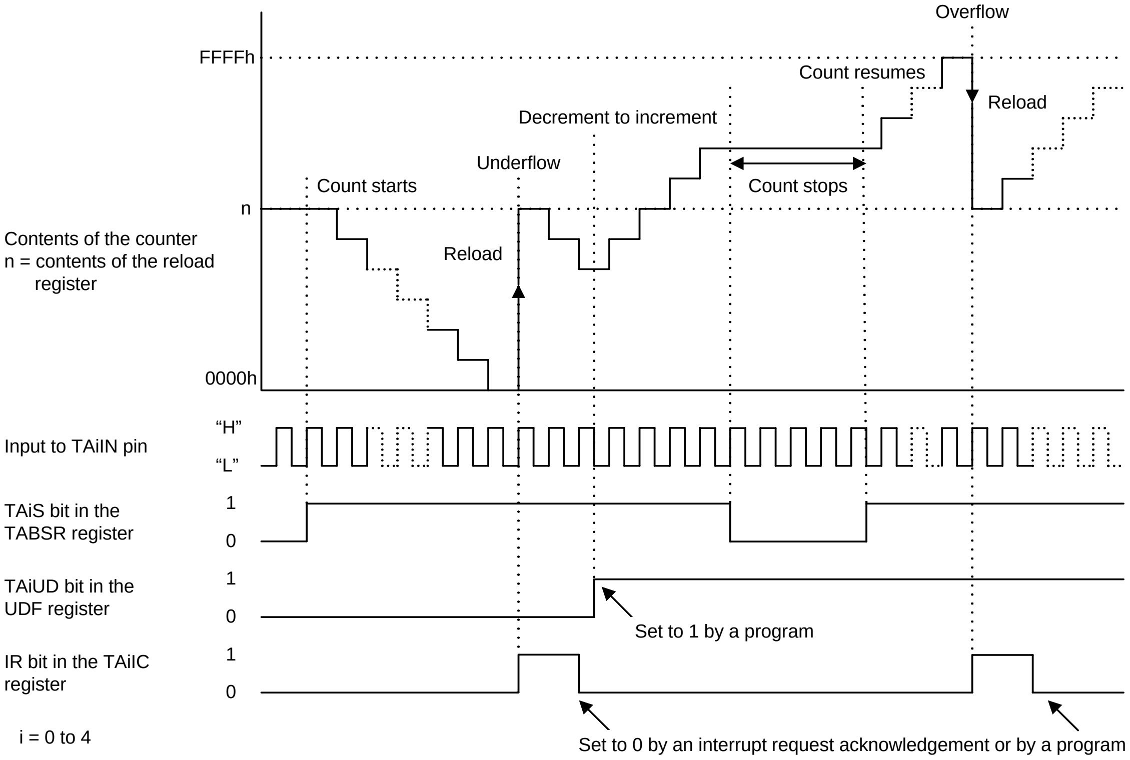
Table 15.5 lists specifications of event counter mode when handling two-phase pulse signals with timers A2, A3, and A4. Figure 15.15 shows a event counter mode operation when not handling two-phase pulse signals. Figure 15.16 shows a event counter mode operation when handling two-phase pulse signals with timers A2, A3, and A4.

Table 15.4 Specifications of Event Counter Mode When Not Handling Two-Phase Pulse Signals

Item	Specification
Count source	• External signal applied to the TAiIN pin (i = 0 to 4) (valid edge is selectable by a program) • Timer B2 overflows or underflows • Timer Aj overflows or underflows (j = i - 1, except j = 4 if i = 0) • Timer Ak overflows or underflows (k = i + 1 except k = 0 if i = 4)
Count operation	• Count direction (increment or decrement) can be selected by external signal or by a program. • Reload/Free-run type can be selected. Reload function: The contents of the reload register are reloaded into the counter and the count continues when the timer underflows or overflows. Free-running function: The counter continues running without reloading when the timer underflows or overflows.
Number of counting	(FFFFh - n + 1): when incrementing n + 1: when decrementing n: setting value of the TAi register, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is set to 1 (count starts)
Count stop condition	The TAiS bit is set to 0 (count stops)
Interrupt request generation timing	When the timer overflows or underflows
TAiIN pin function	Count source input
TAiOUT pin function	Pulse output, or input to select the count direction
Read from timer	A read from the TAi register returns a counter value
Write to timer	• A write to the TAi register while the count is stopped: The value is written to both the reload register and the counter. • A write to the TAi register while counting: The value is written to the reload register (It is transferred to the counter at the next reload timing).[1]
Selectable function	Pulse output function The polarity of the TAiOUT pin is inverted whenever the timer overflows or underflows. The TAiOUT pin outputs “L” signal while the TAiS bit is 0 (count stops).

NOTE:

1. Wait for one or more count source cycles to write after the count starts.


Overflow
FFFFh
Count resumes
Reload
Decrement to increment
Underflow
Count starts
Count stops
n
Contents of the counter
n = contents of the reload register
Reload
0000h
"H"
Input to TAiIN pin
"L"
TAiS bit in the TABSR register
1
0
TAiUD bit in the UDF register
1
0
Set to 1 by a program
IR bit in the TAiIC register
1
0
i = 0 to 4
Set to 0 by an interrupt request acknowledgement or by a program

(Conditions) TAiMR register: Bits TMOD1 and TMOD0 are set to 01b (event counter mode)
The MR1 bit is set to 1 (rising edges of an external signal counted)
The MR2 bit is set to 0 (UDF register setting)
Bits TCK1 to TCK0 bit are set to 00b (reload)

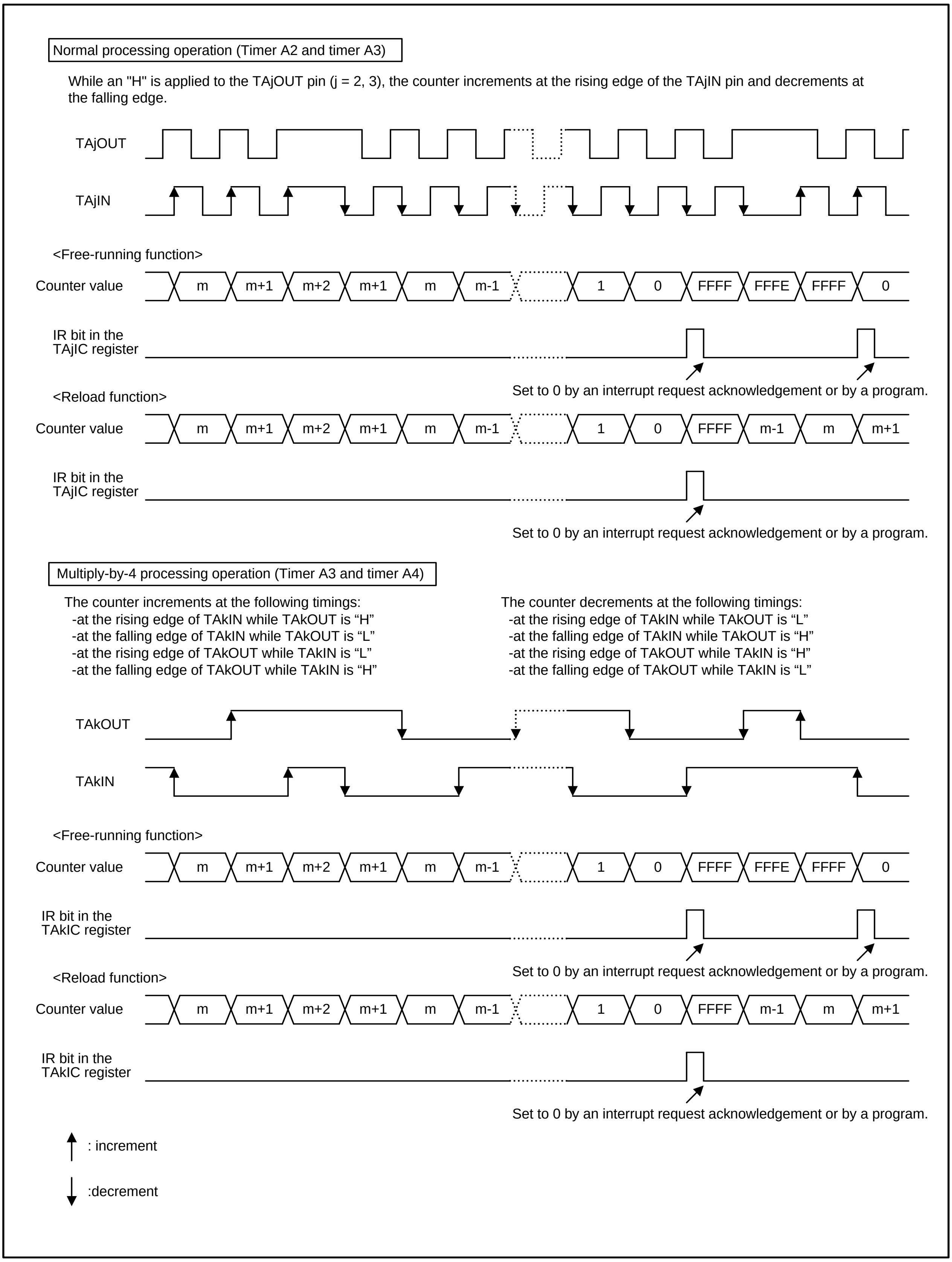
Figure 15.15 Operation in Event Counter Mode When Not Handling Two-Phase Pulse Signals

Table 15.5 Specifications of Event Counter Mode When Handling Two-Phase Pulse Signals on Timers A2, A3, and A4

Item	Specification
Count source	Two-phase pulse signals applied to pins TAiIN and TAiOUT (i = 2 to 4)
Count operation	• Count direction (increment or decrement) is set by a two-phase pulse signal. • Reload/Free-run type can be selected. Reload function: The contents of the reload register are reloaded into the counter and the count continues when the timer underflows or overflows. Free-running function: The counter continues running without reloading when the timer underflows or overflows.
Number of counting	(FFFFh - n + 1): when incrementing n + 1: for decrementing n: setting value of the TAi register, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR Register is set to 1 (count starts)
Count stop condition	The TAiS bit is set to 0 (count stops)
Interrupt request generation timing	When the timer overflows or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	A read from the TAi register returns a counter value
Write to timer	• A write to the TAi register while the count is stopped: The value is written to both the reload register and the counter. • A write to the TAi register while counting: The value is written to the reload register (It is transferred to the counter at the next reload timing).[1]
Selectable function[2]	• Normal processing operation (Timers A2 and A3) While a high-level (“H”) signal is applied to the TAjOUT pin (j = 2, 3), the timer increments a counter value at the rising edge of the TAjIN pin or decrements a counter value at the falling edge. • Multiply-by-4 processing operation (Timers A3 and A4) The timer increments the counter value in the following timings: -at the rising edge of TAkIN while TAkOUT is “H” (k = 3, 4) -at the falling edge of TAkIN while TAkOUT is “L” -at the rising edge of TAkOUT while TAkIN is “L” -at the falling edge of TAkOUT while TAkIN is “H” The timer decrements the counter in the following timings: -at the rising edge of TAkIN while TAkOUT is “L” -at the falling edge of TAkIN while TAkOUT is “H” -at the rising edge of TAkOUT while TAkIN is “H” -at the falling edge of TAkOUT while TAkIN is “L” • Counter reset by a Z-phase pulse signal input (Timer A3) The counter value is cleared to 0 by a Z-phase pulse signal input

NOTES:

1. Wait for one or more count source cycles to write after the count starts.
2. Any operation can be selected for timer A3. Timer A2 is used only for the normal processing operation. Timer A4 is used only for the multiply-by-4 operation.



Normal processing operation (Timer A2 and timer A3)

While an "H" is applied to the TAjOUT pin (j = 2, 3), the counter increments at the rising edge of the TAjIN pin and decrements at the falling edge.

Set to 0 by an interrupt request acknowledgement or by a program.

Multiply-by-4 processing operation (Timer A3 and timer A4)

The counter increments at the following timings:
-at the rising edge of TAkIN while TAkOUT is “H”
-at the falling edge of TAkIN while TAkOUT is “L”
-at the rising edge of TAkOUT while TAkIN is “L”
-at the falling edge of TAkOUT while TAkIN is “H”

The counter decrements at the following timings:
-at the rising edge of TAkIN while TAkOUT is “L”
-at the falling edge of TAkIN while TAkOUT is “H”
-at the rising edge of TAkOUT while TAkIN is “H”
-at the falling edge of TAkOUT while TAkIN is “L”

Set to 0 by an interrupt request acknowledgement or by a program.

Figure 15.16 Operation in Event Counter Mode When Handling Two-Phase Pulse Signals on Timers A2, A3, and A4

15.1.2.1 Counter Reset by Two-Phase Pulse Signal Processing

The counter value of timer can be set to 0 by a Z-phase pulse signal input (counter reset) when processing two-phase pulse signals.

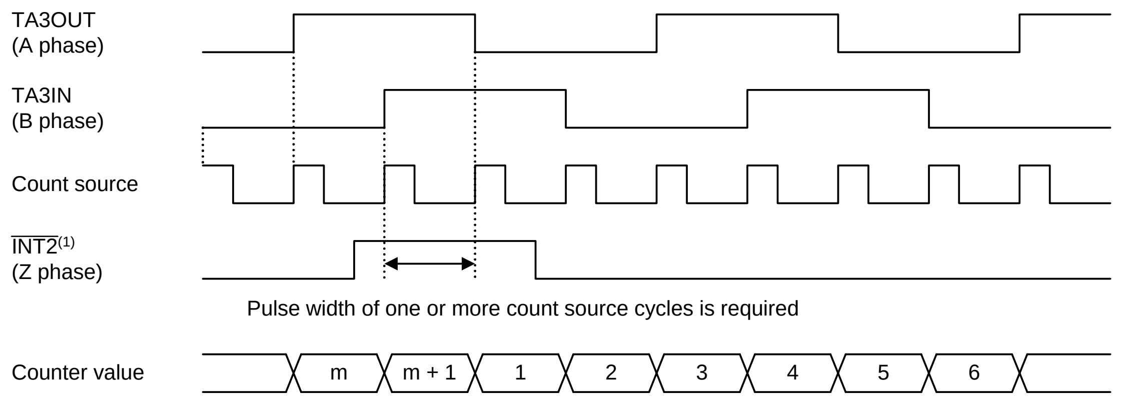
This function can be used when all the following conditions are met; timer A3 event counter mode, two-phase pulse signal processing, free-running count operation type, and multiply-by-4 processing. The Z-phase pulse signal is applied to the $\overline{\text{INT2}}$ pin.

When the TAZIE bit in the ONSF register is set to 1 (Z-phase input enabled), Z-phase pulse input is enabled to reset the counter. To reset the counter by a Z-phase pulse input, set the TA3 register to 0000h beforehand.

A Z-phase pulse input is enabled when the edge of a signal applied to the $\overline{\text{INT2}}$ pin is detected. The POL bit in the INT2IC register can determine the edge polarity. The Z-phase pulse must have a pulse width of one or more timer A3 count source cycles. Figure 15.17 shows relations between two-phase pulses (A-phase and B-phase) and the Z-phase pulse.

Z-phase pulse input resets the counter in the next count source timing followed a Z-phase pulse input.

A timer A3 interrupt request is generated twice in a row if a timer A3 overflow or underflow, and the counter reset by an $\overline{\text{INT2}}$ input occur at the same time. Do not generate a timer A3 interrupt request when this function is used.


TA3OUT
(A phase)
TA3IN
(B phase)
Count source
$\overline{\text{INT2}}$(1)
(Z phase)
Pulse width of one or more count source cycles is required
Counter value
m
m + 1
1
2
3
4
5
6

NOTE:
1. Example when the rising edge of $\overline{\text{INT2}}$ is selected.

Figure 15.17 Relations between Two-Phase Pulses (A-Phase and B-Phase) and Z-Phase Pulse

15.1.3 One-Shot Timer Mode

When a trigger occurs, the counter decrements until underflows. Then, the counter is reloaded and stops until the next trigger occurs.

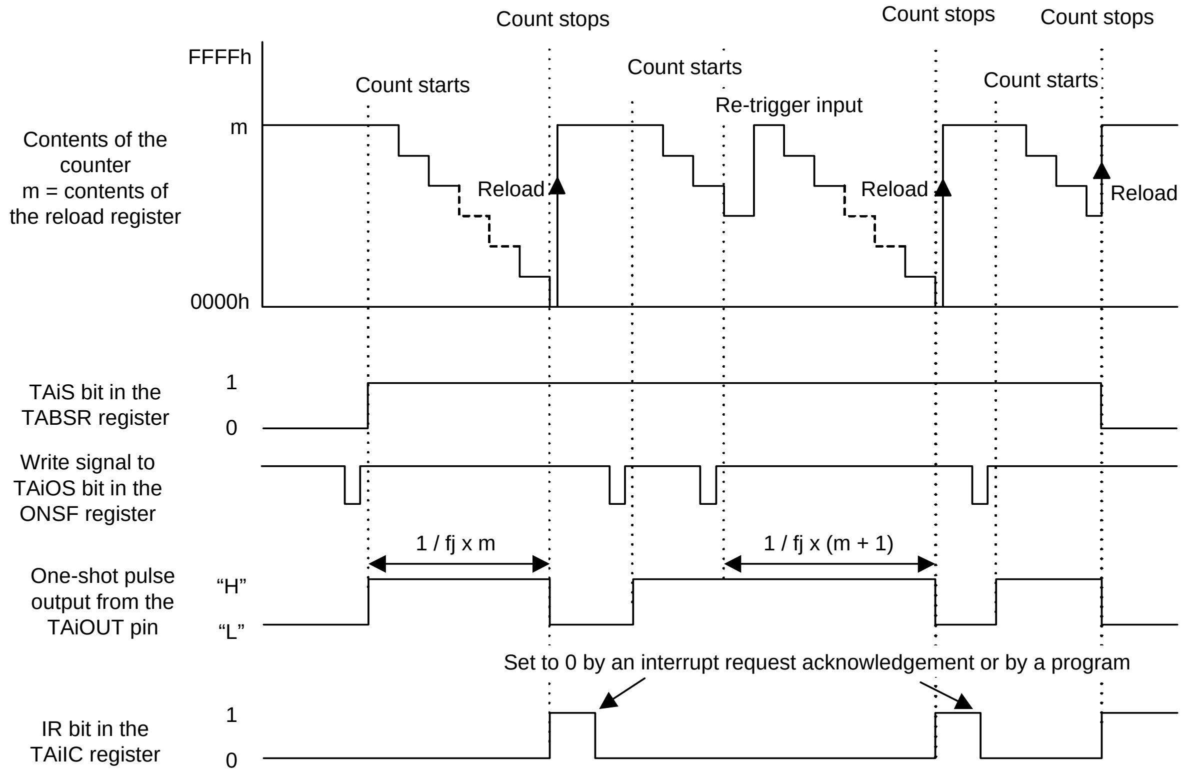
Table 15.6 lists specifications of one-shot timer mode. Figure 15.18 shows a one-shot timer mode operation.

Table 15.6 Specifications of One-Shot Timer Mode

Item	Specification
Count source	f1, f8, f2n[1], fC32
Count operation	• Counter decrements When the counter reaches 0000h, the counter is reloaded and stops until the next trigger occurs. If a trigger occurs while counting, the contents of the reload register are reloaded into the counter and the count continues.
Number of counting	n times n: setting value of the TAi register (i = 0 to 4), 0000h to FFFFh (but the counter does not run if n = 0000h)
Count start condition	A trigger, selectable from the following, occurs while the TAiS bit in the TABSR register is set to 1 (count starts): • the TAiOS bit in the ONSF register is set to 1 (timer starts) • an external trigger is applied to TAiIN pin • timer B2 overflows or underflows, • timer Aj overflows or underflows (j = i - 1, except j = 4 if i = 0), • timer Ak overflows or underflows (k = i + 1, except k = 0 if i = 4)
Count stop condition	• After the counter reaches 0000h and the counter value is reloaded • When the TAiS bit is set to 0 (count stops)
Interrupt request generation timing	When the counter reaches 0000h
TAiIN pin function	Trigger input
TAiOUT pin function	Pulse output
Read from timer	A read from the TAi register returns undefined value
Write to timer	• A write to the TAi register while the count is stopped: The value is written to both the reload register and the counter. • A write to the TAi register while counting: The value is written to the reload register (It is transferred to the counter at the next reload timing).[2]
Selectable function	Pulse output function "L" is output while the count stops. "H" is output while counting.

NOTES:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Wait for one or more count source cycles to write after the count starts.


Count stops
Count stops
Count stops
FFFFh
Count starts
Count starts
Count starts
Re-trigger input
m
Contents of the counter
m = contents of the reload register
Reload
Reload
Reload
0000h
TAiS bit in the TABSR register
1
0
Write signal to TAiOS bit in the ONSF register
1 / fj x m
1 / fj x (m + 1)
One-shot pulse output from the TAiOUT pin
"H"
"L"
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TAiIC register
1
0

fj: Frequency of the count source (f1, f8, f2n(1), fC32)
i = 0 to 4

NOTE:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).

(Conditions) TAiMR register: Bits TMOD1 and TMOD0 are set to 10b (one-shot timer mode).
The MR2 bit is set to 0 (The TAiOS bit is enabled).

Figure 15.18 Operation in One-Shot Timer Mode (Timer A)

15.1.4 Pulse Width Modulation Mode

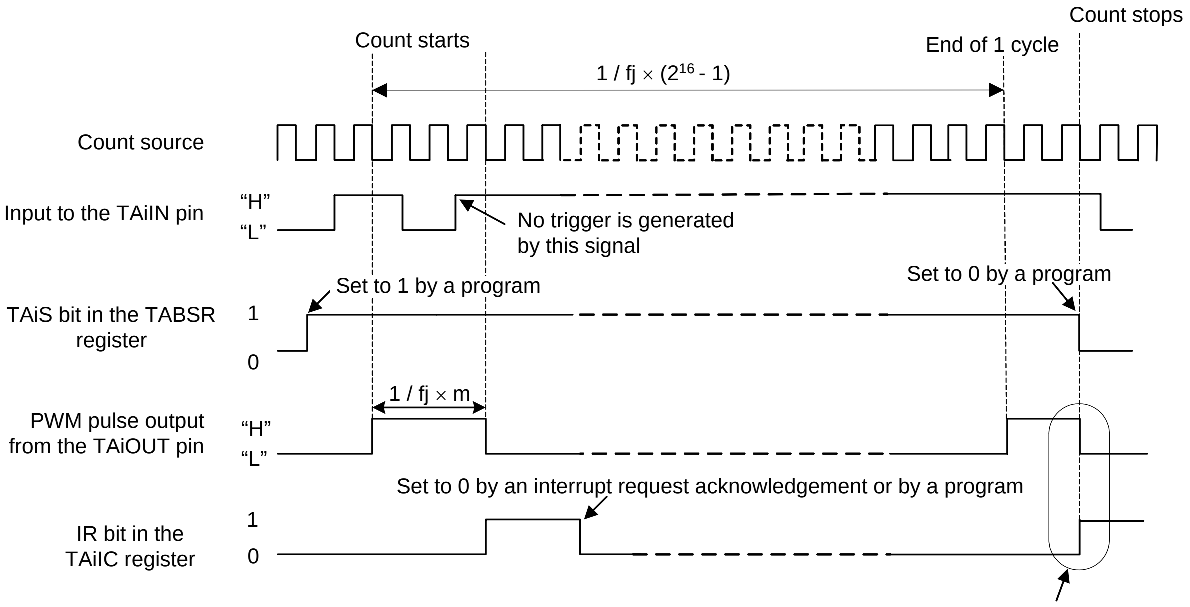
In pulse width modulation mode, the timer outputs pulse signals of a given width repeatedly. The counter functions as an 8-bit pulse width modulator or 16-bit pulse width modulator.

Table 15.7 lists specifications of pulse width modulation mode. Figures 15.19 and 15.20 show examples of a 16-bit pulse width modulator and 8-bit pulse width modulator operations.

Table 15.7 Specifications of Pulse Width Modulation Mode

Item	Specification
Count source	f1, f8, f2n[1], fC32
Count operation	• Counter decrements (The counter functions as the 8-bit or 16-bit pulse width modulator.) The contents of the reload register are reloaded at the rising edge of the PWM pulse and the count continues. The count continues without reloading even if the re-trigger occurs while counting.
16-bit PWM	• “H” width = n / fj n: setting value of the TAi register (i = 0 to 4), 0000h to FFFEh fj: count source frequency • Cycle = $(2^{16} - 1)$ / fj The cycle is fixed to this value
8-bit PWM	• “H” width = n x (m + 1) / fj • Cycle = $(2^{8} - 1)$ x (m + 1) / fj m: setting value of low-order bit address of the TAi register, 00h to FFh n: setting value of high-order bit address of the TAi register, 00h to FEh
Count start condition	When a trigger is not used (the MR2 bit in the TAiMR register is 0): Set the TAiS bit in the TABSR register to 1 When a trigger is used (the MR2 bit in the TAiMR register is 1): A trigger, selectable from the following occurs while the TAiS bit in the TABSR register is set to 1(count starts): • an external trigger is applied to TAiIN pin • timer B2 overflows or underflows • timer Aj overflows or underflows (j = i - 1, except j = 4 if i = 0) • timer Ak overflows or underflows (k = i + 1, except k = 0 if i = 4)
Count stop condition	The TAiS bit is set to 0 (count stops)
Interrupt request generation timing	At the falling edge of the PWM pulse
TAiIN pin function	Trigger input
TAiOUT pin function	Pulse output
Read from timer	A read from the TAi register returns undefined value
Write to timer	• A write to the TAi register while the count is stopped: The value is written to both the reload register and the counter. • A write to the TAi register while counting: The value is written to the reload register (It is transferred to the counter at the next reload timing).[2]

NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Wait for one or more count source cycles to write after the count starts.


Count stops
Count starts
End of 1 cycle
1 / fj × (2^16 - 1)
Count source
Input to the TAiIN pin
"H"
"L"
No trigger is generated by this signal
Set to 1 by a program
Set to 0 by a program
TAiS bit in the TABSR register
1
0
1 / fj × m
PWM pulse output from the TAiOUT pin
"H"
"L"
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TAiIC register
1
0

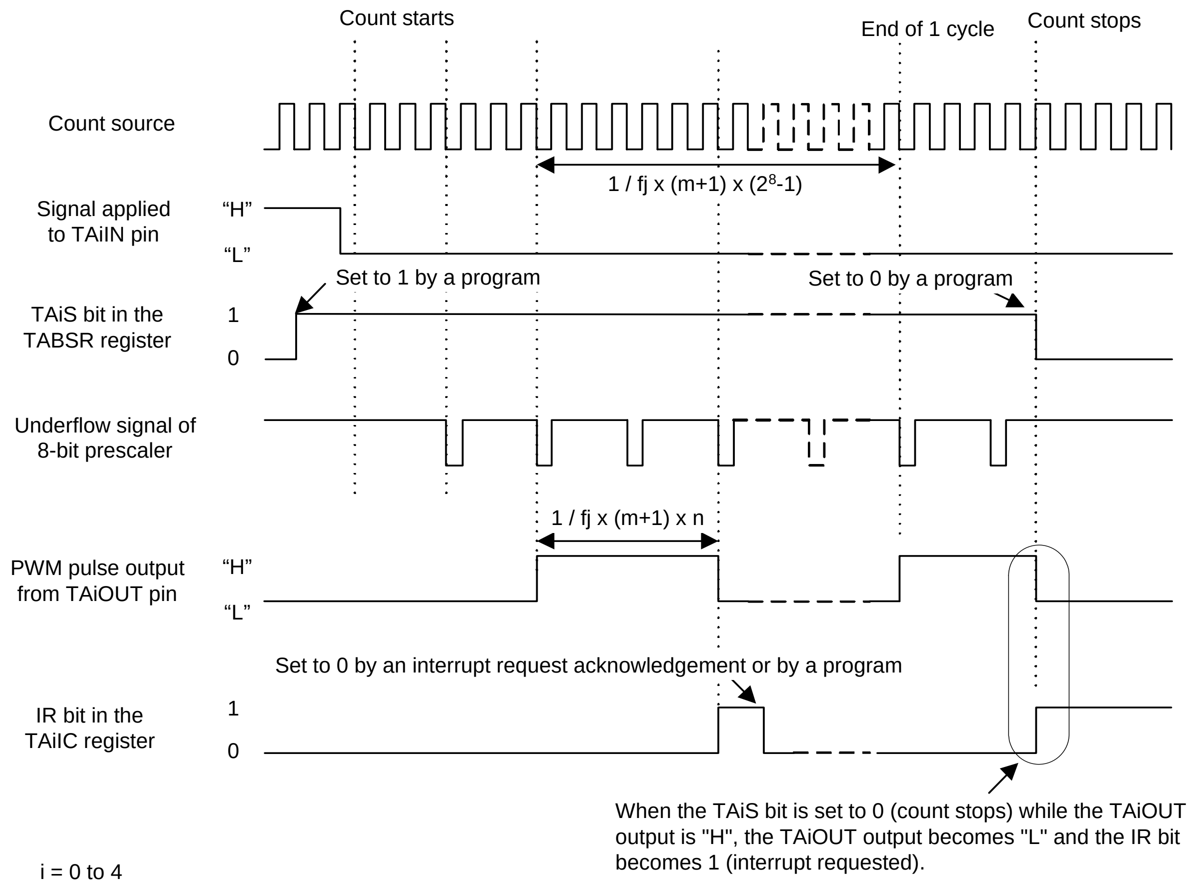
When the TAiS bit is set to 0 (count stops) while the TAiOUT output is "H", the TAiOUT output becomes "L" and the IR bit is set to 1 (interrupt requested).

i = 0 to 4
fj: Count source frequency (f1, f8, f2n[1], fC32)
m: Setting value of the TAi register (0000h to FFFEh)

(Conditions) TAi register is set to 0005h.
TAiMR register: MR1 bit is set to 1 (rising edge of signal applied to the TAiIN pin)

NOTE:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).

Figure 15.19 16-Bit Pulse Width Modulator Operation (Timer A)


Count starts
End of 1 cycle
Count stops
Count source
1 / fj x (m+1) x (2⁸-1)
Signal applied to TAiIN pin
"H"
"L"
Set to 1 by a program
Set to 0 by a program
TAiS bit in the TABSR register
1
0
Underflow signal of 8-bit prescaler
1 / fj x (m+1) x n
PWM pulse output from TAiOUT pin
"H"
"L"
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TAiIC register
1
0
When the TAiS bit is set to 0 (count stops) while the TAiOUT output is "H", the TAiOUT output becomes "L" and the IR bit becomes 1 (interrupt requested).

i = 0 to 4
fj: Count source frequency (f1, f8, f2n[1], fC32)
n: high-order bits in the TAi register (00h to FEh)
m: low-order bits in the TAi register (00h to FFh)

(Conditions) High-order bits in the TAi register are set to 02h.
Low-order bits in the TAi register are set to 02h.
TAiMR register: The MR1 bit is set to 0 (falling edge of signal applied to the TAiIN pin.)

NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. The 8-bit pulse width modulator counts underflow signals of the 8-bit prescaler.

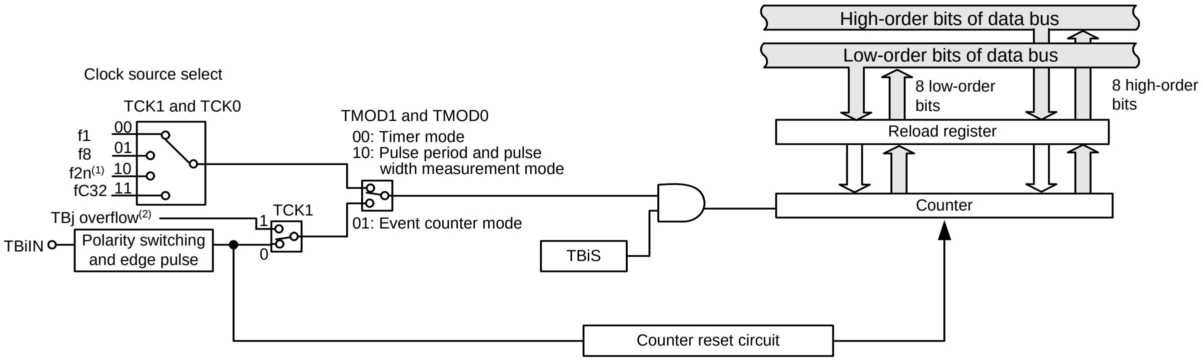
Figure 15.20 8-bit Pulse Width Modulator Operation (Timer A)

15.2 Timer B

Timer B contains the following three modes. Bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 5) determine which mode is used.

- Timer mode: The timer counts the internal count source.
- Event counter mode: The timer counts overflows/underflows of another timer, or the external pulses.
- Pulse period measurement mode, pulse width measurement mode: The timer measures the pulse period or pulse width of the external signal.

Figure 15.21 shows a block diagram of timer B. Figures 15.22 to 15.26 show the registers associated with timer B. Table 15.8 shows TBiIN pin settings (i = 0 to 5).



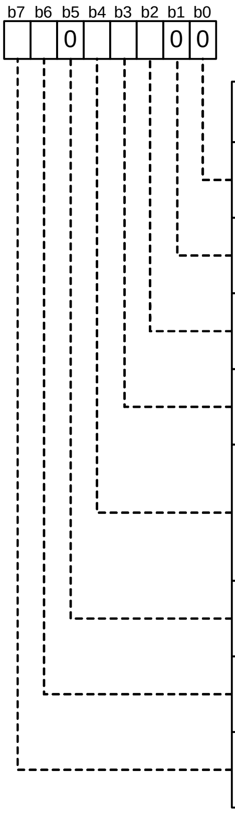
i= 0 to 5
j = i - 1, except j = 2 if i = 0, j = 5 if i = 3.
NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Overflow signal or underflow signal.

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

TBi	Addresses		TBj
Timer B0	0351h	0350h	Timer B2
Timer B1	0353h	0352h	Timer B0
Timer B2	0355h	0354h	Timer B1
Timer B3	0311h	0310h	Timer B5
Timer B4	0313h	0312h	Timer B3
Timer B5	0315h	0314h	Timer B4

Figure 15.21 Timer B Block Diagram

Timer Bi Mode Register (i = 0 to 5)(Timer Mode)



Symbol	Address	After Reset
TB0MR to TB5MR	035Bh, 035Ch, 035Dh, 031Bh, 031Ch, 031Dh	00XX 0000b

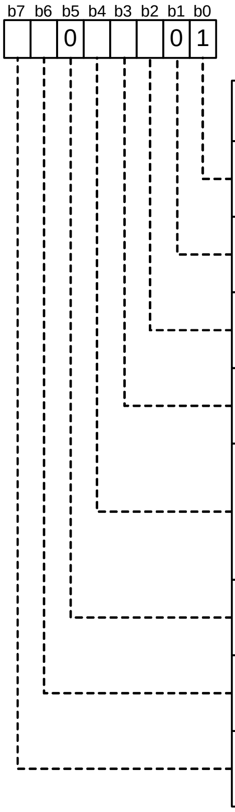
Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 0: Timer mode	RW
TMOD1			RW
MR0	Disabled in timer mode. Can be set to either 0 or 1		RW
MR1			RW
MR2	Registers TB0MR and TB3MR: Set to 0 in timer mode.		RW
	Registers TB1MR, TB2MR, TB4MR, and TB5MR: Unimplemented. Write 0. Read as undefined value.		–
MR3	Disabled in timer mode. Write 0. Read as undefined value.		–
TCK0	Count source select bits	b7 b6 0 0: f1 0 1: f8 1 0: f2n[1] 1 1: fC32	RW
TCK1			RW

NOTE:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15). To select f2n, set the CST bit in the TCSPR register to 1 before setting bits TCK1 and TCK0 to 10b.

Figure 15.22 TB0MR to TB5MR Registers in Timer Mode

Timer Bi Mode Register (i = 0 to 5)(Event Counter Mode)



Symbol	Address	After Reset
TB0MR to TB5MR	035Bh, 035Ch, 035Dh, 031Bh, 031Ch, 031Dh	00XX 0000b

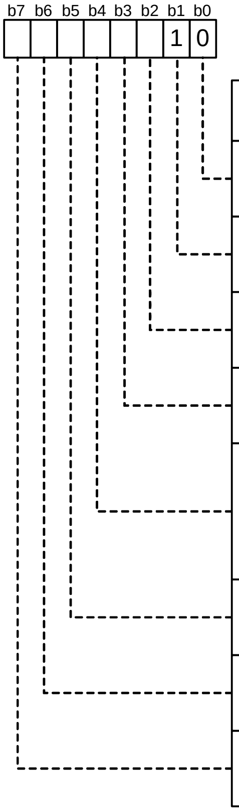
Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 0 1: Event counter mode	RW
TMOD1			RW
MR0	Count polarity select bits[1]	b3 b2 0 0: Falling edges of an external signal counted 0 1: Rising edges of an external signal counted 1 0: Falling and rising edges of an external signal counted 1 1: Do not set to this value	RW
MR1			RW
MR2	Registers TB0MR and TB3MR: Set to 0 in event counter mode.		RW
	Registers TB1MR, TB2MR, TB4MR, and TB5MR: Unimplemented. Write 0. Read as undefined value.		–
MR3	Disabled in event counter mode. Write 0. Read as undefined value.		–
TCK0	Disabled in event counter mode. Can be set to either 0 or 1		RW
TCK1	Event clock select bit	0: Signal applied to the TBiIN pin 1: TBj overflows or underflows[2]	RW

NOTES:

1. Bits MR1 and MR0 are enabled when the TCK1 bit is set to 0. Bits MR1 and MR0 can be set to either 0 or 1 when the TCK1 bit is set to 1.
2. j = i - 1, except j = 2 if i = 0 and j = 5 if i = 3.

Figure 15.23 TB0MR to TB5MR Registers in Event Counter Mode

Timer Bi Mode Register (i = 0 to 5)
(Pulse Period Measurement Mode, Pulse Width Measurement Mode)



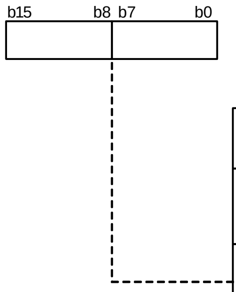
Symbol	Address	After Reset
TB0MR to TB5MR	035Bh, 035Ch, 035Dh, 031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	b1 b0 1 0: Pulse period measurement mode Pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement mode select bits[1]	b3 b2 0 0: Pulse period measurement 1 0 1: Pulse period measurement 2 1 0: Pulse width measurement 1 1: Do not set to this value	RW
MR1			RW
MR2	Registers TB0MR and TB3MR: Set to 0 in pulse period measurement mode, pulse width measurement mode.		RW
	Registers TB1MR, TB2MR, TB4MR, and TB5MR: Unimplemented. Write 0. Read as undefined value.		–
MR3	Timer Bi overflow flag[2]	0: No overflow has occurred 1: Overflow has occurred[3]	RO
TCK0	Count source select bits	b7 b6 0 0: f1 0 1: f8 1 0: f2n[4] 1 1: fC32	RW
TCK1			RW

NOTES:
1. Bits MR1 and MR0 determine the following measurement modes:
   Pulse period measurement 1 (bits MR1 and MR0 are set to 00b):
   Measures the width between the falling edges of a pulse
   Pulse period measurement 2 (bits MR1 and MR0 bits are set to 01b):
   Measures the width between the rising edges of a pulse
   Pulse width measurement (bits MR1 and MR0 bits are set to 10b):
   Measures the width between a falling edge and a rising edge of a pulse, and between a rising edge and a falling edge of a pulse
2. The MR3 bit is undefined when reset.
3. To set the MR3 bit to 0 (no overflow), wait for one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1 (overflow), while the TBiS bit in TABSR or TBSR register is set to 1 (count starts).
4. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15). To select f2n, set the CST bit in the TCSPR register to 1 before setting bits TCK1 and TCK0 to 10b.

Figure 15.24 TB0MR to TB5MR Registers in Pulse Period Measurement Mode, Pulse Width Measurement Mode

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



Symbol	Address	After Reset
TB0 to TB2	0351h - 0350h, 0353h - 0352h, 0355h - 0354h	Undefined
TB3 to TB5	0311h - 0310h, 0313h - 0312h, 0315h - 0314h	Undefined

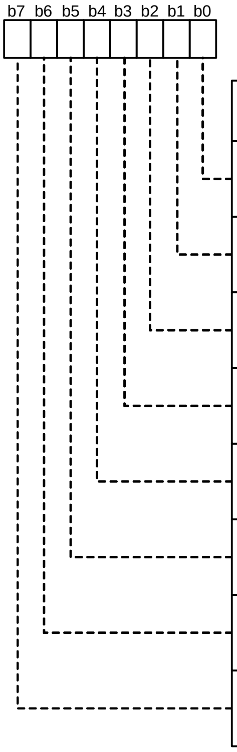
Mode	Function	Setting Range	RW
Timer Mode	If a count source frequency is fj, and the setting value of the TBi register is n, the counter cycle is (n+1) / fj.	0000h to FFFFh	RW
Event Counter Mode	If the setting value of the TBi register is n, the count times are (n+1)[2]	0000h to FFFFh	RW
Pulse Period Measurement Mode, Pulse Width Measurement Mode	Increment the counter between one valid edge and another valid edge of a pulse applied to the TBiIN pin	–	RO

NOTES:

1. Read and write this register in 16-bit units.
2. Timer Bi counts overflows/underflows of another timer, or the external pulses.

Figure 15.25 TB0 to TB5 Registers

Count Start Register



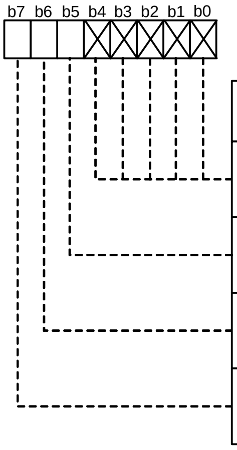
Symbol TABSR

Address 0340h

After Reset 00h

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

Timer B3, B4, B5 Count Start Register



Symbol TBSR

Address 0300h

After Reset 000X XXXXb

Bit Symbol	Bit Name	Function	RW
– (b4-b0)	Unimplemented. Write 0. Read as undefined value.		–
TB3S	Timer B3 count start bit	0: Count stops 1: Count starts	RW
TB4S	Timer B4 count start bit	0: Count stops 1: Count starts	RW
TB5S	Timer B5 count start bit	0: Count stops 1: Count starts	RW

Figure 15.26 TABSR Register, TBSR Register

Table 15.8 TBiIN Pin Settings (i = 0 to 5)

Port	Function	Bit Setting	
		PD7, PD9(1) Registers	PS1, PS3(1) Registers
P7_1	$\overline{\text{TB5IN}}$	PD7_1 = 0	PS1_1 = 0
P9_0	$\overline{\text{TB0IN}}$	PD9_0 = 0	PS3_0 = 0
P9_1	$\overline{\text{TB1IN}}$	PD9_1 = 0	PS3_1 = 0
P9_2	$\overline{\text{TB2IN}}$	PD9_2 = 0	PS3_2 = 0
P9_3	$\overline{\text{TB3IN}}$	PD9_3 = 0	PS3_3 = 0
P9_4	$\overline{\text{TB4IN}}$	PD9_4 = 0	PS3_4 = 0

NOTE:

1. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.

15.2.1 Timer Mode

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

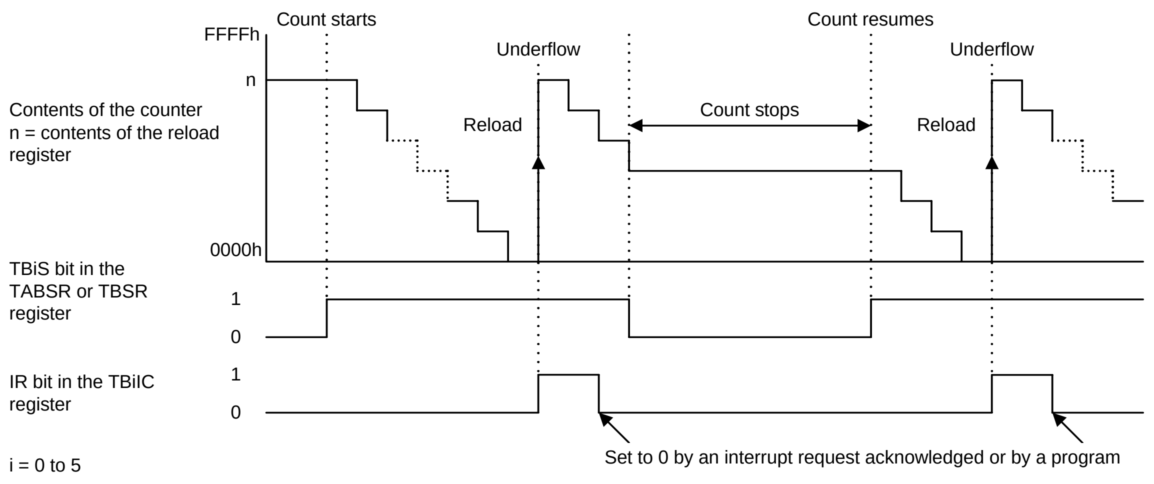
Table 15.9 lists specifications of timer mode. Figure 15.27 shows a timer mode operation (Timer B).

Table 15.9 Specifications of Timer Mode

Item	Specification
Count source	f1, f8, f2n[1], fC32
Count operation	• Counter decrements When the timer underflows, the contents of the reload register are reloaded into the counter and the count continues.
Counter cycle	$\frac{n + 1}{fj}$ fj: count source frequency n: setting value of the TBi register (i=0 to 5), 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is set to 1 (count starts)
Count stop condition	The TBiS bit is set to 0 (count stops)
Interrupt request generation timing	When the timer underflows
TBiIN pin function	Programmable I/O port
Read from timer	A read from the TBi register returns a counter value.
Write to timer	• A write to the TBi register while the count is stopped: The value is written to both the reload register and the counter. • A write to the TBi register while counting: The value is written to the reload register (It is transferred to the counter at the next reload timing).[2]

NOTES:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Wait for one or more count source cycles to write after the count starts.


Count starts
FFFFh
Underflow
Count resumes
Underflow
n
Contents of the counter
n = contents of the reload register
Reload
Count stops
Reload
0000h
TBiS bit in the TABSR or TBSR register
1
0
IR bit in the TBiIC register
1
0
Set to 0 by an interrupt request acknowledged or by a program
i = 0 to 5

(Condition) TBiMR register: Bits TMOD1 and TMOD0 are set to 00b (timer mode).

Figure 15.27 Operation in Timer Mode (Timer B)

15.2.2 Event Counter Mode

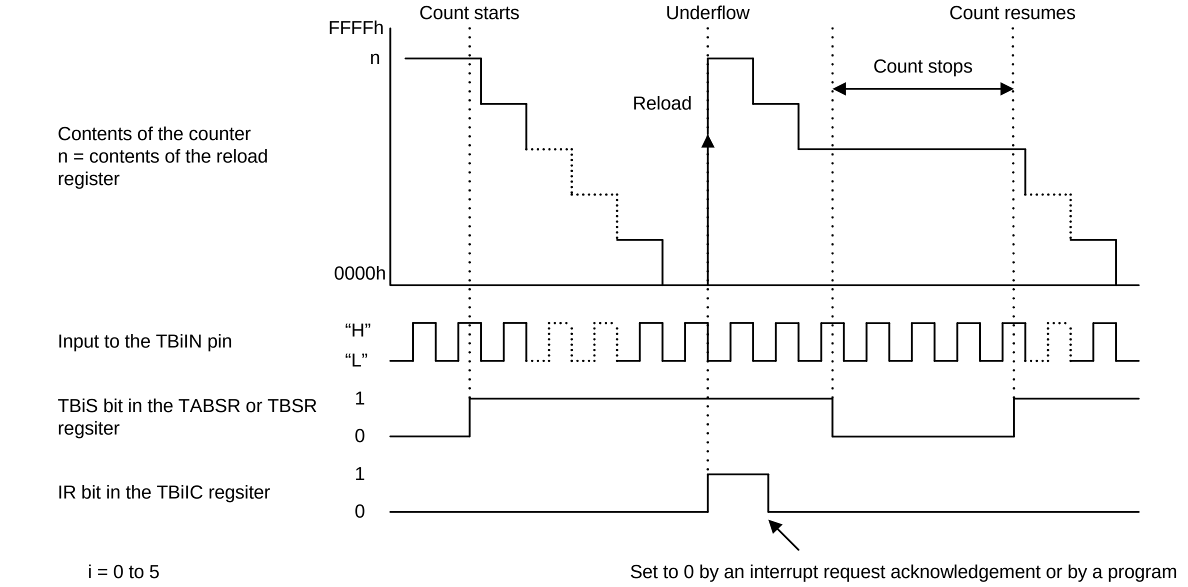
In event counter mode, the timer counts overflows/underflows of another timer, or the external pulses. Table 15.10 lists specifications of event counter mode. Figure 15.28 shows an event counter mode operation.

Table 15.10 Specifications of Event Counter Mode

Item	Specification
Count source	• External signal applied to the TBiIN pin (i = 0 to 5) (valid edge can be selected by a program) • TBj overflows or underflows (j = i - 1, except j = 2 if i = 0, j = 5 if i = 3)
Count operation	• Counter decrements When the timer underflows, the contents of the reload register are reloaded into the counter and the count continues.
Number of counting	(n + 1) times n: Setting value of the TBi register 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is set to 1 (count starts)
Count stop condition	The TBiS bit is set to 0 (count stops)
Interrupt request generation timing	When the timer underflows
TBiIN pin function	Count source input
Read from timer	A read from the TBi register returns a counter value.
Write to timer	• A write to the TBi register while the count is stopped: The value is written to both the reload register and the counter. • A write to the TBi register while counting: The value is written to the reload register (It is transferred to the counter at the next reload timing).(1)

NOTE:

1. Wait for one or more count source cycles to write after the count starts.


Count starts
Underflow
Count resumes
FFFFh
n
Count stops
Reload
Contents of the counter
n = contents of the reload register
0000h
Input to the TBiIN pin
“H”
“L”
TBiS bit in the TABSR or TBSR regsiter
1
0
IR bit in the TBiIC regsiter
1
0
i = 0 to 5
Set to 0 by an interrupt request acknowledgement or by a program

(Condition) TBiMR register: Bits TMOD1 and TMOD0 are set to 01b (event counter mode)
Bits MR1 and MR0 are set to 00b (count the falling edge of the external signal)
The TCK1 bit is set to 0 (signal input to TBiIN pin)

Figure 15.28 Operation in Event Counter Mode (Timer B)

15.2.3 Pulse Period Measurement Mode, Pulse Width Measurement Mode

In pulse period measurement mode and pulse width measurement mode, the timer measures pulse period or pulse width of the external signal.

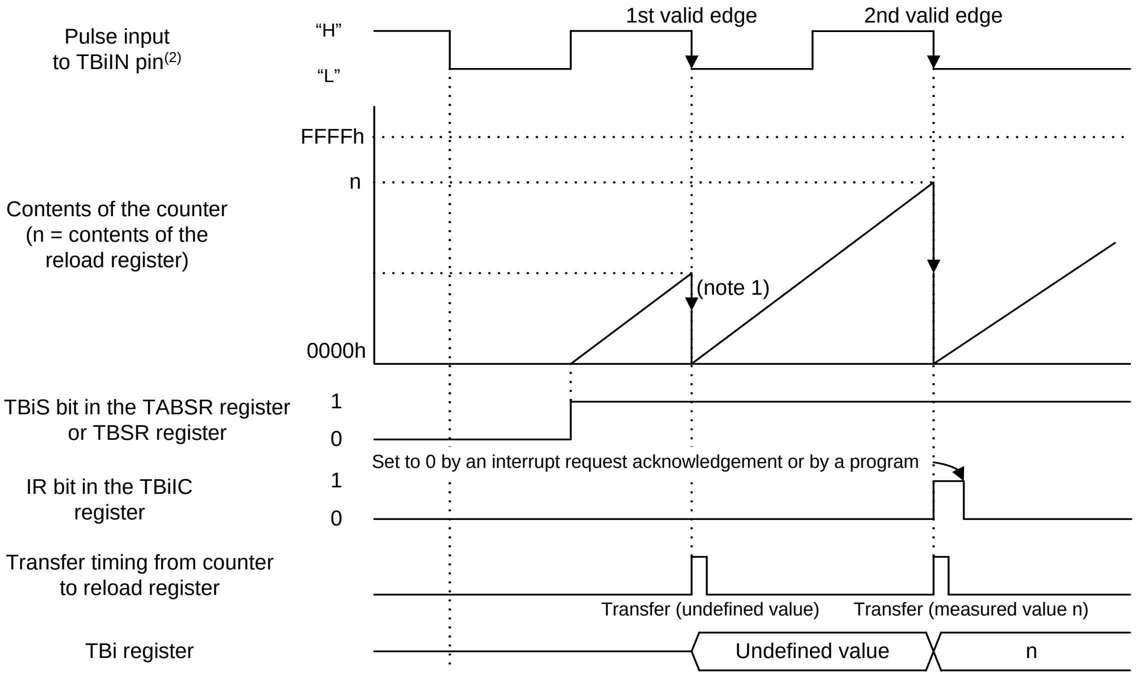
Table 15.11 shows specifications in pulse period measurement mode and pulse width measurement mode. Figure 15.29 shows a pulse period measurement operation. Figure 15.30 shows a pulse width measurement operation.

Table 15.11 Specifications of Pulse Period Measurement Mode, Pulse Width Measurement Mode

Item	Specification
Count source	f1, f8, f2n(1), fC32
Count operation	• Counter increments The counter value is transferred to the reload register when the valid edge of a pulse is detected. Then the counter becomes 0000h and the count continues.
Count start condition	The TBiS bit (i = 0 to 5) in the TABSR or TBSR register is set to 1 (count starts)
Count stop condition	The TBiS bit is set to 0 (count stops)
Interrupt request generation timing	• When the valid edge of a pulse is input(2) • When the timer overflows(3) The MR3 bit in the TBiMR register is set to 1 (overflow) simultaneously.
TBiIN pin function	Pulse input
Read from timer	A read from the TBi register returns the contents of the reload register (measurement results)(4)
Write to timer	The TBi register cannot be written

NOTES:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. An interrupt request is not generated when the first valid edge is input after the count starts.
3. To set the MR3 bit to 0 (no overflow), wait for one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1, while the TBiS bit is set to 1.
4. A value read from the TBi register is undefined until the second valid edge is detected after the count starts.

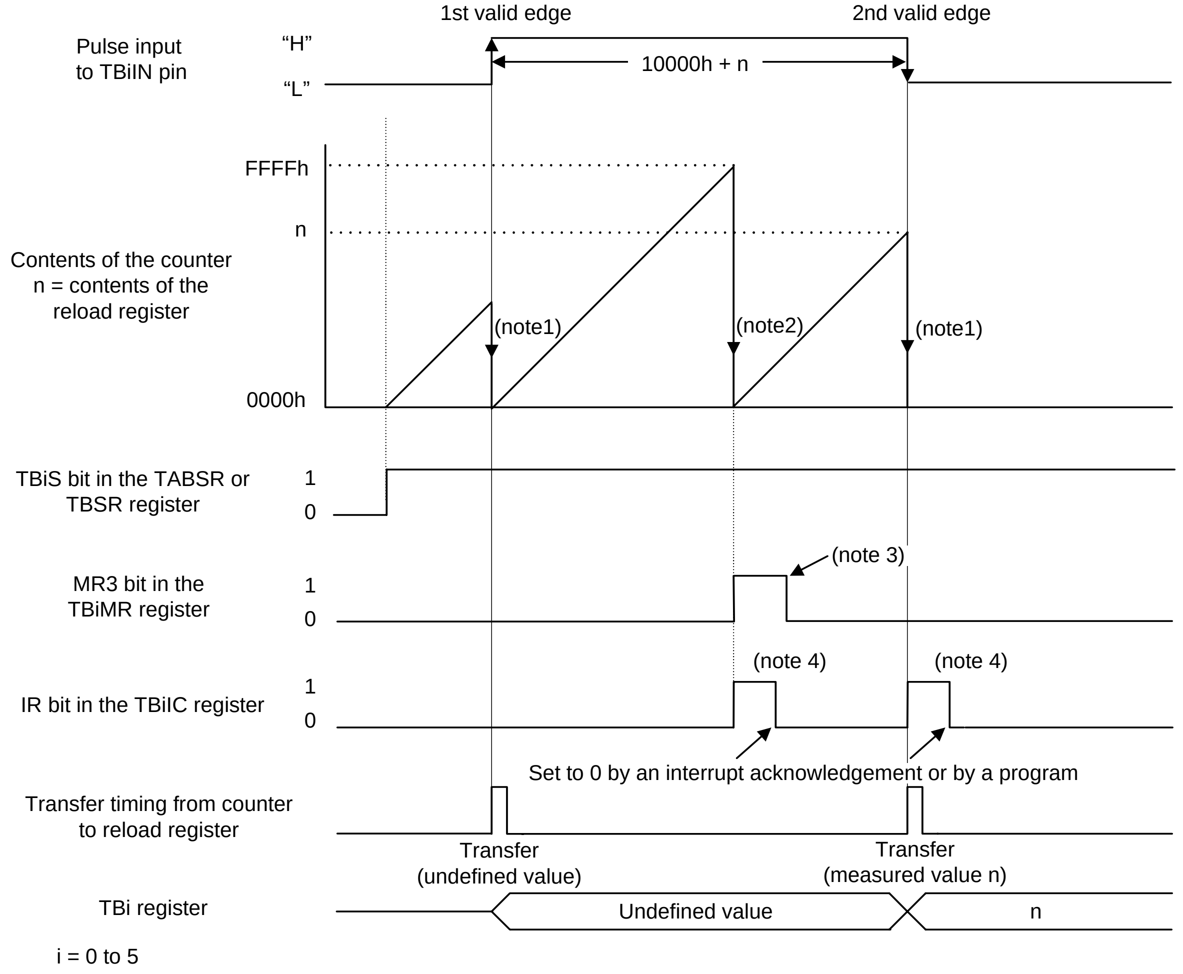


i = 0 to 5

NOTES:

1. Counter is reset due to the completion of the measurement.
2. If an overflow and a valid edge input occur simultaneously, an interrupt request is generated only once, which results in the valid edge not being recognized. Do not let an overflow occur.

Figure 15.29 Operation in Pulse Period Measurement Mode (Timer B)



i = 0 to 5

NOTES:

1. Counter is reset due to the completion of the measurement.
2. Overflow
3. To set the MR3 bit to 0 (no overflow), wait for one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1 (overflow), while the TBiS bit in TABSR or TBSR register is set to 1 (count starts).
4. Determine whether an interrupt source is a valid edge input or an overflow by reading the port level in the TBi interrupt routine.

Figure 15.30 Operation in Pulse Width Measurement Mode (Timer B)

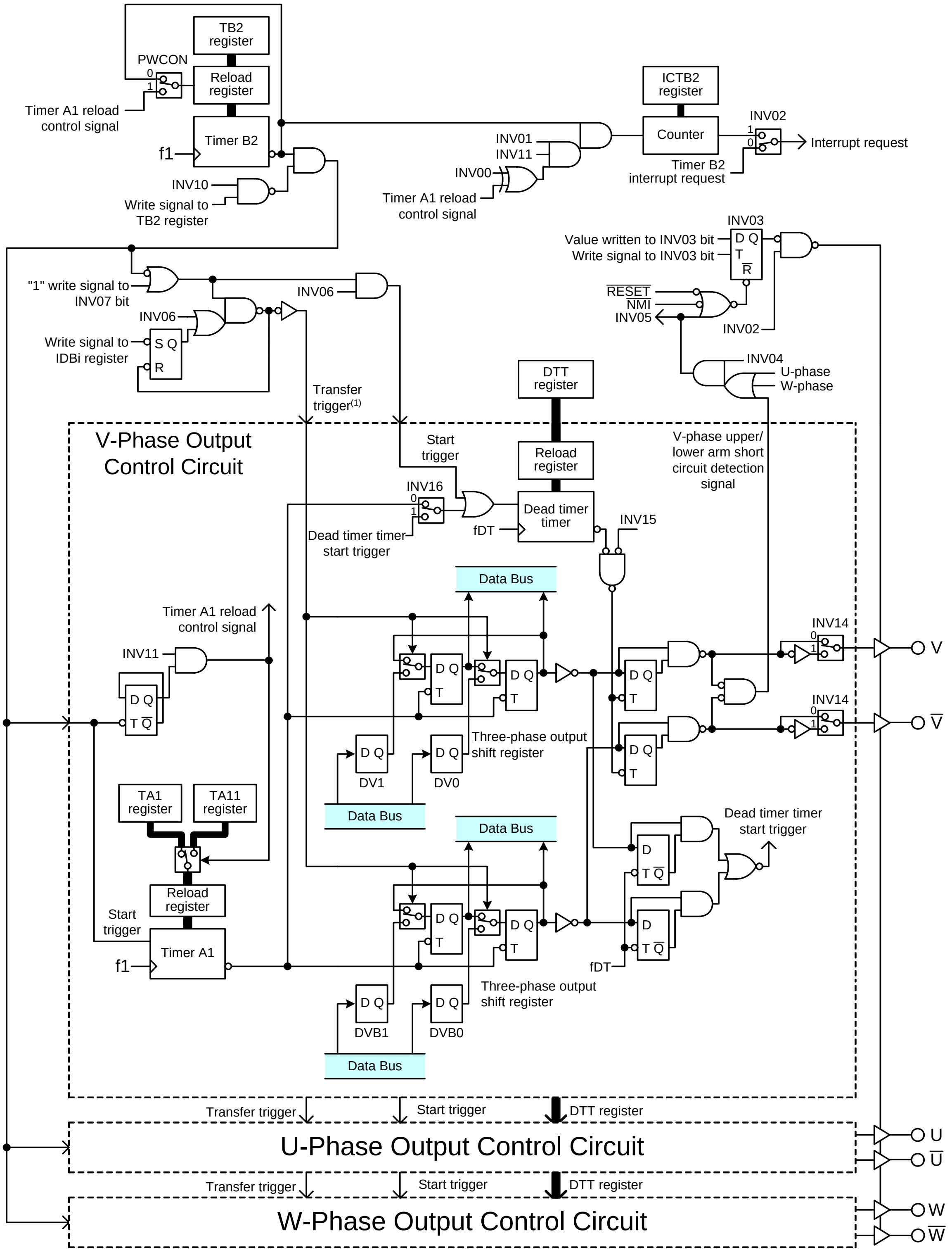
16. Three-Phase Motor Control Timer Function

The PWM waveform can be output by using timers B2, A1, A2, and A4. Timer B2 is used for the carrier wave control, and timers A4, A1, and A2 for the U-, V-, and W-phase PWM control.

Table 16.1 lists specifications of the three-phase motor control timer functions. Table 16.2 lists pin settings. Figure 16.1 shows a block diagram. Figures 16.2 to 16.10 show registers associated with the three-phase motor control timer function.

Table 16.1 Specifications of Three-Phase Motor Control Timers

Item	Specification
Control method	Three-phase full wave method
Modulation modes	• Triangular wave modulation mode • Sawtooth wave modulation mode
Active level	Selectable either active High or active Low
Timers to be used	• Timer B2 (Carrier wave cycle control: used in timer mode) • Timers A4, A1, and A2 (U-, V-, W-phase PWM control: used in one-shot timer mode):
Short circuit prevention features	• Prevention function against upper and lower arm short circuit caused by program errors • Arm short circuit prevention function using dead time timer • Forced cutoff function by $\overline{\text{NMI}}$ input


TB2 register
PWCON
0
1
Reload register
Timer A1 reload control signal
f1
Timer B2
INV10
Write signal to TB2 register
ICTB2 register
Counter
INV02
1
0
Interrupt request
Timer B2 interrupt request
INV01
INV11
INV00
Timer A1 reload control signal
INV03
Value written to INV03 bit
Write signal to INV03 bit
D Q
T
R
RESET
NMI
INV05
INV02
"1" write signal to INV07 bit
INV06
INV06
Write signal to IDBi register
S Q
R
Transfer trigger(1)
INV04
U-phase
W-phase
DTT register
V-Phase Output Control Circuit
Start trigger
Reload register
V-phase upper/ lower arm short circuit detection signal
INV16
Dead timer timer
fDT
Dead timer timer start trigger
INV15
Data Bus
Timer A1 reload control signal
INV11
INV14
V
INV14
V̄
Three-phase output shift register
DV1
DV0
Data Bus
TA1 register
TA11 register
Reload register
Start trigger
Timer A1
f1
Data Bus
Dead timer timer start trigger
fDT
Three-phase output shift register
DVB1
DVB0
Data Bus
Transfer trigger
Start trigger
DTT register
U-Phase Output Control Circuit
U
Ū
Transfer trigger
Start trigger
DTT register
W-Phase Output Control Circuit
W
W̄

INV00 to INV07: bits in the INVC0 register
INV10 to INV15: bits in the INVC1 register
DVi, DVBi: bits in the IDBi register (i = 0, 1)
PWCON: bit in the TB2SC register

NOTE:

1. When the INV06 bit is set to 0 (triangular wave modulation mode), a transfer trigger is generated at the first timer B2 underflow after writing to the IDBi register (i = 0, 1).

Figure 16.1 Three-Phase Motor Control Timer Function Block Diagram

Three-Phase PWM Control Register 0[1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
INVC0	0308h	00h

Bit Symbol	Bit Name	Function	RW
INV00	ICTB2 count condition select bits	b1 b0 0 0: } Timer B2 underflow 0 1: } 1 0: Timer B2 underflow at the rising edge of the timer A1 reload control signal[2] (every odd-numbered timer B2 underflow) 1 1: Timer B2 underflow at the falling edge of the timer A1 reload control signal[2] (every even-numbered timer B2 underflow)	RW
INV01			RW
INV02	Three-phase motor control timer function enable bit[3]	0: Three-phase motor control timer function not used 1: Three-phase motor control timer function used[4,5]	RW
INV03	Three-phase motor control timer output control bit	0: Three-phase motor control timer output disabled[5,6] 1: Three-phase motor control timer output enabled	RW
INV04	Upper and lower arm simultaneous turn-on disable bit	0: Simultaneous turn-on enabled 1: Simultaneous turn-on disabled	RW
INV05	Upper and lower arm simultaneous turn-on detect flag	0: Not detected 1: Detected [7]	RO
INV06	Modulation mode select bit	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode	RW
INV07	Software trigger select bit	Transfer trigger is generated when the INV07 bit is set to 1. Trigger for the dead time timer is also generated when the INV06 bit is set to 1. This bit is read as 0.	RW

NOTES:

1. Set the INVC0 register after the PRC1 bit in the PRCR register is set to 1 (write enable). Set bits INV06 and INV02 to INV00 while timers A1,A2, A4, and B2 are stopped.
2. Set the INV01 bit to 1 after setting a value to the ICTB2 register. Also, when the INV01 bit is set to 1, set the timer A1 count start bit to 1 prior to the first timer B2 underflow.
3. Set pins after the INV02 bit is set to 1. Refer to the table, **Pin settings when using three-phase motor control timer function**.
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 0 and the INV02 bit to 1, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ (including when other output functions are assiged to these pins) are all placed in high-impedance states.
6. The INV03 bit becomes 0 when one of the following occurs:
   - -Reset
   - -The both upper and lower arms output the active level signals at the same time while the INV04 bit is set to 1
   - -The INV03 bit is set to 0 by a program
   - -Signal applied to the $\overline{NMI}$ pin changes from "H" to "L" (while an "L" is applied to the $\overline{NMI}$ pin, the INV03 bit cannot be set to 1).
7. The INV05 bit cannot be set to 1 by a program. To set the INV05 bit to 0, write a 0 to the INV04 bit.

Figure 16.2 INVC0 Register

Three-Phase PWM Control Register 1[1]

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	After Reset
INVC1	0309h	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 a write to the TB2 register	RW
INV11	Timers A11, A21, and A41 control bit	0: Timers A11, A21, and A41 not used (Three-phase mode 0) 1: Timers A11, A21, and A41 used (Three-phase mode 1)	RW
INV12	Dead time timer count source (fDT) select bit	0: f1 1: f1 divided-by-2	RW
INV13	Carrier wave rise/fall detect flag[2]	0: Timer B2 underflow occurred an even number of times 1: Timer B2 underflow occurred an odd number of times	RO
INV14	Active level control bit	0: Active Low 1: Active High	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 one-shot pulse of timer (A4, A1, and A2[3]) 1: Rising edge of the three-phase output shift register (U-, V-, W-phase)	RW
– (b7)	Reserved bit	Set to 0	RW

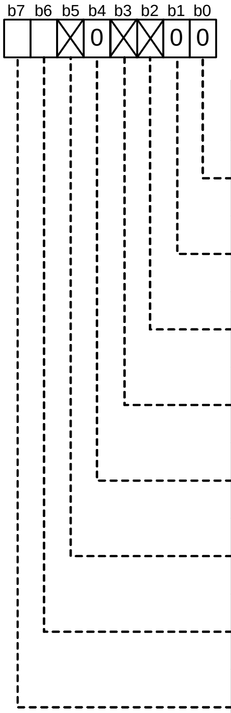
NOTES:

1. Set the INVC1 register after the PRC1 bit in the PRCR register is set to 1 (write enable). Set the INVC1 register while timers A1, A2, A4, and B2 are stopped.
2. The INV13 bit is enabled only when the INV06 bit is set to 0 (triangular wave modulation mode) and the INV11 bit to 1.
3. If the following conditions are all met, set the INV16 bit to 1.
   - The INV15 bit is set to 0
   - Bits Dij (i = U, V or W, j = 0, 1) and DiBj in the IDBj register always have different values when the INV03 bit in the INVC0 register is set to 1 (three-phase control timer output enabled).
     (The upper arm and lower arm always output opposite level signals at any time except dead time.)

   If any of the above conditions is not met, set the INV16 bit to 0.

Figure 16.3 INVC1 Register

Timer B2 Mode Register



Symbol	Address	After Reset
TB2MR	035Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	Set to 00b (timer mode) to use the three-phase motor control timer function	RW
TMOD1			RW
MR0	Disabled to use the three-phase motor control timer function. Can be set to either 0 or 1.		RW
MR1			RW
MR2	Set to 0 to use the three-phase motor control timer function		RW
MR3	Unimplemented. Write 0. Read as undefined value.		–
TCK0	Count source select bits	Set to 00b (f1) to use the three-phase motor control timer function	RW
TCK1			RW

Figure 16.4 TB2MR Register when Using Three-Phase Motor Control Timer Function

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

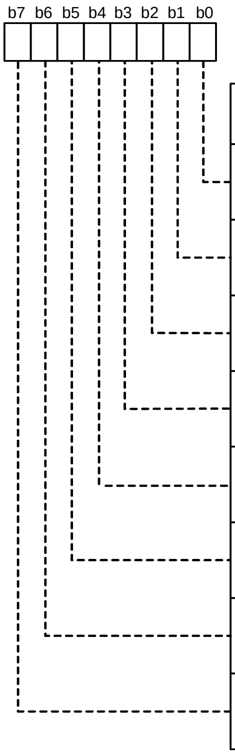
b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	1	0

Symbol	Address	After Reset
TA1MR, TA2MR, TA4MR	0357h, 0358h, 035Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating mode select bits	Set to 10b (one-shot timer mode) to use the three-phase motor control timer function	RW
TMOD1			RW
– (b2)	Reserved bit	Set to 0	RW
MR1	External trigger select bit	Set to 0 to use the three-phase motor control timer function	RW
MR2	Trigger select bit	Set to 1 (selected by the TRGSR register) to use the three-phase motor control timer function	RW
MR3	Set to 0 to use the three-phase motor control timer function		RW
TCK0	Count source select bits	Set to 00b (f1) to use the three-phase motor control timer function	RW
TCK1			RW

Figure 16.5 TA1MR, TA2MR, and TM4MR Registers when Using Three-Phase Motor Control Timer Function

Trigger Select Register

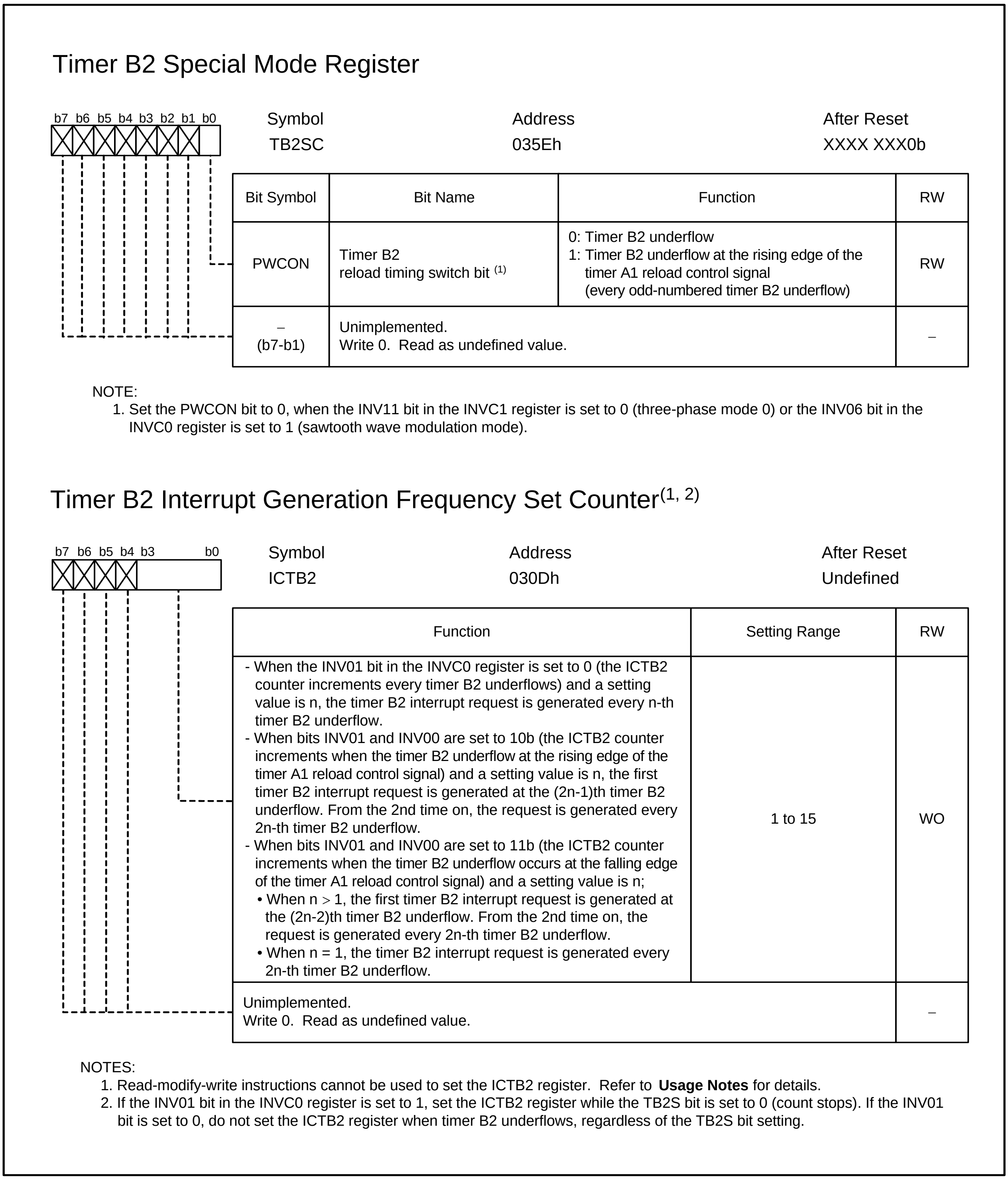


Symbol	Address	After Reset
TRGSR	0343h	00h

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

NOTE:
1. Overflow or underflow.

Figure 16.6 TRGSR Register when Using Three-Phase Motor Control Timer Function



Timer B2 Special Mode Register

Symbol	Address	After Reset
TB2SC	035Eh	XXXX XXX0b

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 reload timing switch bit [(1)]	0: Timer B2 underflow 1: Timer B2 underflow at the rising edge of the timer A1 reload control signal (every odd-numbered timer B2 underflow)	RW
– (b7-b1)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. Set the PWCON bit to 0, when the INV11 bit in the INVC1 register is set to 0 (three-phase mode 0) or the INV06 bit in the INVC0 register is set to 1 (sawtooth wave modulation mode).

Timer B2 Interrupt Generation Frequency Set Counter[(1, 2)]

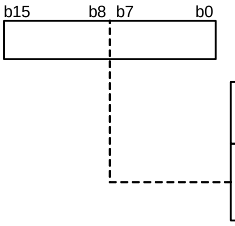
Symbol	Address	After Reset
ICTB2	030Dh	Undefined

Function	Setting Range	RW
- When the INV01 bit in the INVC0 register is set to 0 (the ICTB2 counter increments every timer B2 underflows) and a setting value is n, the timer B2 interrupt request is generated every n-th timer B2 underflow. - When bits INV01 and INV00 are set to 10b (the ICTB2 counter increments when the timer B2 underflow at the rising edge of the timer A1 reload control signal) and a setting value is n, the first timer B2 interrupt request is generated at the (2n-1)th timer B2 underflow. From the 2nd time on, the request is generated every 2n-th timer B2 underflow. - When bits INV01 and INV00 are set to 11b (the ICTB2 counter increments when the timer B2 underflow occurs at the falling edge of the timer A1 reload control signal) and a setting value is n; • When n > 1, the first timer B2 interrupt request is generated at the (2n-2)th timer B2 underflow. From the 2nd time on, the request is generated every 2n-th timer B2 underflow. • When n = 1, the timer B2 interrupt request is generated every 2n-th timer B2 underflow.	1 to 15	WO
Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Read-modify-write instructions cannot be used to set the ICTB2 register. Refer to **Usage Notes** for details.
2. If the INV01 bit in the INVC0 register is set to 1, set the ICTB2 register while the TB2S bit is set to 0 (count stops). If the INV01 bit is set to 0, do not set the ICTB2 register when timer B2 underflows, regardless of the TB2S bit setting.

Figure 16.7 TB2SC Register, ICTB2 Register

Timer B2 Register(1)

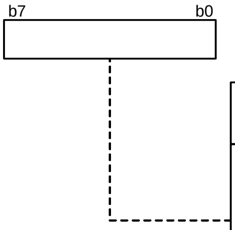


Symbol	Address	After Reset
TB2	0355h - 0354h	Undefined

Function	Setting Range	RW
If a setting value is n, f1 is divided by n+1. Timers A1, A2, and A4 start every time timer B2 underflows.	0000h to FFFFh	RW

NOTE:
1. Read and write this register in 16-bit units.

Dead Time Timer(1, 2, 3)



Symbol	Address	After Reset
DTT	030Ch	Undefined

Function	Setting Range	RW
This one-shot timer is used to delay the timing for a turn-on signal to be switched to its active level in order to prevent the upper and lower arm short circuit. If a setting value is n, the count source is counted n times after the start trigger occurs, and then the timer stops.	01h to FFh	WO

NOTES:
1. Read-modify-write instructions cannot be used to set the DTT register. Refer to **Usage Notes** for details.
2. The DTT register setting is enabled when the INV15 bit in the INVC1 register is set to 0 (dead time enabled). No dead time is generated when the INV15 bit is set to 1 (dead time disabled).
3. The INV16 bit in the INVC1 register determines the start trigger of the DTT register. The INV12 bit determines the count source.

Figure 16.8 TB2 Register, DTT Register when Using Three-Phase Motor Control Timer Function

Timer Ai, Ai1 Register[1, 2, 3, 4, 5] (i = 1, 2, 4)

b15 b8 b7 b0

Symbol	Address	After Reset
TA1, TA2, TA4	0349h - 0348h, 034Bh - 034Ah, 034Fh - 034Eh	Undefined
TA11, TA21, TA41	0303h - 0302h, 0305h - 0304h, 0307h - 0306h	Undefined

Function	Setting Range	RW
If a setting value is n, f1 is counted n times after a start trigger occurs, and then the timer stops. Output signal level for each phase changes when timers A1, A2, or A4 stop.	0000h to FFFFh	WO

NOTES:

1. Write these registers in 16-bit units. Read-modify-write instructions cannot be used to set registers TAi and TAi1. Refer to **Usage Notes** for details.
2. If the TAi or TAi1 register is set to 0000h, the counter does not start and the timer Ai interrupt is not generated.
3. When the INV15 bit in the INVC1 register is set to 0 (dead timer enabled), an output signal is switched to its active level with delay simultaneously with the dead time timer underflow.
4. When the INV11 bit is set to 0 (Timers A11, A21, and A41 not used (three-phase mode 0)), the contents of the TAi register are transferred to the reload register by a timer Ai start trigger. When the INV11 bit is set to 1 (Timers A11, A21, and A41 are used (three-phase mode 1)), the contents of the TAi1 register are transferred by the first timer Ai start trigger, and then contents of the TAi register are transferred by the next timer Ai start trigger. Subsequently, the contents of registers TAi1 and TAi are transferred alternately to the reload register by each timer Ai start trigger.
5. Do not set registers TAi and TAi1 in the timer B2 underflow timing.

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

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
IDB0, IDB1	030Ah, 030Bh	XX11 1111b

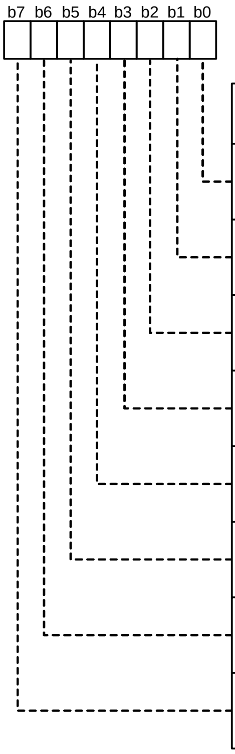
Bit Symbol	Bit Name	Function	RW
DUi	Upper arm (U-phase) output buffer i	Set output levels of the three-phase output shift registers. The set value is reflected in each turn-on signal as follows: 0: Active (ON) 1: Inactive (OFF) When read, the contents of the three-phase output shift registers are returned.	RW
DUBi	Lower arm ($\overline{U}$-phase) output buffer i		RW
DVi	Upper arm (V-phase) output buffer i		RW
DVBi	Lower arm ($\overline{V}$-phase) output buffer i		RW
DWi	Upper arm (W-phase) output buffer i		RW
DWBi	Lower arm ($\overline{W}$-phase) output buffer i		RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:

1. When values are written to registers IDB0 and IDB1, these values are transferred to the three-phase output shift registers by a transfer trigger. The value written in the IDB0 register becomes the initial output level of each phase when the transfer trigger occurs. The value written in the IDB1 register becomes the next output signal level when the falling edge of the timer A1, A2 and A4 one-shot pulses is detected.

Figure 16.9 TA1, TA2, TA4, TA11, TA21, and TA41 Registers, IDB0, IDB1 Registers

Count Start Register



Symbol TABSR

Address 0340h

After Reset 00h

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

Figure 16.10 TABSR Register when Using Three-Phase Motor Control Timer Function

Table 16.2 Pin Settings when Using Three-Phase Motor Control Timer Function[1]

Port	Function	Bit Setting		
		PSC Register	PSL1, PSL2, Registers	PS1, PS2 Registers[2]
P7_2	V	PSC_2 = 1	PSL1_2 = 0	PS1_2 = 1
P7_3	$\overline{V}$	–	PSL1_3 = 1	PS1_3 = 1
P7_4	W	–	PSL1_4 = 1	PS1_4 = 1
P7_5	$\overline{W}$	–	PSL1_5 = 0	PS1_5 = 1
P8_0	U	–	PSL2_0 = 1	PS2_0 = 1
P8_1	$\overline{U}$	–	PSL2_1 = 0	PS2_1 = 1

NOTES:

1. Set these registers after setting the INV02 bit in the INVC0 register to 1 (three-phase motor control timer function used).
2. Set registers PS1 and PS2 after setting the other registers.

16.1 Triangular Wave Modulation Mode

In triangular wave modulation mode, one cycle of carrier waveform consists of two timer B2 underflow cycles.
A timer Ai one-shot pulse (i = 1, 2, and 4) is generated by using a timer B2 underflow signal as a trigger. Two of the timer Ai one-shot pulses are used to output one cycle of the PWM waveform. Table 16.3 lists specifications and settings of triangular wave modulation mode.
Triangular wave modulation mode has two operation modes, three-phase mode 0 and three-phase mode 1.
TAi register is used in three-phase mode 0. Every time a timer B2 underflow interrupt occurs, the one-shot pulse width is set in the TAi register.
Registers TAi and TAi1 are used in three-phase mode 1. Two different widths of the one-shot pulse can be set in these registers. If a setting value of the ICTB2 register is n, a timer B2 underflow interrupt is generated every n-th or every 2n-th timer B2 underflow to set values in registers TAi and TAi1.

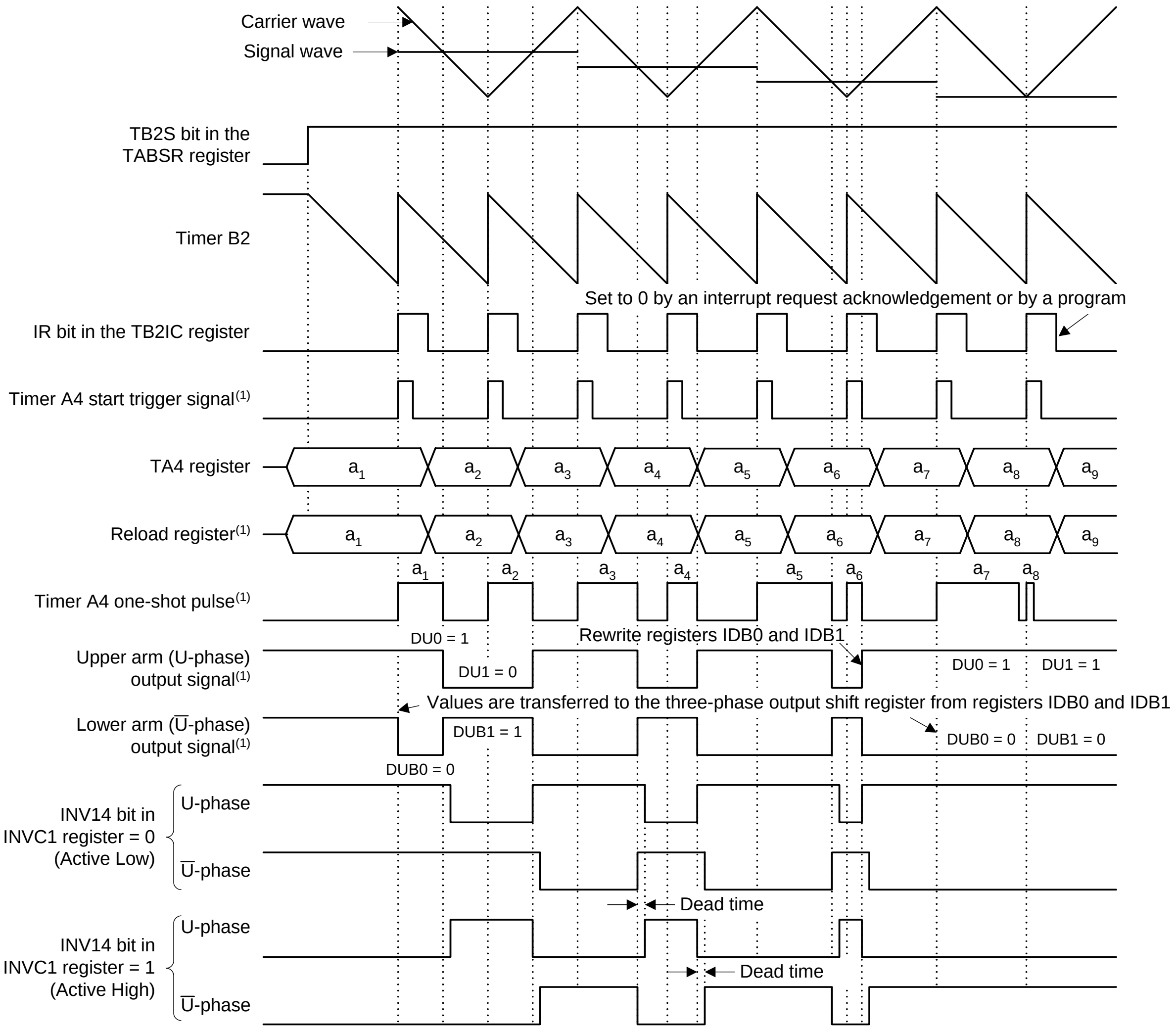
Table 16.3 Specifications and Settings of Triangular Wave Modulation Mode

Item	Three-Phase Mode 0	Three-Phase Mode 1		
INV06 bit	0	0		
INV11 bit	0	1		
Bits INV01 and INV00	00b or 01b	00b	10b	11b
PWCON bit	0	0 or 1		
ICTB2 register	1	n		
Carrier wave cycle	$\frac{2}{f1} \times (m + 1)$	$\frac{2}{f1} \times (m+1)$		
Upper arm active level output width	$\frac{1}{f1} \times(m+1 - a_{2k-1}+a_{2k})$	$\frac{1}{f1} \times (m+1 - b_k+a_k)$		
INV13 bit	0 or 1	Indicates the timer A1 reload control signal state.		
Timer B2 interrupt generation timing	Timer B2 underflow	Every n-th timer B2 underflow	Every 2n-th timer B2 underflow	
			Every odd-numbered (2n × j - 1) timer B2 underflow	Every even-numbered (2n × j) timer B2 underflow
Timer B2 reload timing	Timer B2 underflow	• Timer B2 underflow (PWCON = 0) • Timer B2 underflow at the rising edge of the timer A1 reload control signal (PWCON = 1)		
Transfer timing from IDBp register to three-phase output shift register	When a value is written to the IDBp register (p = 0, 1), the value is transferred only once by the first transfer trigger.			
Dead time timer start timing	• At the falling edge of the one-shot pulse of timer A1, A2 and A4 (INV16 = 0) • At the rising edge of the three-phase output shift register (INV16 = 1)			

m: Value of the TB2 register
a_{2k-1}: Value set to the TAi register at odd-numbered time.
a_{2k}: Value set to the TAi register at even-numbered time.
b_k: Value set to the TAi1 register at k-th time.
a_k: Value set to the TAi register at k-th time.
j: the number of interrupts

Figure 16.11 shows an example of the triangular wave modulation operation (three-phase mode 0). Figures 16.12 and 16.13 show examples of the triangular wave modulation operation (three-phase mode 1).

Triangular Waveform as a Carrier Wave (Three-phase mode 0)


Carrier wave
Signal wave
TB2S bit in the TABSR register
Timer B2
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TB2IC register
Timer A4 start trigger signal(1)
TA4 register
a1 a2 a3 a4 a5 a6 a7 a8 a9
Reload register(1)
a1 a2 a3 a4 a5 a6 a7 a8 a9
a1 a2 a3 a4 a5 a6 a7 a8
Timer A4 one-shot pulse(1)
DU0 = 1
Rewrite registers IDB0 and IDB1
Upper arm (U-phase) output signal(1)
DU1 = 0
DU0 = 1
DU1 = 1
Values are transferred to the three-phase output shift register from registers IDB0 and IDB1
Lower arm (U̅-phase) output signal(1)
DUB1 = 1
DUB0 = 0
DUB1 = 0
DUB0 = 0
INV14 bit in INVC1 register = 0 (Active Low)
U-phase
U̅-phase
Dead time
INV14 bit in INVC1 register = 1 (Active High)
U-phase
Dead time
U̅-phase

NOTE:

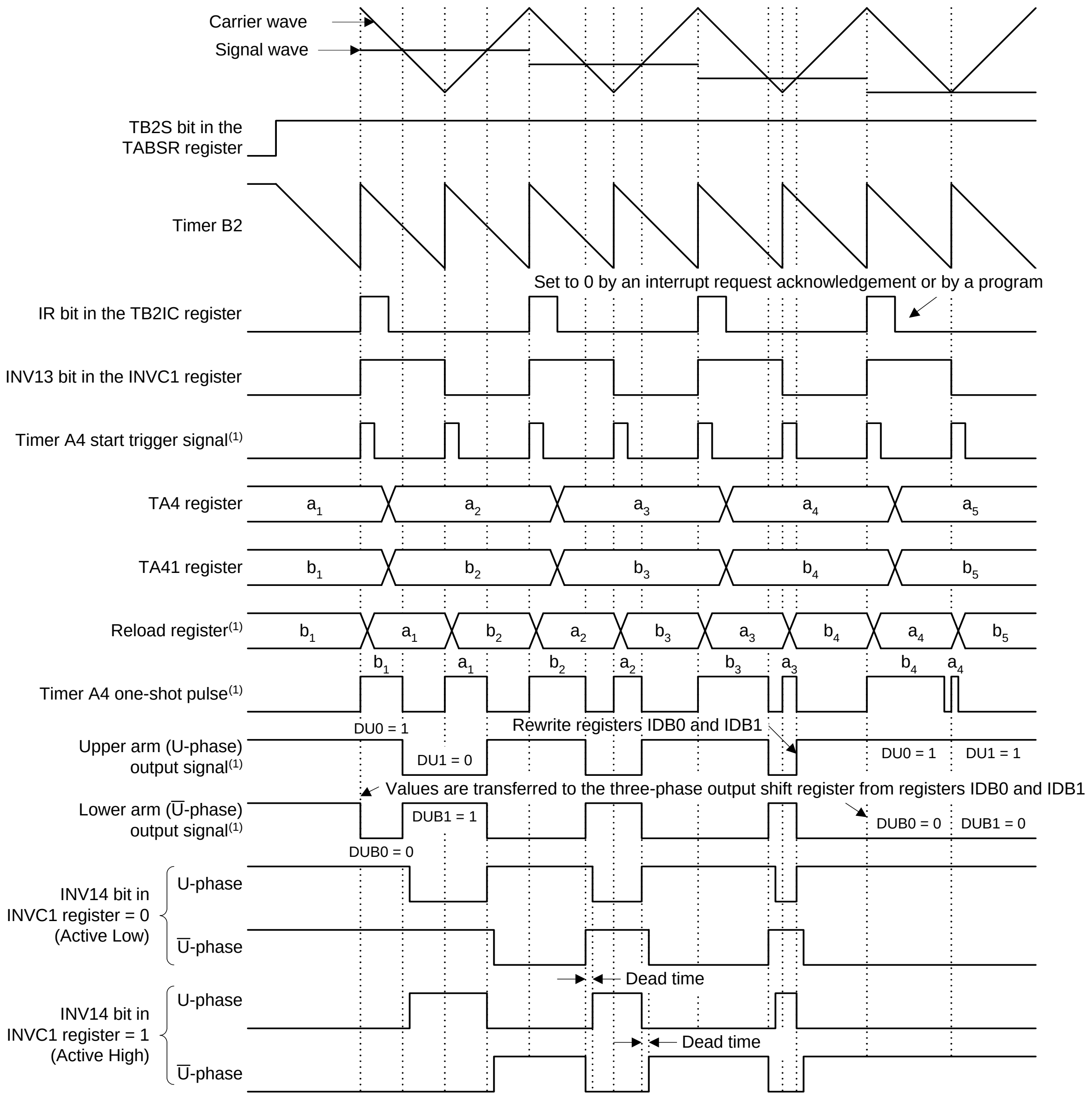
1. Internal signals. See **Three-Phase Motor Control Timer Function Block Diagram.**

The above applies under the following conditions:
- INVC0 register: INV01 bit = 0 (ICTB2 counter is incremented by 1 when timer B2 underflows)
  INV02 bit = 1 (Three-phase motor control timer function used)
  INV03 bit = 1 (Three-phase motor control timer output enabled)
  INV06 bit = 0 (Triangular wave modulation mode)
- INVC1 register: INV10 bit = 0 (Timer B2 underflow)
  INV11 bit = 0 (Timers A11, A21, A41 not used (Three-phase mode 0))
  INV15 bit = 0 (Dead time enabled)
  INV16 bit = 1 (Rising edge of the three-phase output shift register (U-, V-, W-phase))
- ICTB2 register = 01h (Timer B2 interrupt is generated every timer B2 underflow)

The following shows examples to change PWM output levels.
- Default value of the timer: TA4 = a_1 (The TA4 register is rewritten every time the timer B2 interrupt occurs.)
  First time TA4 = a_2, second time TA4 = a_3, third time TA4 = a_4, fourth time TA4 = a_5, fifth time TA4 = a_6
- Default value of the registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, and DUB1 = 1
  They are changed to DU0 = 1, DUB0 = 0, DU1 = 1, and DUB1 = 0 at the sixth timer B2 interrupt.

Figure 16.11 Triangular Wave Modulation Operation (Three-Phase Mode 0)

Triangular Waveform as a Carrier Wave (Three-phase mode 1: INV01 and INV00 = 10b)


Carrier wave
Signal wave
TB2S bit in the TABSR register
Timer B2
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TB2IC register
INV13 bit in the INVC1 register
Timer A4 start trigger signal(1)
TA4 register
a_1 a_2 a_3 a_4 a_5
TA41 register
b_1 b_2 b_3 b_4 b_5
Reload register(1)
b_1 a_1 b_2 a_2 b_3 a_3 b_4 a_4 b_5
b_1 a_1 b_2 a_2 b_3 a_3 b_4 a_4
Timer A4 one-shot pulse(1)
DU0 = 1
Rewrite registers IDB0 and IDB1
Upper arm (U-phase) output signal(1)
DU1 = 0
DU0 = 1
DU1 = 1
Values are transferred to the three-phase output shift register from registers IDB0 and IDB1
Lower arm (Ū-phase) output signal(1)
DUB1 = 1
DUB0 = 0
DUB1 = 0
DUB0 = 0
INV14 bit in INVC1 register = 0 (Active Low)
U-phase
Ū-phase
Dead time
INV14 bit in INVC1 register = 1 (Active High)
U-phase
Dead time
Ū-phase

NOTE:

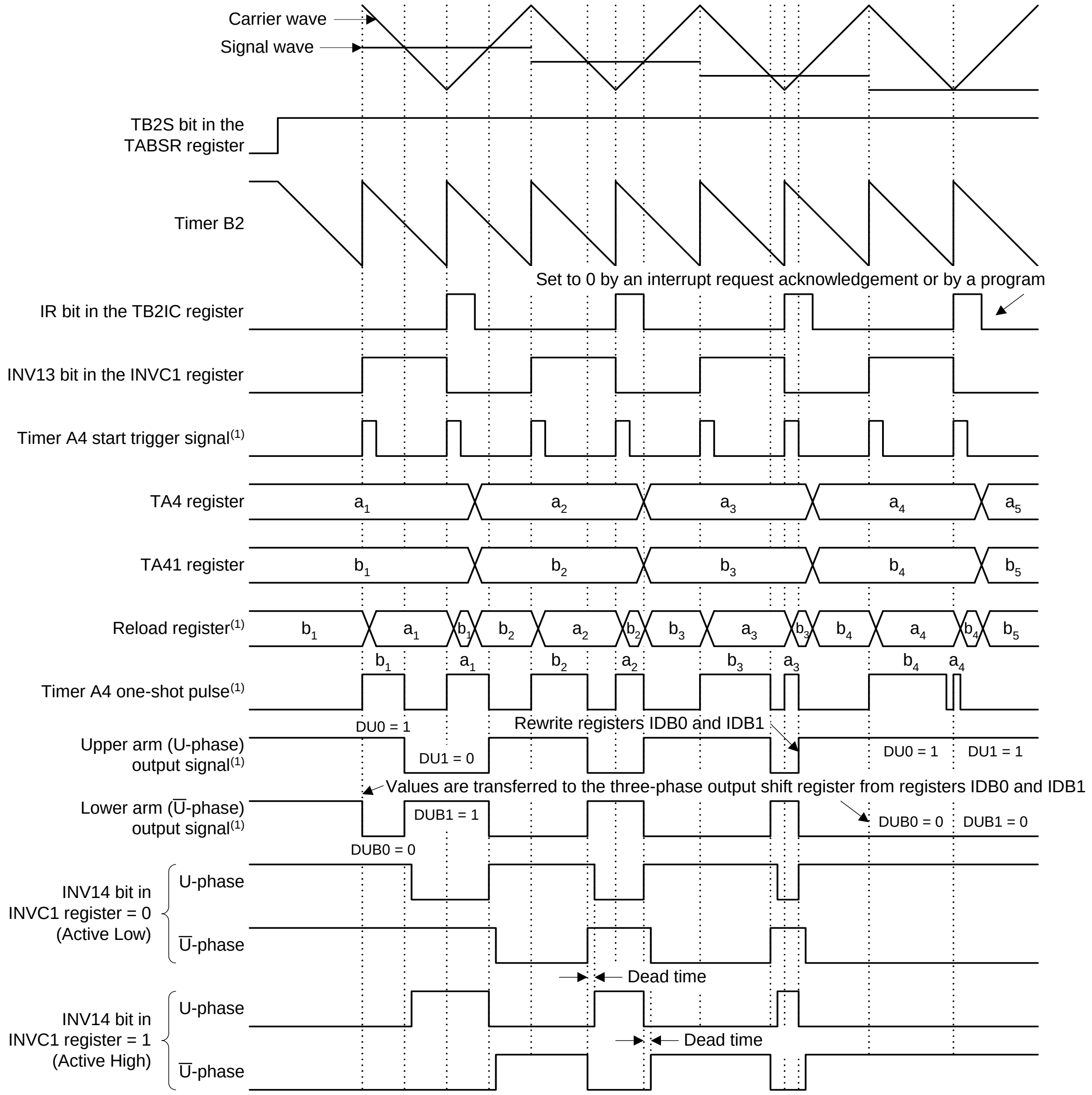
1. Internal signals. See **Three-Phase Motor Control Timer Function Block Diagram.**

The above applies under the following conditions:
- INVC0 register: Bits INV01 and INV00 = 10b (ICTB2 counter is incremented by 1 at the rising edge of the timer A1 reload control signal)
  INV02 bit = 1 (Three-phase motor control timer function used)
  INV03 bit = 1 (Three-phase motor control timer output enabled)
  INV06 bit = 0 (Triangular wave modulation mode)
- INVC1 register: INV10 bit = 0 (Timer B2 underflow)
  INV11 bit = 1 (Timer A11, A21, A41 used (Three-phase mode 1))
  INV15 bit = 0 (Dead time enabled)
  INV16 bit = 1 (Rising edge of the three-phase output shift register (U-, V-, W-phase))
- ICTB2 register = 01h (First timer B2 interrupt occurs when timer B2 underflows for the first time, and the subsequent interrupts occur every second timer B2 underflow.)

The following shows examples to change PWM output levels.
- Default value of the timer: TA41 = b_1, TA4 = a_1 (Registers TA4 and TA41 are rewritten every time the timer B2 interrupt occurs.)
  First time TA41 = b_2, TA4 = a_2, second time TA41 = b_3, TA4 = a_3
- Default value of the registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, and DUB1 = 1
  They are changed to DU0 = 1, DUB0 = 0, DU1 = 1, and DUB1 = 0 at the third timer B2 interrupt.

Figure 16.12 Triangular Wave Modulation Operation (Three-Phase Mode 1)(INV01 and INV00 = 10b)

Triangular Waveform as a Carrier Wave (Three-phase mode 1: INV0 and INV00 = 11b)


Carrier wave
Signal wave
TB2S bit in the TABSR register
Timer B2
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TB2IC register
INV13 bit in the INVC1 register
Timer A4 start trigger signal(1)
TA4 register
a_1 a_2 a_3 a_4 a_5
TA41 register
b_1 b_2 b_3 b_4 b_5
Reload register(1)
b_1 a_1 b_1 b_2 a_2 b_2 b_3 a_3 b_3 b_4 a_4 b_4 b_5
b_1 a_1 b_2 a_2 b_3 a_3 b_4 a_4
Timer A4 one-shot pulse(1)
DU0 = 1
Rewrite registers IDB0 and IDB1
Upper arm (U-phase) output signal(1)
DU1 = 0
DU0 = 1
DU1 = 1
Values are transferred to the three-phase output shift register from registers IDB0 and IDB1
Lower arm ($\overline{U}$-phase) output signal(1)
DUB1 = 1
DUB0 = 0
DUB1 = 0
DUB0 = 0
INV14 bit in INVC1 register = 0 (Active Low)
U-phase
$\overline{U}$-phase
Dead time
INV14 bit in INVC1 register = 1 (Active High)
U-phase
$\overline{U}$-phase
Dead time

NOTE:
1. Internal signals. See **Three-Phase Motor Control Timer Function Block Diagram.**

The above applies under the following conditions:
- INVC0 register: Bits INV01 and INV00 = 11b (ICTB2 counter is incremented by 1 at the falling edge of the timer A1 reload control signal)
  INV02 bit = 1 (Three-phase control timer function used)
  INV03 bit = 1 (Three-phase control timer output enabled)
  INV06 bit = 0 (Triangular wave modulation mode)
- INVC1 register: INV10 bit = 0 (Timer B2 underflow)
  INV11 bit = 1 (Timers A11, A21, A41 used (Three-phase mode 1))
  INV15 bit = 0 (Dead time enabled)
  INV16 bit = 1 (Rising edge of the three-phase output shift register (U-, V-, W-phase))
- ICTB2 register = 01h (Every second timer B2 underflow.)
  (ICTB2 register = 02h, if INV01 bit = 0)

The following shows examples to change PWM output levels.
- Default value of the timer: TA41 = b_1, TA4 = a_1 (Registers TA4 and TA41 are rewritten every time the timer B2 interrupt occurs.)
  First time TA41 = b_2, TA4 = a_2, second time TA41 = b_3, TA4 = a_3
- Default value of the registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, and DUB1 = 1
  They are changed to DU0 = 1, DUB0 = 0, DU1 = 1, and DUB1 = 0 at the third timer B2 interrupt.

Figure 16.13 Triangular Wave Modulation Operation (Three-Phase Mode 1)(INV01 and INV00 = 11b)

16.2 Sawtooth Wave Modulation Mode

In sawtooth wave modulation mode, one cycle of carrier waveform consists of one timer B2 underflow cycle.
A timer Ai one-shot pulse (i = 1, 2, and 4) is generated by using a timer B2 underflow signal as a trigger. Single one-shot pulse from timer Ai is used to output one cycle of the PWM waveform. Table 16.4 lists specifications and settings of sawtooth wave modulation mode.

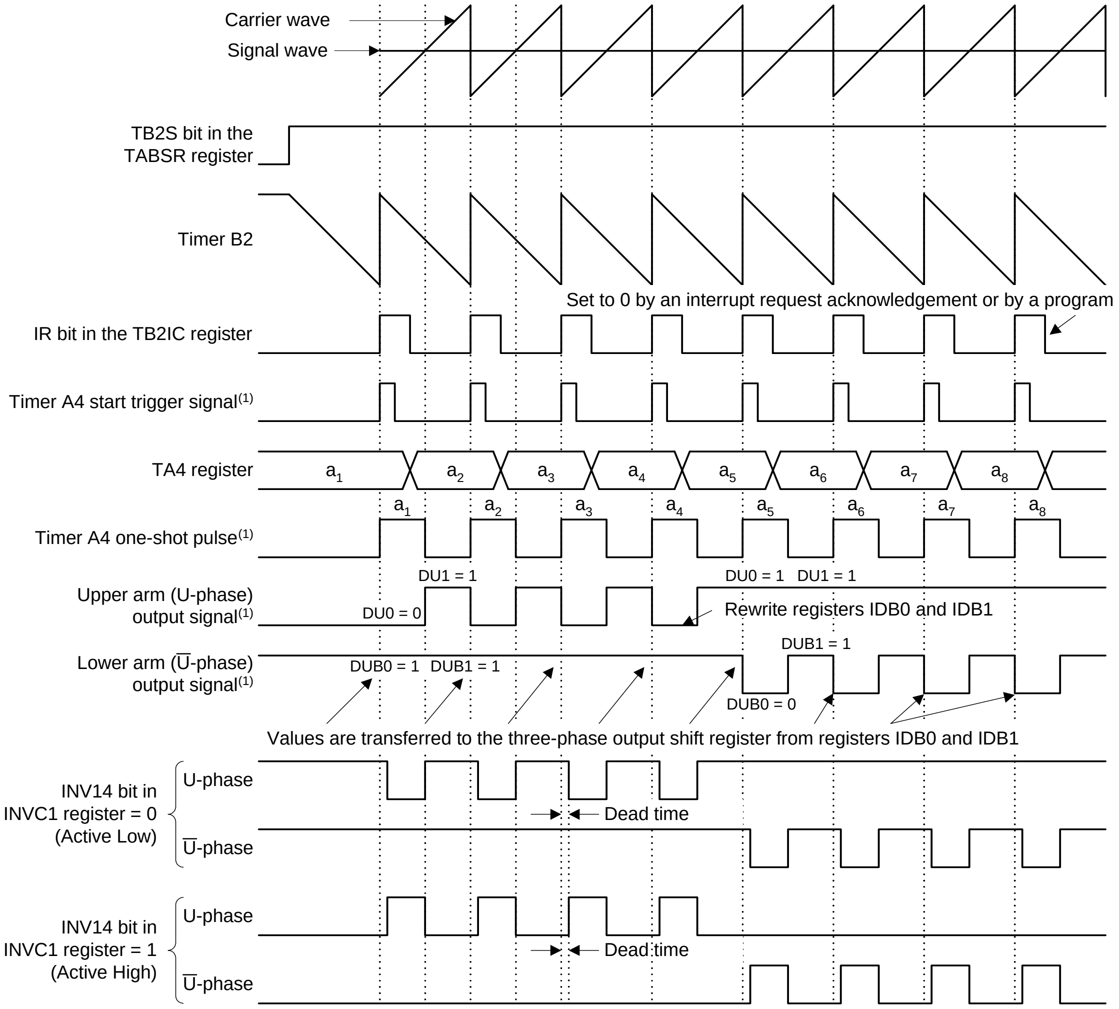
Table 16.4 Specifications and Settings of Sawtooth Wave Modulation Mode

Item	Three-Phase Mode 0
INV06 bit	1
INV11 bit	0
Bits INV01 and INV00	00b or 01b
PWCON bit	0
ICTB2 register	n
INV16 bit	0
Carrier wave cycle	$\frac{1}{f1} \times (m + 1)$
Upper arm active level output width	$\frac{1}{f1} \times a_k$
Timer B2 interrupt generation timing	Every n-th timer B2 underflow
Timer B2 reload timing	Timer B2 underflow
Transfer timing from IDBp register to three-phase output shift register (p = 0, 1)	Every time a transfer trigger occurs.
Dead time timer start timing	• At the falling edge of the one-shot pulse of timer A1, A2 and A4 • Transfer trigger

m: Value of the TB2 register
a_k: Value set to the TAi register at k-th time.

Figure 16.14 shows an example of the sawtooth wave modulation operation.

Sawtooth Waveform as a Carrier Wave


Carrier wave
Signal wave
TB2S bit in the TABSR register
Timer B2
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the TB2IC register
Timer A4 start trigger signal(1)
TA4 register
a1 a2 a3 a4 a5 a6 a7 a8
Timer A4 one-shot pulse(1)
Upper arm (U-phase) output signal(1)
DU1 = 1
DU0 = 0
DU0 = 1 DU1 = 1
Rewrite registers IDB0 and IDB1
Lower arm (U̅-phase) output signal(1)
DUB0 = 1 DUB1 = 1
DUB1 = 1
DUB0 = 0
Values are transferred to the three-phase output shift register from registers IDB0 and IDB1
INV14 bit in INVC1 register = 0 (Active Low)
U-phase
U̅-phase
Dead time
INV14 bit in INVC1 register = 1 (Active High)
U-phase
U̅-phase
Dead time

NOTE:
1. Internal signals. See **Three-Phase Motor Control Timer Function Block Diagram.**

The above applies under the following conditions:
- INVC0 register: INV01 bit = 0 (ICTB2 counter is incremented by 1 when timer B2 underflows)
  INV02 bit = 1 (Three-phase motor control timer function used)
  INV03 bit = 1 (Three-phase motor control timer output enabled)
  INV06 bit = 1 (Sawtooth wave modulation mode)
- INVC1 register: INV10 bit = 0 (Timer B2 underflow)
  INV11 bit = 0 (Timers A11, A21, A41 not used (Three-phase mode 0))
  INV15 bit = 0 (Dead time enabled)
  INV16 bit = 0 (Falling edge of one-shot pulse of timers A1, A2, and A4)
- ICTB2 register = 01h (Timer B2 interrupt is generated every timer B2 underflow)
- TB2SC register: PWCON bit = 0 (Timer B2 underflow)

The following shows examples to change PWM output levels.
- Default value of the timer: TA4 = a_1 (The TA4 register is changed every time the timer B2 interrupt occurs.)
  First time TA4 = a_2, second time TA4 = a_3, third time TA4 = a_4, fourth time = a_5
- Default value of the registers IDB0 and IDB1: DU0 = 0, DUB0 = 1, DU1 = 1, and DUB1 = 1
  They are changed to DU0 = 1, DUB0 = 0, DU1 = 1, and DUB1 = 1 at the fourth timer B2 interrupt.

Figure 16.14 Sawtooth Wave Modulation Operation

16.3 Short Circuit Prevention Features

16.3.1 Prevention Against Upper/Lower Arm Short Circuit by Program Errors

This function prevents the upper and lower arm short circuit caused by setting the upper and lower output buffers in registers IDB0 and IDB1 to active simultaneously by program errors and so on.
To use this function, set the INV04 bit in the INVC0 register to 1 (simultaneous turn-on signal output disabled). If any pair of output buffers (U and $\overline{\text{U}}$, V and $\overline{\text{V}}$, or W and $\overline{\text{W}}$) are simultaneously set to active, the INV05 bit becomes 1 (detected), and the INV03 bit becomes 0 (three-phase motor control timer output disabled). Then, the port outputs are forcibly cutoff and the pins are placed in the high-impedance states. When this prevention function is performed, set the registers associated with the three-phase motor control timer function again.

16.3.2 Arm Short Circuit Prevention Using Dead Time Timer

The dead time timer prevents arm short circuit caused by turn-off delay of external upper and lower transistors. To enable the dead time timer, set the INV15 bit in the INVC1 register to 0 (dead time enabled). The count source for dead time timer (fDT) can be selected using the INV12 bit, and the dead time can be set using the DTT register.
The dead time is obtained from the following formulas.

$$\frac{1}{\text{f1}} \times \text{n} \quad (\text{INV12} = 0)$$

$$\frac{2}{\text{f1}} \times \text{n} \quad (\text{INV12} = 1) \qquad \text{n: Value in the DTT register}$$

Figure 16.15 shows an example of dead time timer operation.

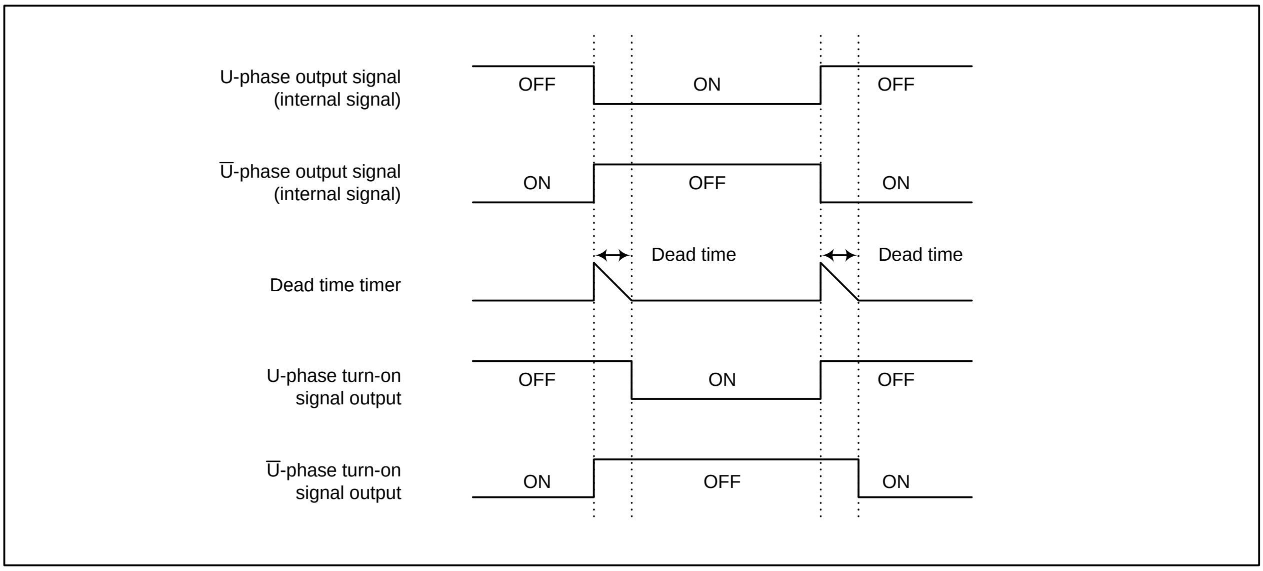

U-phase output signal (internal signal)
OFF
ON
OFF
$\overline{\text{U}}$-phase output signal (internal signal)
ON
OFF
ON
Dead time timer
Dead time
Dead time
U-phase turn-on signal output
OFF
ON
OFF
$\overline{\text{U}}$-phase turn-on signal output
ON
OFF
ON

Figure 16.15 Dead Time Timer Operation

16.3.3 Forced-Cutoff Function by the $\overline{\text{NMI}}$ Input

When an “L” signal is input to the $\overline{\text{NMI}}$ pin, the INV03 bit in the INVC0 register becomes 0 (three-phase motor control timer output disabled), the port outputs are forcibly cutoff, and then the pins are placed in the high-impedance states. Also, the $\overline{\text{NMI}}$ interrupt occurs at the same time.
To enable the three-phase motor control timer function after the forced cutoff is performed, set the registers associated with the three-phase motor control timer function again while an “H” signal is input to the $\overline{\text{NMI}}$ pin.
Forced-cutoff function by the $\overline{\text{NMI}}$ input can be used when the INV02 bit in the INVC0 register is set to 1 (three-phase motor control timer function used) and the INV03 bit is set to 1 (three-phase motor control timer output enabled).

17. Serial Interfaces

Serial interfaces consist of five channels (UART0 to UART4).
Each UARTi (i = 0 to 4) has an exclusive timer to generate the serial clock and operates independently of each other.
UARTi has the following modes.

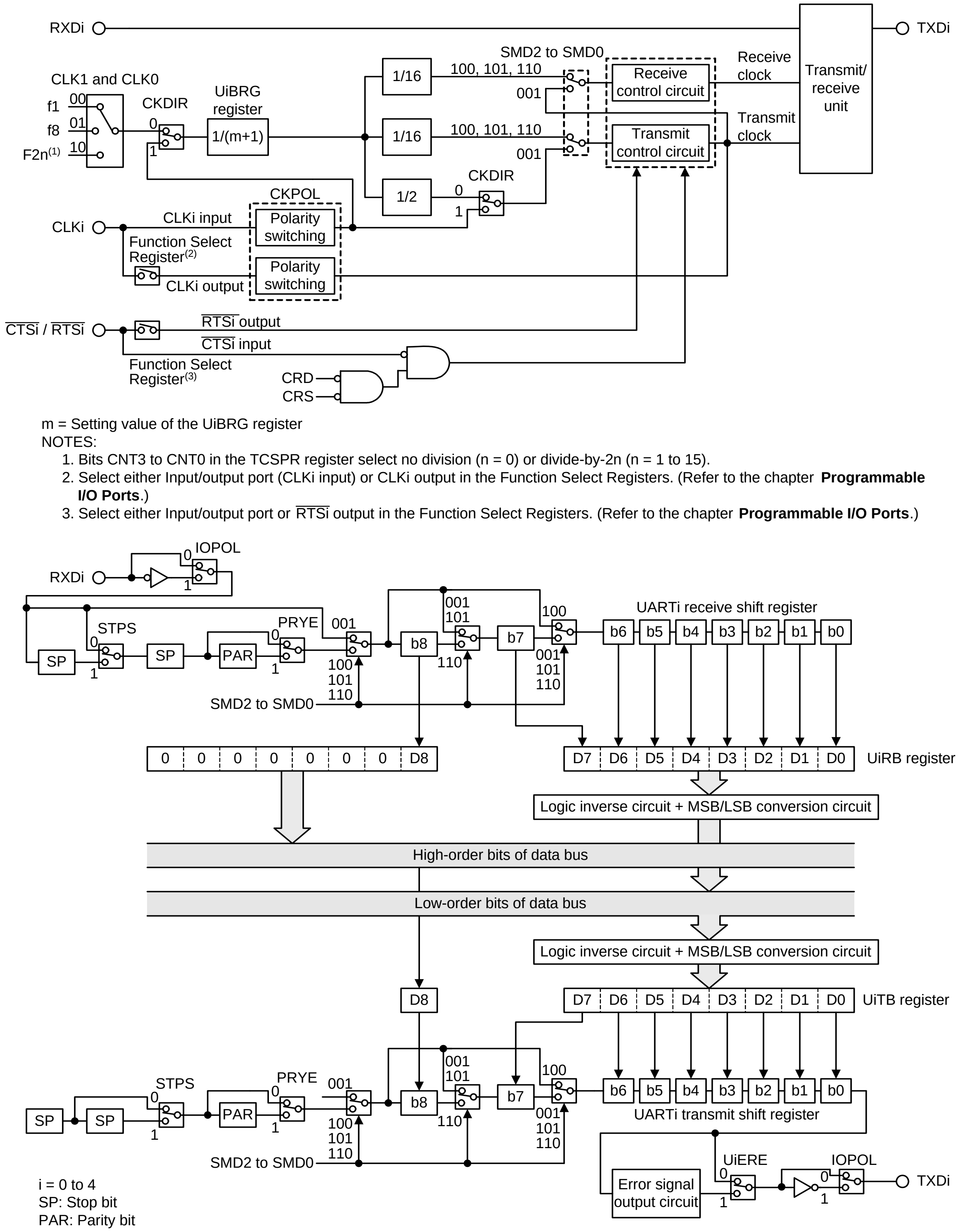
- Clock synchronous mode
- Clock asynchronous mode
- Special mode 1 (I^{2}C mode)
- Special mode 2
- Special mode 3 (clock-divided synchronous function, GCI mode)
- Special mode 4 (SIM mode)
- Special mode 5 (bus conflict detect function, IE mode) (optional)(1)

NOTE:

1. Please contact a Renesas sales office for optional features.

17.1 UART0 to UART4

Figure 17.1 shows a UART0 to UART4 block diagram. Figures 17.2 to 17.10 show the registers associated with UART0 to UART4. Refer to the tables listing for register and pin settings in each mode.



m = Setting value of the UiBRG register

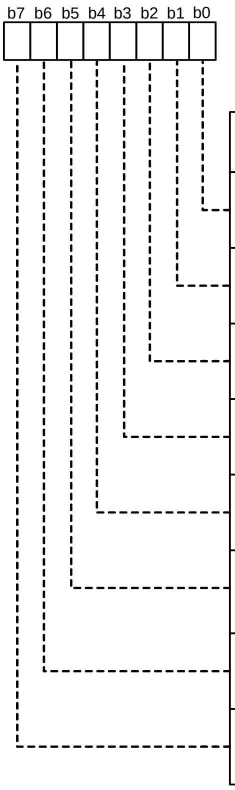
NOTES:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Select either Input/output port (CLKi input) or CLKi output in the Function Select Registers. (Refer to the chapter **Programmable I/O Ports**.)
3. Select either Input/output port or RTSi output in the Function Select Registers. (Refer to the chapter **Programmable I/O Ports**.)

i = 0 to 4
SP: Stop bit
PAR: Parity bit
SMD2 to SMD0, STPS, PRYE, IOPOL, and CKDIR: bits in the UiMR register
CLK1 and CLK0, CKPOL, CRD, and CRS: bits in the UiC0 register
UiERE: bit in the UiC1 register

Figure 17.1 UART0 to UART 4 Block Diagram

UARTi transmit/receive mode register (i = 0 to 4)

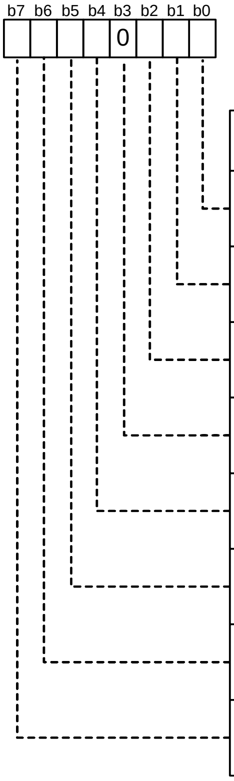


Symbol	Address	After Reset
U0MR to U2MR	0368h, 02E8h, 0338h	00h
U3MR, U4MR	0328h,02F8h	00h

Bit Symbol	Bit Name	Function	RW
SMD0	Serial interface mode select bits	b2 b1 b0 0 0 0: Serial interface disabled 0 0 1: Clock synchronous mode 0 1 0: I^2C mode 1 0 0: UART mode, 7-bit data length 1 0 1: UART mode, 8-bit data length 1 1 0: UART mode, 9-bit data length Do not set to values other than the above	RW
SMD1			RW
SMD2			RW
CKDIR	Clock select bit	0: Internal clock 1: External clock	RW
STPS	Stop bit length select bit	0: 1 stop bit 1: 2 stop bits	RW
PRY	Parity select bit	Enabled when PRYE=1 0: Odd parity 1: Even parity	RW
PRYE	Parity enable bit	0: Parity disabled 1: Parity enabled	RW
IOPOL	TXD, RXD input/output polarity switch bit	0: Not inverted 1: Inverted	RW

Figure 17.2 U0MR to U4MR Registers

UARTi Special Mode Register (i = 0 to 4)



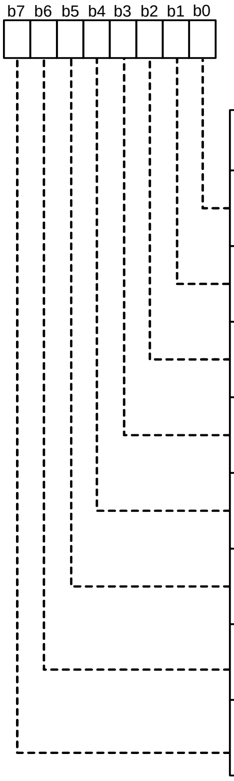
Symbol	Address	After Reset
U0SMR to U2SMR	0367h, 02E7h, 0337h	00h
U3SMR, U4SMR	0327h, 02F7h	00h

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 detect flag control bit(1)	0: Updated per bit 1: Updated per byte	RW
BBS	Bus busy flag(1, 2)	0: Stop condition detected (bus is free) 1: Start condition detected (bus is busy)	RW
– (b3)	Reserved bit	Set to 0	RW
ABSCS	Bus conflict detect sampling clock select bit(3)	0: Rising edge of serial clock 1: Timer Aj underflow (j = 0, 3, 4)(4)	RW
ACSE	Auto clear function select bit for transmit enable bit(3)	0: No auto clear function 1: Auto cleared when bus conflict occurs	RW
SSS	Transmit start condition select bit(3)	0 : Not related to RXDi 1 : Synchronized with RXDi	RW
SCLKDIV	Clock division synchronous bit(5,6)	0: External clock not divided 1: External clock divided by 2	RW

NOTES:
1. These bits are used in I^{2}C mode.
2. The BBS bit can be set to 0 by a program. Writing a 1 has no effect.
3. These bits are used in IE mode.
4. UART0: Timer A3 underflow signal, UART1: Timer A4 underflow signal, UART2: Timer A0 underflow signal, UART3: Timer A3 underflow signal, UART4: Timer A4 underflow signal.
5. The SCLKDIV bit is used in GCI mode.
6. Refer to the note for the SU1HIM bit in the UiSMR2 register.

Figure 17.3 U0SMR to U4SMR Registers

UARTi Special Mode Register 2 (i = 0 to 4)



Symbol	Address	After Reset
U0SMR2 to U2SMR2	0366h, 02E6h, 0336h	00h
U3SMR2, U4SMR2	0326h, 02F6h	00h

Bit Symbol	Bit Name	Function	RW
IICM2	I^2C mode select bit 2	0: ACK/NACK interrupt used 1: Transmit/receive interrupt used	RW
CSC	Clock synchronous bit(1)	0: Not clock synchronized 1: Clock synchronized	RW
SWC	SCL wait output bit(2)	0: No wait state/release wait states 1:SCLi pin is held "L" after receiving 8th bit.	RW
ALS	SDA output auto stop bit(1)	When arbitration lost is detected, 0: SDAi output not stopped 1: SDAi output stopped	RW
STC	UARTi auto initialization bit(2)	When start condition is detected, 0: UARTi not initialized 1: UARTi initialized	RW
SWC2	SCL wait output bit 2(1)	0: Serial clock output from SCLi pin 1: SCLi pin is held "L"	RW
SDHI	SDA output stop bit(2)	0: Output data 1: Output stopped (Hi-impedance state)	RW
SU1HIM	External clock synchronous enable bit(3)	0: Not synchronized with external clock 1: Synchronized with external clock	RW

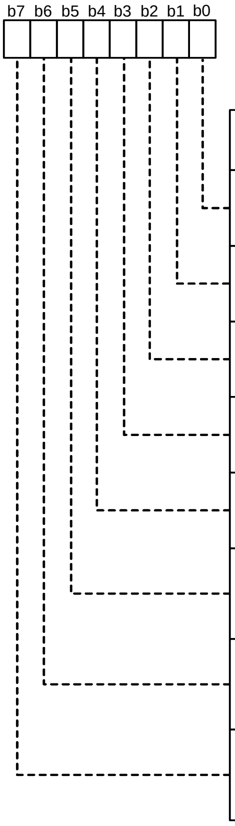
NOTES:

1. These bits are used when the MCU is in master mode in I^2C mode.
2. These bits are used when the MCU is in slave mode in I^2C mode.
3. The external clock synchronous function can be selected with the combination of the SU1HIM bit and the SCLKDIV bit in the UiSMR register. The SU1HIM bit is used in GCI mode.

SCLKDIV Bit in the UiSMR register	SU1HIM Bit in the UiSMR2 register	External Clock Synchronous Function Select
0	0	Not synchronized
0	1	Same frequency as external clock
1	0 or 1	External clock divided by 2

Figure 17.4 U0SMR2 to U4SMR2 Registers

UARTi Special Mode Register 3 (i = 0 to 4)



Symbol	Address	After Reset
U0SMR3 to U2SMR3	0365h, 02E5h, 0335h	00h
U3SMR3, U4SMR3	0325h, 02F5h	00h

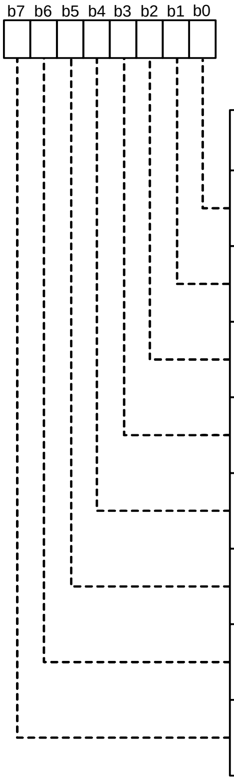
Bit Symbol	Bit Name	Function	RW
SSE	SS function enable bit[1]	0: SS function disabled 1: SS function enabled[2]	RW
CKPH	Clock phase set bit[1]	0: No Clock delay 1: Clock delay	RW
DINC	Serial I/O pin set bit[1]	0: Pins TXDi and RXDi selected (master mode) 1: Pins STXDi and SRXDi selected (slave mode)	RW
NODC	Clock output select bit	0: CLKi is CMOS output 1: CLKi is N-channel open drain output	RW
ERR	Mode error flag[1]	0: No mode error 1: Mode error occurred[3]	RW
DL0	SDAi digital delay set bits[4, 5]	SDAi output is delayed by the following cycles. b7 b6 b5 0 0 0: No delay 0 0 1: 1-to-2 cycles of BRG count source 0 1 0: 2-to-3 cycles of BRG count source 0 1 1: 3-to-4 cycles of BRG count source 1 0 0: 4-to-5 cycles of BRG count source 1 0 1: 5-to-6 cycles of BRG count source 1 1 0: 6-to-7 cycles of BRG count source 1 1 1: 7-to-8 cycles of BRG count source	RW
DL1			RW
DL2			RW

NOTES:

1. These bits are used in special mode 2.
2. When the $\overline{\text{SS}}$ pin is set to 1, set the CRD bit in the UiC0 register to 1 (CTS function disabled).
3. The ERR bit can be set to 0 by a program. Writing a 1 has no effect.
4. Digital delay is added to a SDAi output using bits DL2 to DL0 in I^{2}C mode. Set them to 000b (no delay) in other than I^{2}C mode.
5. When the external clock is selected, SDAi output is delayed by approximately 100 ns in addition.

Figure 17.5 U0SMR3 to U4SMR3 Registers

UARTi Special Mode Register 4 (i = 0 to 4)



Symbol	Address	After Reset
U0SMR4 to U2SMR4	0364h, 02E4h, 0334h	00h
U3SMR4, U4SMR4	0324h, 02F4h	00h

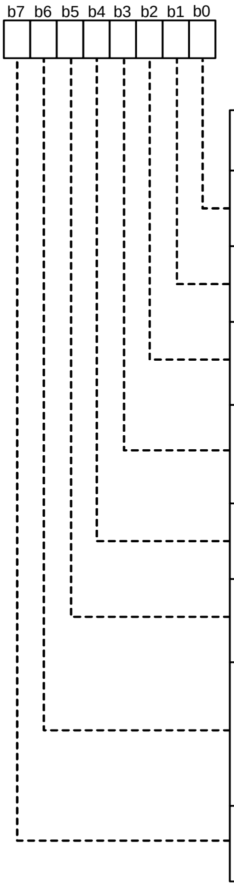
Bit Symbol	Bit Name	Function	RW
STAREQ	Start condition generate bit[1, 3]	0: Clear 1: Start	RW
RSTAREQ	Restart condition generate bit[1, 3]	0: Clear 1: Start	RW
STPREQ	Stop condition generate bit[1, 3]	0: Clear 1: Start	RW
STSPSEL	SCL, SDA output select bit[1, 5]	0: Serial input/output circuit selected 1: Start/stop condition generation circuit selected[4]	RW
ACKD	ACK data bit[2, 5]	0: ACK 1: NACK	RW
ACKC	ACK data output enable bit[2, 5]	0: Serial data output 1: ACK data output	RW
SCLHI	SCL output stop bit[1, 5]	When the bus is free, 0: SCLi output not stopped 1: SCLi output stopped	RW
SWC9	SCL wait output bit 3[1, 5]	0: No wait state/release wait state 1: SCLi pin is held "L" after receiving 9th bit	RW

NOTES:

1. These bits are used when the MCU is in master mode in I^{2}C mode.
2. These bits are used when the MCU is in slave mode in I^{2}C mode.
3. When each condition generation is completed, the corresponding bit becomes 0. When a condition generation is failed, the bit remains 1.
4. Set the STSPSEL bit to 1 (start/stop condition generation circuit selected) after setting the STAREQ bit, RSTAREQ bit, or STPREQ bit to 1 (start).
5. Bits STPSEL, ACKD, ACKC, SCLHI, and SWC9 can be set to 1 when the IICM bit in the UiSMR register is set to 1 (I^{2}C mode). When the IICM bit is set to 0 (Other than I^{2}C mode), these bits become 0.

Figure 17.6 U0SMR4 to U4SMR4 Registers

UARTi Transmit/Receive Control Register 0 (i = 0 to 4)



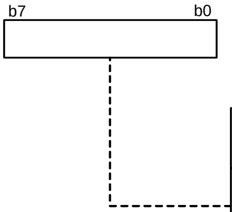
Symbol	Address	After Reset
U0C0 to U2C0	036Ch, 02ECh, 033Ch	0000 1000b
U3C0, U4C0	032Ch, 02FCh	0000 1000b

Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG count source select bits[1]	b1 b0 0 0: f1 selected 0 1: f8 selected 1 0: f2n selected[2] 1 1: Do not set to this value	RW
CLK1			RW
CRS	CTS function select bit	Enabled when CRD = 0 0: CTS function selected 1: CTS function not selected	RW
TXEPT	Transmit shift register empty flag	0: Data in the transmit shift register (during transmit operation) 1: No data in the transmit shift register (transmit operation is completed)	RO
CRD	CTS function disable bit	0: CTS function enabled 1: CTS function disabled	RW
NCH	Data output select bit[3]	0: TXDi/SDAi and SCLi are CMOS output ports 1: TXDi/SDAi and SCLi are N-channel open drain output ports	RW
CKPOL	CLK polarity select bit	0: Transmit data output at the falling edge and receive data input at the rising edge of the serial clock 1: Transmit data output at the rising edge and receive data input at the falling edge of the serial clock	RW
UFORM	Bit order select bit[4]	0 : LSB first 1 : MSB first	RW

NOTES:
1. Set the UiBRG register after setting bits CLK1 and CLK0.
2. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15). To select f2n, set the CST bit in the TCSPR register to 1 before setting bits CLK1 and CLK0 to 10b.
3. P7_0/TXD2, P7_1/SCL2 are N-channel open drain output ports. They cannot be set as CMOS output ports even if the NCH bit is set to 0.
4. The UFORM bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous mode) or 101b (UART mode, 8-bit data length). Set the UFORM bit to 1 when bits SMD2 to SMD0 are set to 010b (I^2C mode), or to 0 when bits SMD2 to SMD0 are set to 100b (UART mode, 7-bit data length) or 110b (UART mode, 9-bit data length).

Figure 17.7 U0C0 to U4C0 Registers

UARTi Baud Rate Register[1, 2] (i = 0 to 4)

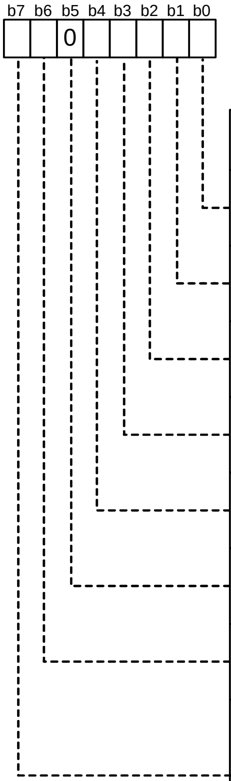


Symbol	Address	After Reset
U0BRG to U2BRG	0369h, 02E9h, 0339h	Undefined
U3BRG, U4BRG	0329h, 02F9h	Undefined

Function	Setting Range	RW
If the setting value is *n*, the UiBRG register divides a count source by *n*+1	00h to FFh	WO

NOTES:
1. Read-modify-write instructions cannot be used to set the UiBRG register. Refer to **Usage Notes** for details.
2. Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register.

UARTi Transmit/Receive Control Register 1 (i = 0 to 4)



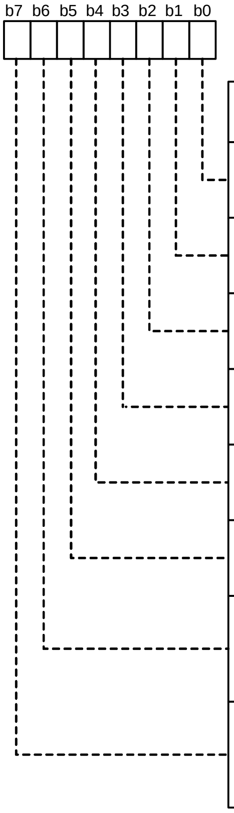
Symbol	Address	After Reset
U0C1 to U2C1	036Dh, 02EDh, 033Dh	0000 0010b
U3C1, U4C1	032Dh, 02FDh	0000 0010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0: Transmit operation disabled 1: Transmit operation enabled	RW
TI	UiTB register empty flag	0: Data in the UiTB register 1: No data in the UiTB register	RO
RE	Receive enable bit	0: Receive operation disabled 1: Receive operation enabled	RW
RI	Receive complete flag	0: No Data in the UiRB register 1: Data in the UiRB register	RO
UiIRS	UARTi transmit interrupt source select bit	0: No data in the UiTB register (TI = 1) 1: Transmit operation is completed (TXEPT = 1)	RW
UiRRM	Continuous receive mode enable bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled[3]	RW
UiLCH	Data logic select bit[1]	0: Not inverted 1: Inverted	RW
SCLKSTPB	Special mode 3 Clock-divided synchronous stop bit	0: Synchronization stopped 1: Synchronization started	RW
UiERE	Special mode 4 Error signal output enable bit[2]	0: Not output 1: Output	

NOTES:
1. The UiLCH bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous mode), 100b (UART mode, 7-bit data length), or 101b (UART mode, 8-bit data length). Set the UiLCH bit to 0 when bits SMD2 to SMD0 are set to 010b (I^{2}C mode) or 110b (UART mode, 9-bit data length).
2. Set bits SMD2 to SMD0 before setting the UiERE bit.
3. When the UiRRM bit is set to 1, set the CKDIR bit in the UiMR register to 1 (external clock) and also disable the RTS function.

Figure 17.8 U0BRG to U4BRG Registers, U0C1 to U4C1 Registers

External Interrupt Source Select Register



Symbol	Address	After Reset
IFSR	031Fh	00h

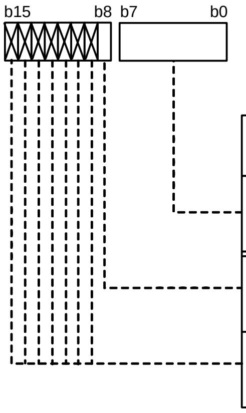
Bit Symbol	Bit Name	Function	RW
IFSR0	$\overline{\text{INT0}}$ interrupt polarity select bit[1]	0: One edge 1: Both edges	RW
IFSR1	$\overline{\text{INT1}}$ interrupt polarity select bit[1]	0: One edge 1: Both edges	RW
IFSR2	$\overline{\text{INT2}}$ interrupt polarity select bit[1]	0: One edge 1: Both edges	RW
IFSR3	$\overline{\text{INT3}}$ interrupt polarity select bit[1]	0: One edge 1: Both edges	RW
IFSR4	$\overline{\text{INT4}}$ interrupt polarity select bit[1]	0: One edge 1: Both edges	RW
IFSR5	$\overline{\text{INT5}}$ interrupt polarity select bit[1]	0: One edge 1: Both edges	RW
IFSR6	UART0, UART3 interrupt source select bit	0: UART3 bus conflict, start condition detection, stop condition detection 1: UART0 bus conflict, start condition detection, stop condition detection	RW
IFSR7	UART1, UART4 interrupt source select bit	0: UART4 bus conflict, start condition detection, stop condition detection 1: UART1 bus conflict, start condition detection, stop condition detection	RW

NOTE:

1. Set the IFSRi bit (i = 0 to 5) to 0 to select a level-sensitive triggering. When selecting both edges, set the POL bit in the corresponding INTiIC register to 0 (falling edge).

Figure 17.9 IFSR Register

UARTi Transmit Buffer Register [1] (i = 0 to 4)

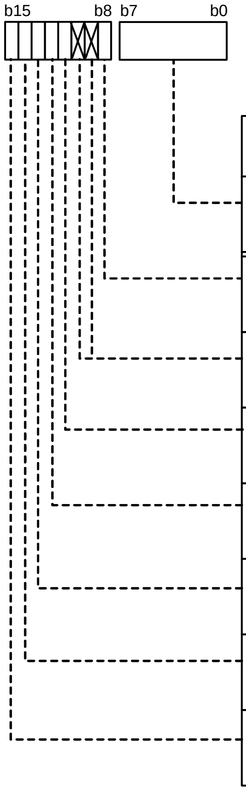


Symbol	Address	After Reset
U0TB to U2TB	036Bh - 036Ah, 02EBh - 02EAh, 033Bh - 033Ah	Undefined
U3TB, U4TB	032Bh - 032Ah, 02FBh - 02FAh	Undefined

Bit Symbol	Function	RW
– (b7-b0)	Transmit data (D7 to D0)	WO
– (b8)	Transmit data (D8)	WO
– (b15-b9)	Unimplemented. Write 0. Read as undefined value.	–

NOTE:
1. Read-modify-write instructions cannot be used to set the UiTB register. Refer to **Usage Notes** for details.

UARTi Receive Buffer Register (i = 0 to 4)



Symbol	Address	After Reset
U0RB to U2RB	036Fh - 036Eh, 02EFh - 02EEh, 033Fh - 033Eh	Undefined
U3RB, U4RB	032Fh - 032Eh, 02FFh - 02FEh	Undefined

Bit Symbol	Bit Name	Function	RW
– (b7-b0)	–	Received data (D7 to D0)	RO
– (b8)	–	Received data (D8)	RO
– (b10-b9)	Unimplemented. Write 0. Read as undefined value.		–
ABT	Arbitration lost detect flag[1]	0: Not detected (won) 1: Detected (lost)	RW
OER	Overrun error flag[2]	0: No overrun error 1: Overrun error	RO
FER	Framing error flag[2, 3]	0: No framing error 1: Framing error	RO
PER	Parity error flag[2, 3]	0: No parity error 1: Parity error	RO
SUM	Error sum flag[2, 3]	0 No error occurred 1: Error occurred	RO

NOTES:
1. Only a 0 can be written to the ABT bit.
2. When bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled) or the RE bit in the UiC1 register is set to 0 (receive operation disabled), bits OER, FER, PER and SUM become 0.
When all of bits OER, FER and PER become 0, the SUM bit also becomes 0.
Bits FER and PER become 0 by reading the low-order byte in the UiRB register.
3. Bits FER, PER and SUM are disabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous mode) or 010b (I^2C mode). A read from these bits returns undefined value.

Figure 17.10 U0TB to U4TB Registers, U0RB to U4RB Registers

17.1.1 Clock Synchronous Mode

Full-duplex clock synchronous serial communications are allowed in this mode. CTS/RTS function can be used for transmit and receive control.

Table 17.1 lists specifications of clock synchronous mode. Table 17.2 lists pin settings. Figure 17.11 shows register settings. Figure 17.12 shows an example of a transmit and receive operation when an internal clock is selected. Figure 17.13 shows an example of a receive operation when an external clock is selected.

Table 17.1 Clock Synchronous Mode Specifications

Item	Specification
Data format	Data length: 8 bits long
Serial clock	Internal clock or external clock can be selected by the CKDIR bit in the UiMR register (i = 0 to 4)
Baud rate	• When the CKDIR bit is set to 0 (internal clock): fj / (2 (m + 1) fj = f1, f8, f2n(1) m: setting value of the UiBRG register (00h to FFh) • When the CKDIR bit is set to 1 (external clock): clock input to the CLKi pin
Transmit/receive control	Selectable among the CTS function, RTS function, or CTS/RTS function disabled
Transmit and receive start condition	Internal clock is selected: • Set the TE bit in the UiC1 register to 1 (transmit operation enabled) • The TI bit in the UiC1 register is 0 (data in the UiTB register) • Set the RE bit in the UiC1 register to 1 (receive operation enabled) • "L" signal is applied to the $\overline{\text{CTSi}}$ pin when the CTS function is used External clock is selected(2): • Set the TE bit to 1 • The TI bit is 0 • Set the RE bit to 1 • The RI bit in the UiC1 register is 0 when the RTS function is used When above 4 conditions are met, $\overline{\text{RTSi}}$ pin outputs "L" If transmit-only operation is performed, the RE bit setting is not required in both cases.
Interrupt request generation timing	Transmit interrupt (The UiIRS bit in the UiC1 register selects one of the following): • The UiIRS bit is set to 0 (no data in the UiTB register): when data is transferred from the UiTB register to the UARTi transmit shift register (transmit operation started) • The UiIRS bit is set to 1 (transmit operation completed): when data transmit operation from the UARTi transmit shift register is completed Receive interrupt: • When data is transferred from the UARTi receive shift register to the UiRB register (receive operation completed)
Error detection	• Overrun error(3) Overrun error occurs when the 7th bit of the next data is received before reading the UiRB register
Selectable function	• CLK polarity Transmit data output timing and receive data input timing can be selected • LSB first or MSB first Data is transmitted and received from either bit 0 or bit 7 • Serial data logic inverse Transmit and receive data are logically inverted • Continuous receive mode The TI bit becomes 0 by reading the UiRB register

NOTES:

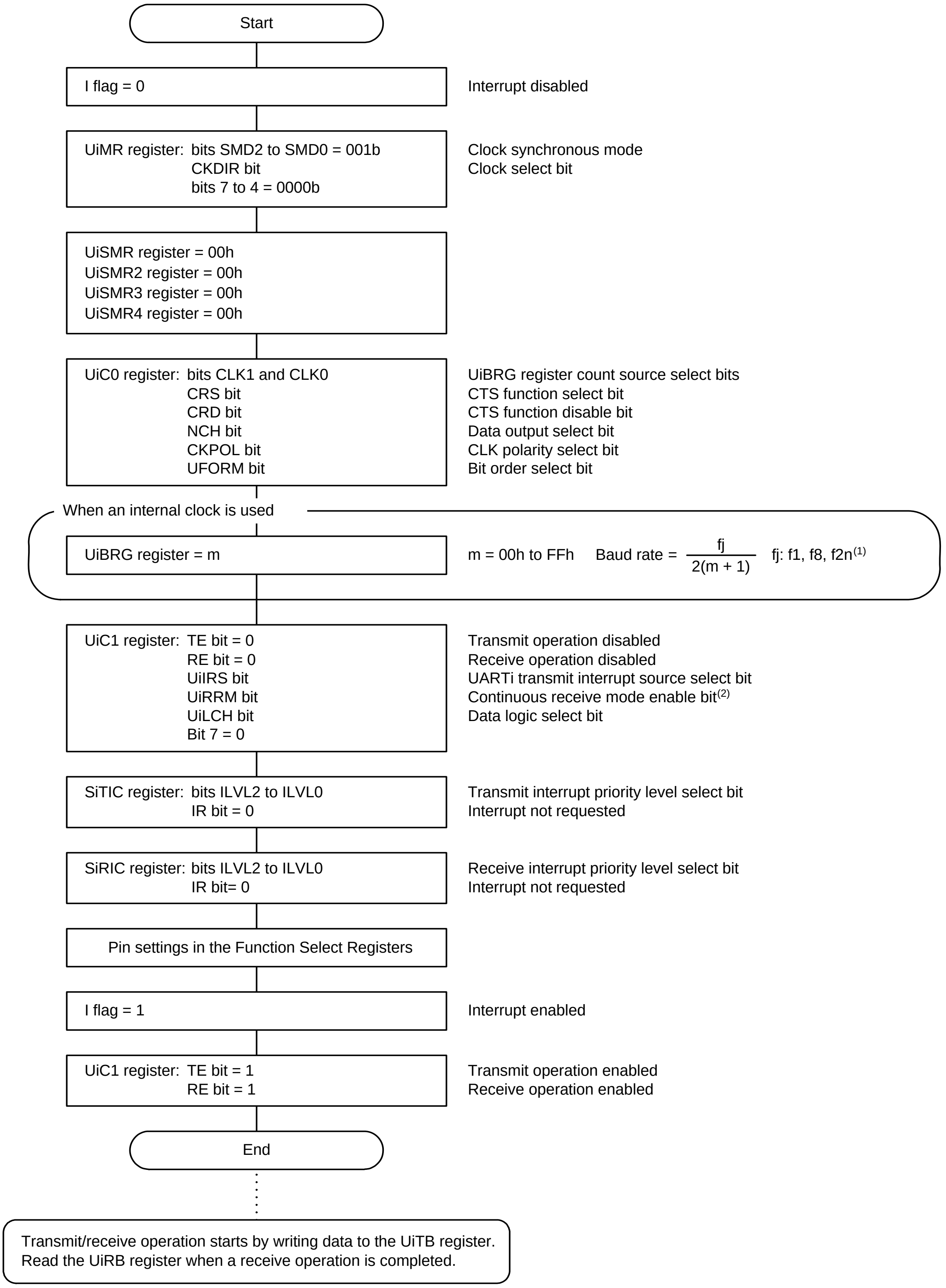
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. If an external clock is selected, ensure that an "H" signal is applied to the CLKi pin when the CKPOL bit in the UiC0 register is set to 0, and that an "L" signal is applied when the CKPOL bit is set to 1.
3. If an overrun error occurs, a read from the UiRB register returns undefined values. The IR bit in the SiRIC register remains unchanged as 0 (interrupt not requested).

Table 17.2 Pin Settings in Clock Synchronous Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC Register	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers(1)(2)
P6_0	$\overline{\text{CTS0}}$ input	PD6_0 = 0	–	–	PS0_0 = 0
	$\overline{\text{RTS0}}$ output	–	–	PSL0_0 = 0	PS0_0 = 1
P6_1	CLK0 input	PD6_1 = 0	–	–	PS0_1 = 0
	CLK0 output	–	–	PSL0_1 = 0	PS0_1 = 1
P6_2	RXD0 input	PD6_2 = 0	–	–	PS0_2 = 0
P6_3	TXD0 output(4)	–	–	PSL0_3 = 0	PS0_3 = 1
P6_4	$\overline{\text{CTS1}}$ input	PD6_4 = 0	–	–	PS0_4 = 0
	$\overline{\text{RTS1}}$ output	–	–	PSL0_4 = 0	PS0_4 = 1
P6_5	CLK1 input	PD6_5 = 0	–	–	PS0_5 = 0
	CLK1 output	–	–	PSL0_5 = 0	PS0_5 = 1
P6_6	RXD1 input	PD6_6 = 0	–	–	PS0_6 = 0
P6_7	TXD1 output(4)	–	–	PSL0_7 = 0	PS0_7 = 1
P7_0(3)	TXD2 output(4)	–	PSC_0 = 0	PSL1_0 = 0	PS1_0 = 1
P7_1	RXD2 input	PD7_1 = 0	–	–	PS1_1 = 0
P7_2	CLK2 input	PD7_2 = 0	–	–	PS1_2 = 0
	CLK2 output	–	PSC_2 = 0	PSL1_2 = 0	PS1_2 = 1
P7_3	$\overline{\text{CTS2}}$ input	PD7_3 = 0	–	–	PS1_3 = 0
	$\overline{\text{RTS2}}$ output	–	PSC_3 = 0	PSL1_3 = 0	PS1_3 = 1
P9_0	CLK3 input	PD9_0 = 0	–	–	PS3_0 = 0
	CLK3 output	–	–	PSL3_0 = 0	PS3_0 = 1
P9_1	RXD3 input	PD9_1 = 0	–	–	PS3_1 = 0
P9_2	TXD3 output(4)	–	–	PSL3_2 = 0	PS3_2 = 1
P9_3	$\overline{\text{CTS3}}$ input	PD9_3 = 0	–	PSL3_3 = 0	PS3_3 = 0
	$\overline{\text{RTS3}}$ output	–	–	–	PS3_3 = 1
P9_4	$\overline{\text{CTS4}}$ input	PD9_4 = 0	–	PSL3_4 = 0	PS3_4 = 0
	$\overline{\text{RTS4}}$ output	–	–	–	PS3_4 = 1
P9_5	CLK4 input	PD9_5 = 0	–	PSL3_5 = 0	PS3_5 = 0
	CLK4 output	–	–	–	PS3_5 = 1
P9_6	TXD4 output(4)	–	–	–	PS3_6 = 1
P9_7	RXD4 input	PD9_7 = 0	–	–	PS3_7 = 0

NOTES:

1. Set registers PS0, PS1, and PS3 after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
3. P7_0 is an N-channel open drain output port.
4. After UARTi (i = 0 to 4) operating mode is selected in the UiMR register and the pin function is set in the Function Select Registers, the TXDi pin outputs an “H” signal until a transmit operation starts (the TXDi pin is in a high-impedance state when N-channel open drain output is selected).

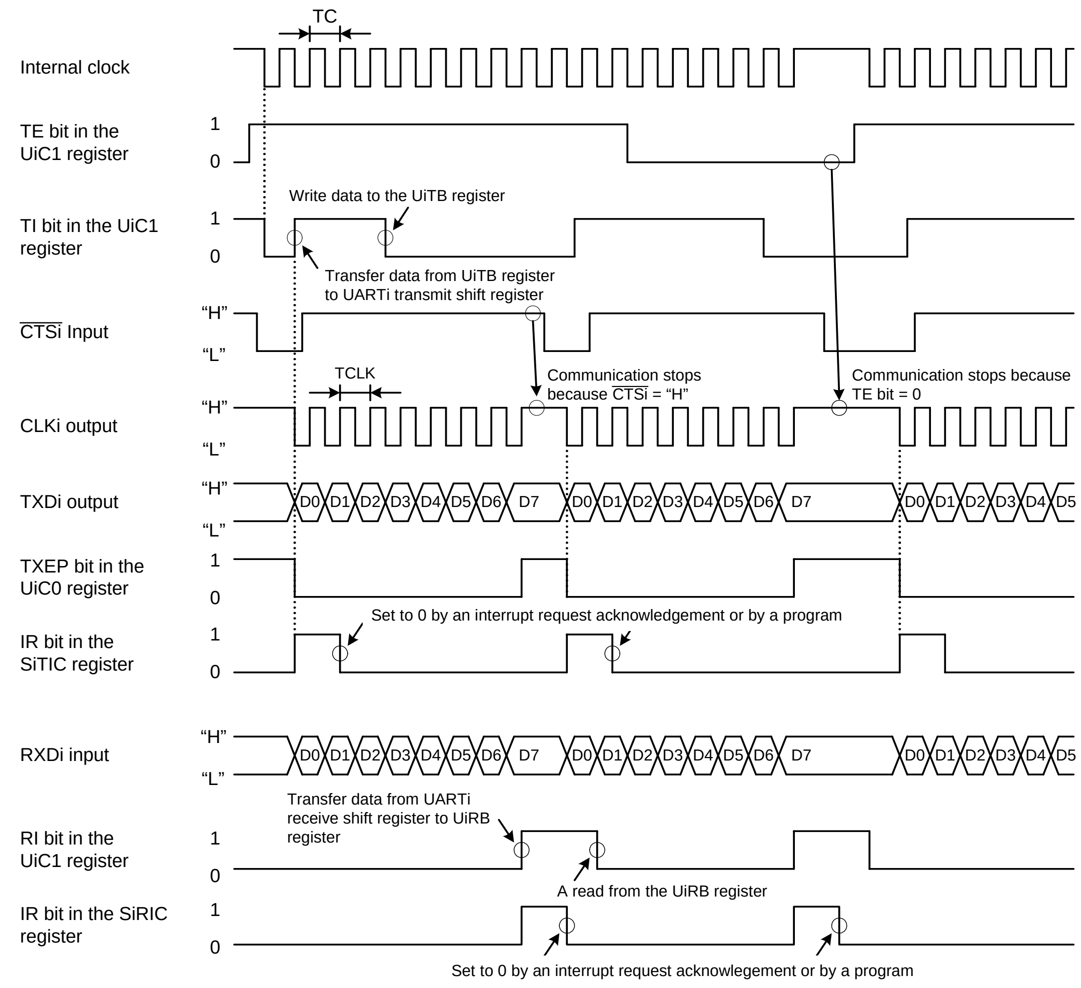

Start
I flag = 0
Interrupt disabled
UiMR register: bits SMD2 to SMD0 = 001b
CKDIR bit
bits 7 to 4 = 0000b
Clock synchronous mode
Clock select bit
UiSMR register = 00h
UiSMR2 register = 00h
UiSMR3 register = 00h
UiSMR4 register = 00h
UiC0 register: bits CLK1 and CLK0
CRS bit
CRD bit
NCH bit
CKPOL bit
UFORM bit
UiBRG register count source select bits
CTS function select bit
CTS function disable bit
Data output select bit
CLK polarity select bit
Bit order select bit
When an internal clock is used
UiBRG register = m
m = 00h to FFh
Baud rate = fj / 2(m + 1)
fj: f1, f8, f2n(1)
UiC1 register: TE bit = 0
RE bit = 0
UiIRS bit
UiRRM bit
UiLCH bit
Bit 7 = 0
Transmit operation disabled
Receive operation disabled
UARTi transmit interrupt source select bit
Continuous receive mode enable bit(2)
Data logic select bit
SiTIC register: bits ILVL2 to ILVL0
IR bit = 0
Transmit interrupt priority level select bit
Interrupt not requested
SiRIC register: bits ILVL2 to ILVL0
IR bit= 0
Receive interrupt priority level select bit
Interrupt not requested
Pin settings in the Function Select Registers
I flag = 1
Interrupt enabled
UiC1 register: TE bit = 1
RE bit = 1
Transmit operation enabled
Receive operation enabled
End
Transmit/receive operation starts by writing data to the UiTB register.
Read the UiRB register when a receive operation is completed.

i = 0 to 4

NOTES:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. The UiRRM bit can be set to 1 (continuous receive mode enabled), only when the CKDIR bit in the UiMR register is set to 1 (external clock) and RTS function is disabled.

Figure 17.11 Register Settings in Clock Synchronous Mode


TC
Internal clock
TE bit in the UiC1 register
1
0
Write data to the UiTB register
TI bit in the UiC1 register
1
0
Transfer data from UiTB register to UARTi transmit shift register
CTSi Input
"H"
"L"
TCLK
Communication stops because CTSi = "H"
Communication stops because TE bit = 0
CLKi output
"H"
"L"
TXDi output
"H"
"L"
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5
TXEP bit in the UiC0 register
1
0
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the SiTIC register
1
0
RXDi input
"H"
"L"
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5
Transfer data from UARTi receive shift register to UiRB register
RI bit in the UiC1 register
1
0
A read from the UiRB register
IR bit in the SiRIC register
1
0
Set to 0 by an interrupt request acknowlegement or by a program

i = 0 to 4

$$TC = TCLK = \frac{2(m + 1)}{fj}$$

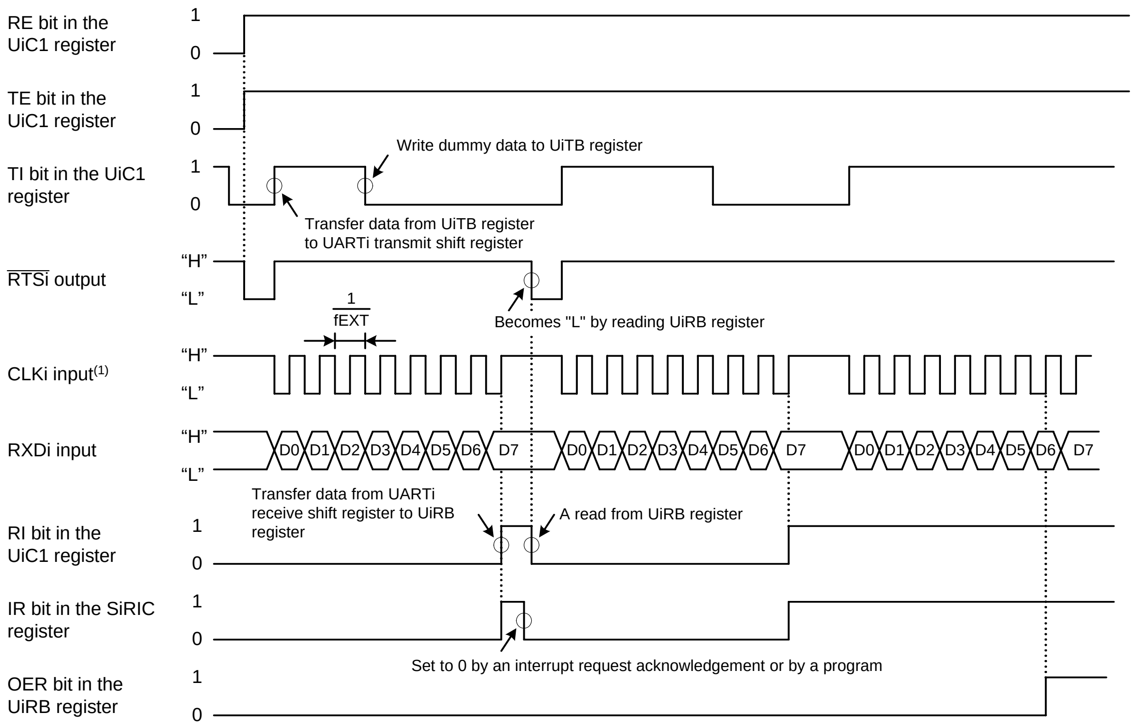
fj = f1, f8, f2n(1)

m = Setting value of the UiBRG register (00h to FFh)

The above applies under the following conditions:
- UiMR register: CKDIR bit = 0 (internal clock)
- UiC0 register: CRD bit in the = 0 and CRS bit = 0 (CTS function used)
  CKPOL bit = 0 (transmit data output at the falling edge of the serial clock)
- UiC1 register: UiIRS bit = 0 (Transmit interrupt request is generated when no data in the UiTB register)

NOTE:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).

Figure 17.12 Transmit and Receive Operations when Internal Clock is Selected


RE bit in the UiC1 register
1
0
TE bit in the UiC1 register
1
0
TI bit in the UiC1 register
1
0
Write dummy data to UiTB register
Transfer data from UiTB register to UARTi transmit shift register
RTSi output
"H"
"L"
1/fEXT
Becomes "L" by reading UiRB register
CLKi input(1)
"H"
"L"
RXDi input
"H"
"L"
D0 D1 D2 D3 D4 D5 D6 D7
Transfer data from UARTi receive shift register to UiRB register
A read from UiRB register
RI bit in the UiC1 register
1
0
IR bit in the SiRIC register
1
0
Set to 0 by an interrupt request acknowledgement or by a program
OER bit in the UiRB register
1
0

i = 0 to 4

fEXT = external clock frequency

The above applies under the following conditions:
- UiMR register: CKDIR bit = 1 (external clock)
- UiC0 reigster: CRD bit = 1 (CTS function disabled)
  CKPOL bit = 0 (receive data input at the rising edge of the serial clock)

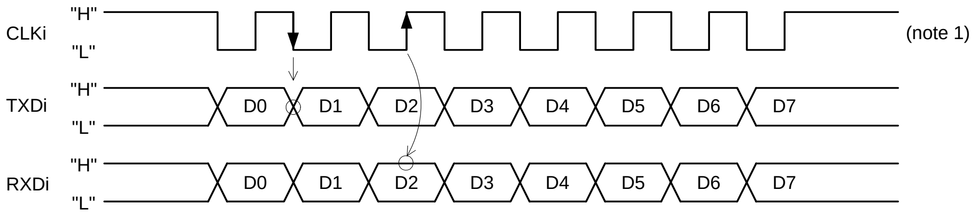
NOTE:
1. Satisfy the following conditions, while the CLKi pin input is "H" before the data receive operation.
   - UiC1 register: TE bit = 1 (transmit operation enabled)
     RE bit = 1 (receive operation enabled)
   - Write dummy data to the UiTB register

Figure 17.13 Receive Operations when External Clock is Selected

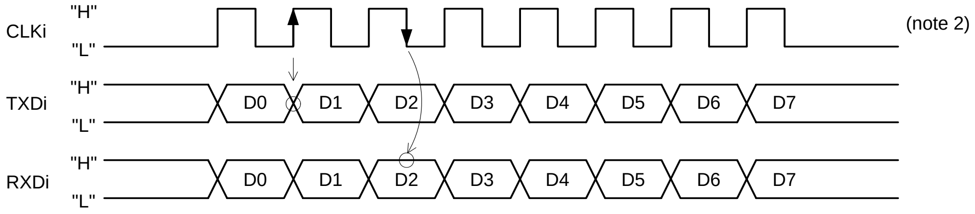
17.1.1.1 CLK Polarity

As shown in figure 17.14, the CKPOL bit in the UiC0 register (i = 0 to 4) determines the polarity of the serial clock.

(1) When the CKPOL bit in the UiC0 register (i = 0 to 4) is set to 0 (transmit data output at the falling edge and receive data input at the rising edge of the serial clock)



(2) When the CKPOL bit is set to 1 (transmit data output at the rising edge and receive data input at the falling edge of the serial clock)



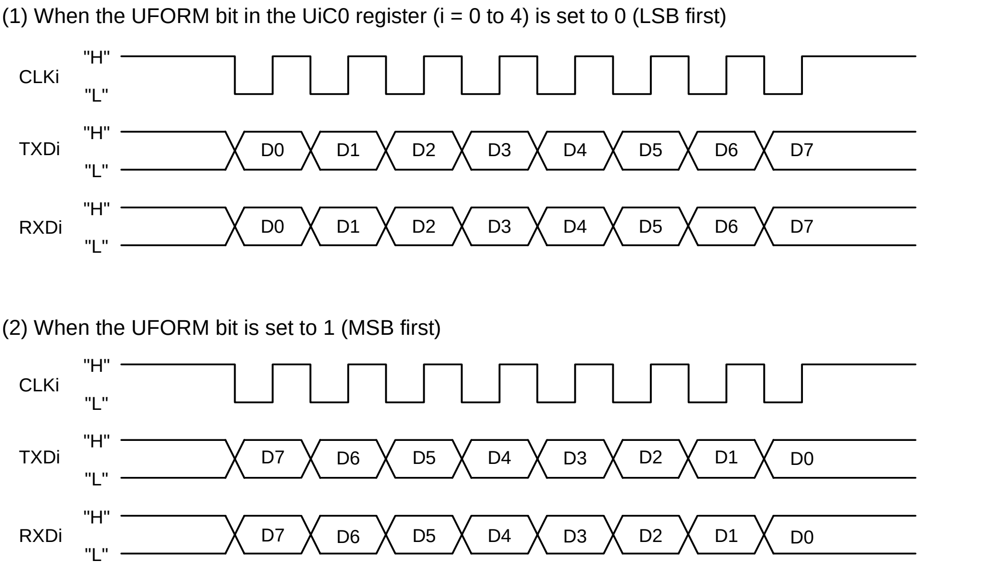
The above applies under the following conditions:
- UFORM bit in the UiC0 register is set to 0 (LSB first)
- UiLCH bit in the UiC1 register is set to 0 (not inverted).

NOTES:
1. The CLKi pin output level is "H" when no transmit and receive operation is in progress.
2. The CLKi pin output level is "L" when no transmit and receive operation is in progress.

Figure 17.14 Serial Clock Polarity

17.1.1.2 LSB First or MSB First

As shown in figure 17.15, the UFORM bit in the UiC0 register (i = 0 to 4) determines a bit order.


(1) When the UFORM bit in the UiC0 register (i = 0 to 4) is set to 0 (LSB first)
CLKi
"H"
"L"
TXDi
D0
D1
D2
D3
D4
D5
D6
D7
RXDi
(2) When the UFORM bit is set to 1 (MSB first)
D7
D6
D5
D4
D3
D2
D1
D0

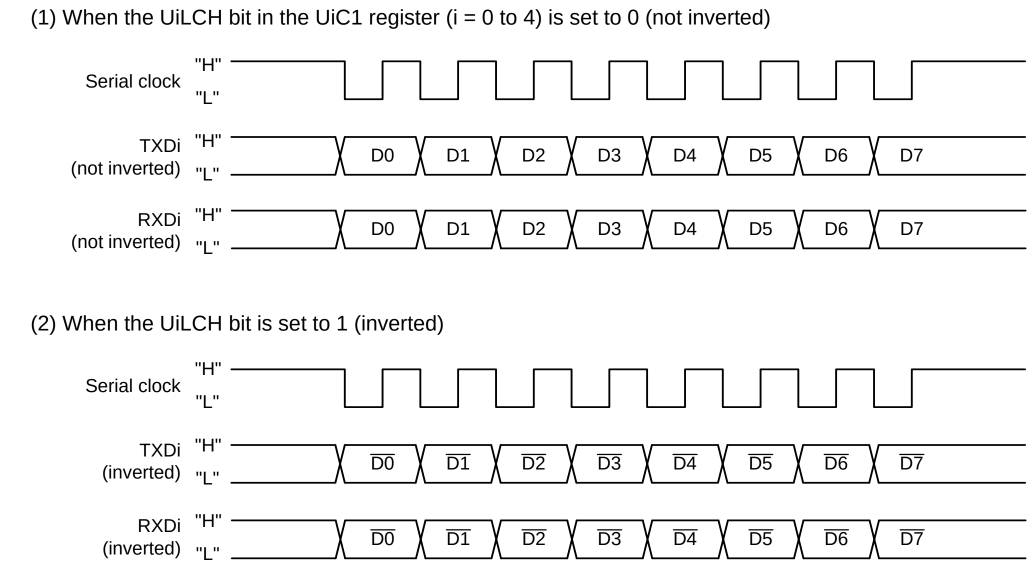
The above applies under the following conditions:

- CKPOL bit in the UiC0 register is set to 0 (transmit data is output at the falling edge and received data is input at the rising edge)
- UiLCH bit in the UiC1 register is set to 0 (not inverted).

Figure 17.15 Bit Order (8-Bit Data Length)

17.1.1.3 Serial Data Logic Inverse

When the UiLCH bit in the UiC1 register is set to 1 (inverted), data logic written in the UiTB register is inverted for transmit operation. A read from the UiRB register returns the inverted logic of receive data. Figure 17.16 shows an example of serial data logic inverse operation.


(1) When the UiLCH bit in the UiC1 register (i = 0 to 4) is set to 0 (not inverted)
Serial clock
"H"
"L"
TXDi
(not inverted)
D0 D1 D2 D3 D4 D5 D6 D7
RXDi
(not inverted)
D0 D1 D2 D3 D4 D5 D6 D7
(2) When the UiLCH bit is set to 1 (inverted)
Serial clock
TXDi
(inverted)
$\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$
RXDi
(inverted)
$\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$

The above applies under the following conditions:
- CKPOL bit in the UiC0 register is set to 0 (transmit data is output at the falling edge and received data is input at the rising edge)
- UFORM bit in the UiC0 register is set to 0 (LSB first).

Figure 17.16 Serial Data Logic Inverse

17.1.1.4 Continuous Receive Mode

Continuous receive mode can be used when all of the following conditions are met.

- External clock is selected (the CKDIR bit in the UiMR register (i = 0 to 4) is set to 1)
- RTS function is disabled ($\overline{\text{RTSi}}$ pin is not selected in the Function Select Register)

When the UiRRM bit in the UiC1 register is set to 1 (continuous receive mode enabled), the TI bit in the UiC1 register becomes 0 (data in the UiTB register) by reading the UiRB register. Do not set dummy data to the UiTB register if the UiRRM bit is set to 1.

17.1.1.5 CTS/RTS Function

- CTS Function

  Transmit and receive operation is controlled by using the input signal to the $\overline{\text{CTSi}}$ pin (i = 0 to 4). To use the CTS function, select the I/O port in the Function Select Register, set the CRD bit in the UiC0 register to 0 (CTS function enabled), and the CRS bit to 0 (CTS function selected).

  With the CTS function used, the transmit and receive operation starts when all the following conditions are met and an “L” signal is applied to the $\overline{\text{CTSi}}$ pin.

  -The TE bit in the UiC1 register is set to 1 (transmit operation enabled)
  -The TI bit in the UiC1 register is 0 (data in the UiTB register)
  -The RE bit in the UiC1 register is set to 1 (receive operation enabled)
  (If transmit-only operation is performed, the RE bit setting is not required)

  When a high-level (“H”) signal is applied to the $\overline{\text{CTSi}}$ pin during transmitting and receiving, the transmit and receive operation is disabled after the transmit and receive operation in progress is completed.

- RTS Function

  The MCU can inform the external device that it is ready for a transmit and receive operation by using the output signal from the $\overline{\text{RTSi}}$ pin. To use the RTS function, select the $\overline{\text{RTSi}}$ pin in the Function Select Register.

  With the RTS function used, the $\overline{\text{RTSi}}$ pin outputs an “L” signal when all the following conditions are met, and outputs an “H” when the serial clock is input to the CLKi pin.

  -The RI bit in the UiC1 register is 0 (no data in the UiRB register)
  -The TE bit is set to 1 (transmit operation enabled)
  -The RE bit is set to 1 (receive operation enabled)
  (If transmit-only operation is performed, the RE bit setting is not required)
  -The TI bit is 0 (data in the UiTB register)

17.1.1.6 Procedure When Communication is Aborted or Communication Error is Occurred

Follow the procedure below when a communication is aborted or a communication error is occurred in clock synchronous mode.

(1) Set the TE bit in the UiC1 register (i = 0 to 4) to 0 (transmit operation disabled) and the RE bit to 0 (receive operation 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 mode).
(4) Set the TE bit to 1 (transmit operation enabled) and the RE bit to 1 (receive operation enabled).

17.1.2 Clock Asynchronous (UART) Mode

Full-duplex asynchronous serial communications are allowed in this mode. Table 17.3 lists specifications of UART mode. Table 17.4 lists pin settings. Figure 17.17 shows register settings. Figure 17.18 shows an example of a transmit operation. Figure 17.19 shows an example of a receive operation.

Table 17.3 UART Mode Specifications

Item	Specification
Data format	• Data length: selectable among 7 bits, 8 bits, or 9 bits long • Start bit: 1 bit long • Parity bit: selectable among odd, even, or none • Stop bit: selectable from 1 bit or 2 bits long
Baud rate	fj / (16 (m + 1)) fj = f1, f8, f2n[1], fEXT m: setting value of the UiBRG register (00h to FFh) fEXT: clock input to the CLKi pin when the CKDIR bit in the UiMR register is set to 1 (external clock)
Transmit/receive control	Selectable among CTS function, RTS function or CTS/RTS function disabled
Transmit start condition	To start transmit operation, all of the following must be met: • Set the TE bit in the UiC1 register to 1 (transmit operation enabled) • The TI bit in the UiC1 register is 0 (data in the UiTB register) • Apply a low-level ("L") signal to the CTSi pin when the CTS function is selected
Receive start condition	To start receive operation, all of the following must be met: • Set the RE bit in the UiC1 register to 1 (receive operation enabled) • The RI bit is 1 (no data in UiRB register) when RTS function is used. When the above two conditions are met, the RTSi pin output an "L" signal. • The start bit is detected
Interrupt request generation timing	Transmit interrupt (The UiIRS bit in the UiC1 register selects one of the following): • The UiIRS bit is set to 0 (no data in the UiTB register): when data is transferred from the UiTB register to the UARTi transmit shift register (transmit operation started) • The UiIRS bit is set to 1 (transmit operation completed): when the final stop bit is output from the UARTi transmit shift register Receive interrupt: • When data is transferred from the UARTi receive shift register to the UiRB register (receive operation completed)
Error detection	• Overrun error[2] Overrun error occurs when the preceding bit of the final stop bit of the next data (the first stop bit when selecting 2 stop bits) is received before reading the UiRB register • Framing error Framing error occurs when the number of the stop bits set by the STPS bit in the UiMR register is not detected • Parity error Parity error occurs when parity is enabled and the received data does not have the correct even or odd parity set by the PRY bit in the UiMR register. • Error sum flag Error sum flag is set to 1 when any of overrun, framing, and parity errors occurs
Selectable function	• LSB first or MSB first Data is transmitted or received from either bit 0 or bit 7 • Serial data logic inverse Transmit and receive data are logically inverted. The start bit and stop bit are not inverted • TXD and RXD I/O polarity inverse The level output from the TXD pin and the level applied to the RXD pin are inverted. All the data including the start bit and stop bit are inverted.

NOTES:

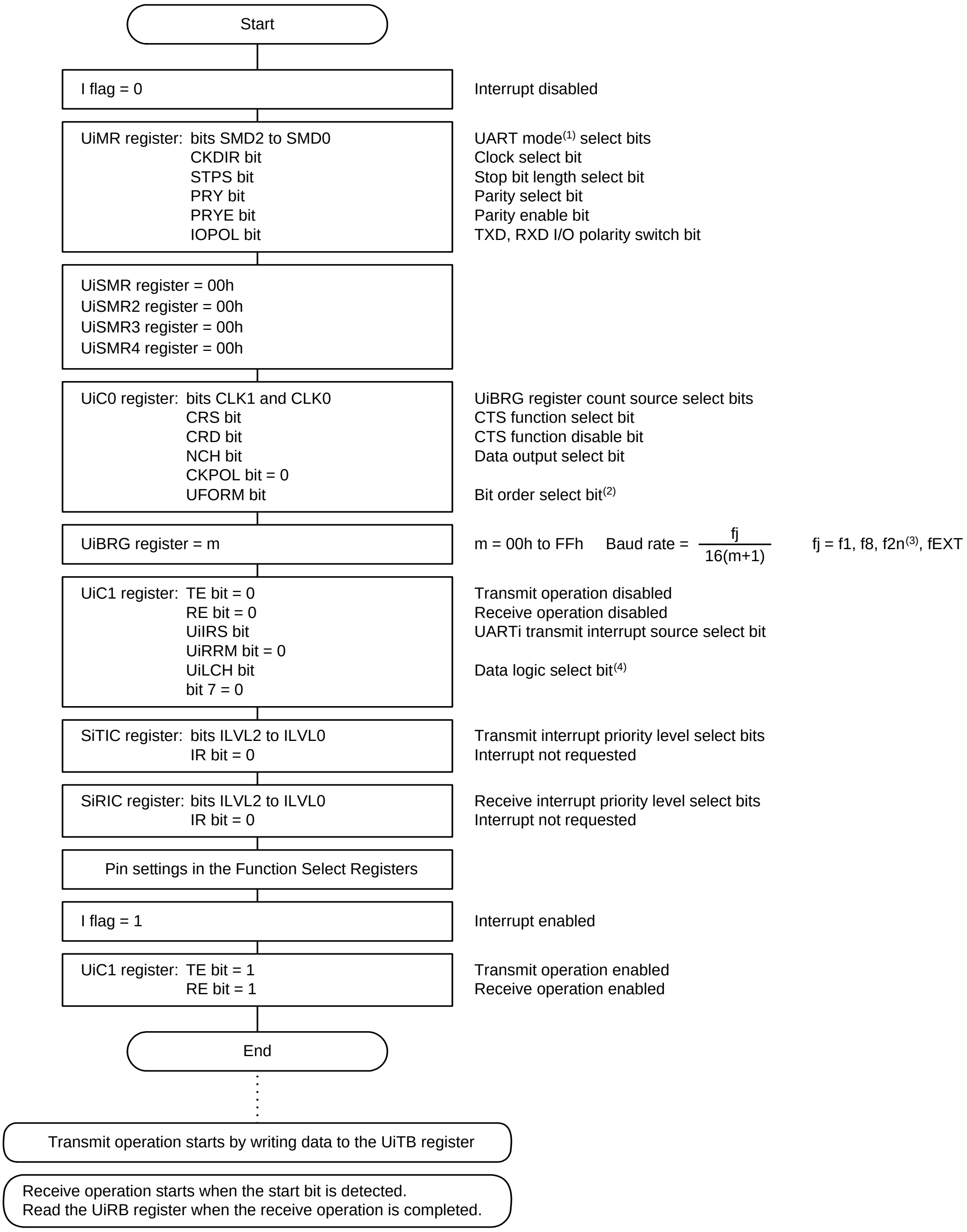
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. If an overrun error occurs, a read from the UiRB register returns undefined values. The IR bit in the SiRIC register remains unchanged as 0 (interrupt not requested).

Table 17.4 Pin Settings in UART Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers[2]	PSC Register	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers[1][2]
P6_0	$\overline{\text{CTS0}}$ input	PD6_0 = 0	–	–	PS0_0 = 0
	$\overline{\text{RTS0}}$ output	–	–	PSL0_0 = 0	PS0_0 = 1
P6_1	CLK0 input	PD6_1 = 0	–	–	PS0_1 = 0
P6_2	RXD0 input	PD6_2 = 0	–	–	PS0_2 = 0
P6_3	TXD0 output[4]	–	–	PSL0_3 = 0	PS0_3 = 1
P6_4	$\overline{\text{CTS1}}$ input	PD6_4 = 0	–	–	PS0_4 = 0
	$\overline{\text{RTS1}}$ output	–	–	PSL0_4 = 0	PS0_4 = 1
P6_5	CLK1 input	PD6_5 = 0	–	–	PS0_5 = 0
P6_6	RXD1 input	PD6_6 = 0	–	–	PS0_6 = 0
P6_7	TXD1 output[4]	–	–	PSL0_7 = 0	PS0_7 = 1
P7_0[3]	TXD2 output[4]	–	PSC_0 = 0	PSL1_0 = 0	PS1_0 = 1
P7_1	RXD2 input	PD7_1 = 0	–	–	PS1_1 = 0
P7_2	CLK2 input	PD7_2 = 0	–	–	PS1_2 = 0
P7_3	$\overline{\text{CTS2}}$ input	PD7_3 = 0	–	–	PS1_3 = 0
	$\overline{\text{RTS2}}$ output	–	PSC_3 = 0	PSL1_3 = 0	PS1_3 = 1
P9_0	CLK3 input	PD9_0 = 0	–	–	PS3_0 = 0
P9_1	RXD3 input	PD9_1 = 0	–	–	PS3_1 = 0
P9_2	TXD3 output[4]	–	–	PSL3_2 = 0	PS3_2 = 1
P9_3	$\overline{\text{CTS3}}$ input	PD9_3 = 0	–	PSL3_3 = 0	PS3_3 = 0
	$\overline{\text{RTS3}}$ output	–	–	–	PS3_3 = 1
P9_4	$\overline{\text{CTS4}}$ input	PD9_4 = 0	–	PSL3_4 = 0	PS3_4 = 0
	$\overline{\text{RTS4}}$ output	–	–	–	PS3_4 = 1
P9_5	CLK4 input	PD9_5 = 0	–	PSL3_5 = 0	PS3_5 = 0
P9_6	TXD4 output[4]	–	–	–	PS3_6 = 1
P9_7	RXD4 input	PD9_7 = 0	–	–	PS3_7 = 0

NOTES:

1. Set registers PS0, PS1, and PS3 after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
3. P7_0 is an N-channel open drain output port.
4. After UARTi (i = 0 to 4) operating mode is selected in the UiMR register and the pin function is set in the Function Select Registers, the TXDi pin outputs an “H” signal until a transmit operation starts (the TXDi pin is in a high-impedance state when N-channel open drain output is selected).

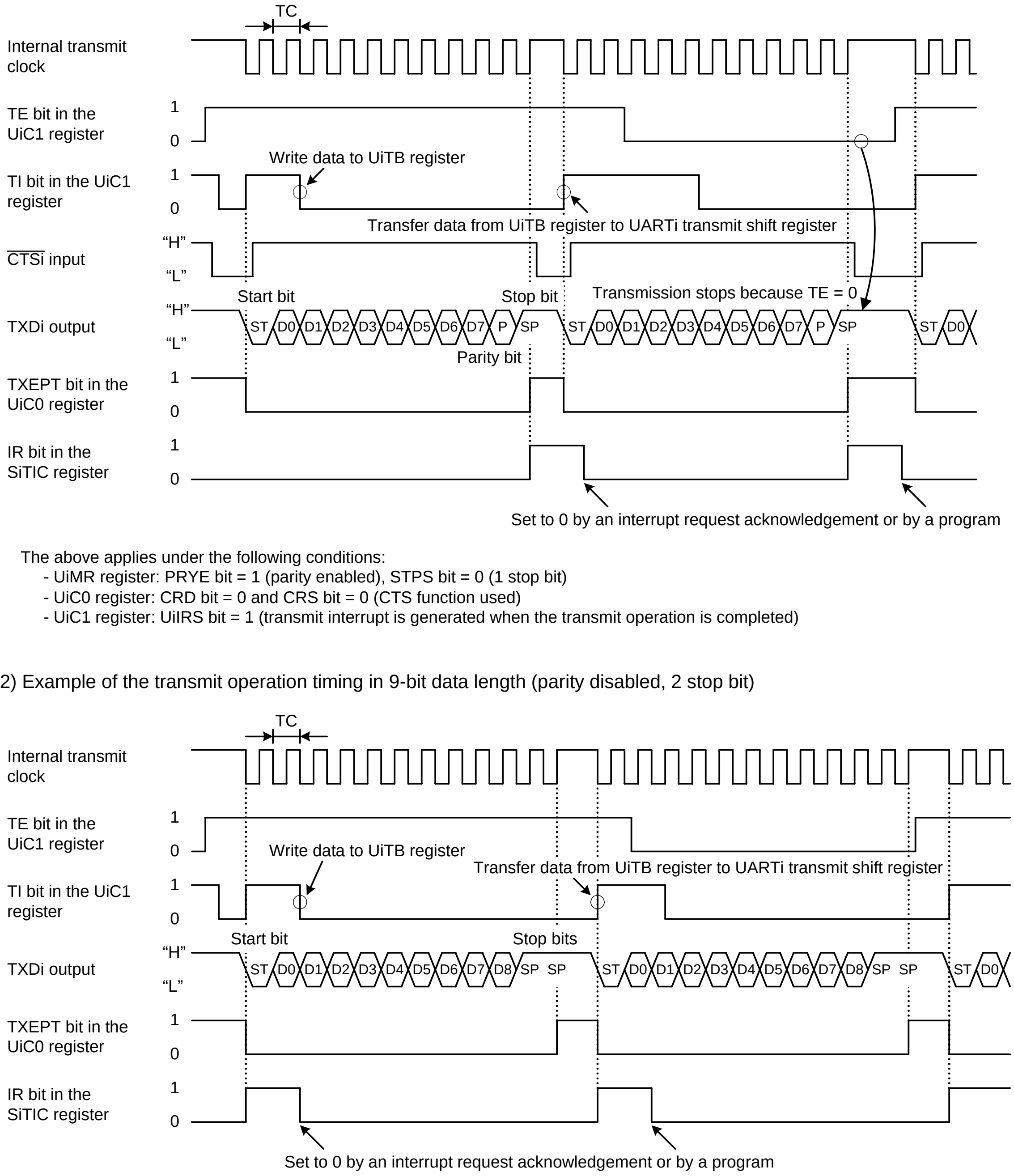

Start
I flag = 0
Interrupt disabled
UiMR register: bits SMD2 to SMD0
CKDIR bit
STPS bit
PRY bit
PRYE bit
IOPOL bit
UART mode(1) select bits
Clock select bit
Stop bit length select bit
Parity select bit
Parity enable bit
TXD, RXD I/O polarity switch bit
UiSMR register = 00h
UiSMR2 register = 00h
UiSMR3 register = 00h
UiSMR4 register = 00h
UiC0 register: bits CLK1 and CLK0
CRS bit
CRD bit
NCH bit
CKPOL bit = 0
UFORM bit
UiBRG register count source select bits
CTS function select bit
CTS function disable bit
Data output select bit
Bit order select bit(2)
UiBRG register = m
m = 00h to FFh
Baud rate = fj / 16(m+1)
fj = f1, f8, f2n(3), fEXT
UiC1 register: TE bit = 0
RE bit = 0
UiIRS bit
UiRRM bit = 0
UiLCH bit
bit 7 = 0
Transmit operation disabled
Receive operation disabled
UARTi transmit interrupt source select bit
Data logic select bit(4)
SiTIC register: bits ILVL2 to ILVL0
IR bit = 0
Transmit interrupt priority level select bits
Interrupt not requested
SiRIC register: bits ILVL2 to ILVL0
IR bit = 0
Receive interrupt priority level select bits
Interrupt not requested
Pin settings in the Function Select Registers
I flag = 1
Interrupt enabled
UiC1 register: TE bit = 1
RE bit = 1
Transmit operation enabled
Receive operation enabled
End
Transmit operation starts by writing data to the UiTB register
Receive operation starts when the start bit is detected.
Read the UiRB register when the receive operation is completed.

i = 0 to 4
fEXT: clock input to the CLKi pin when the external clock is selected

NOTES:
1. Set bits SMD2 to SMD0 to the following: 100b (7 bits long), 101b (8 bits long), or 110b (9 bits long).
2. A bit order can be selected when 8-bit data length is selected. Set to 0 when 7-bit or 9-bit data length is selected.
3. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
4. Whether data logic is inverted or not can be selected when 7-bit or 8-bit data length is selected. Set to 0 when 9-bit data length is selected.

Figure 17.17 Register Settings in UART Mode

(1) Example of the transmit operation timing in 8-bit data length (parity enabled, 1 stop bit)


TC
Internal transmit clock
TE bit in the UiC1 register
1
0
TI bit in the UiC1 register
1
0
Write data to UiTB register
Transfer data from UiTB register to UARTi transmit shift register
CTSi input
"H"
"L"
Start bit
Stop bit
Transmission stops because TE = 0
TXDi output
"H"
"L"
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
ST D0
Parity bit
TXEPT bit in the UiC0 register
1
0
IR bit in the SiTIC register
1
0
Set to 0 by an interrupt request acknowledgement or by a program

The above applies under the following conditions:
- UiMR register: PRYE bit = 1 (parity enabled), STPS bit = 0 (1 stop bit)
- UiC0 register: CRD bit = 0 and CRS bit = 0 (CTS function used)
- UiC1 register: UiIRS bit = 1 (transmit interrupt is generated when the transmit operation is completed)

(2) Example of the transmit operation timing in 9-bit data length (parity disabled, 2 stop bit)

TC
Internal transmit clock
TE bit in the UiC1 register
1
0
Write data to UiTB register
Transfer data from UiTB register to UARTi transmit shift register
TI bit in the UiC1 register
1
0
Start bit
Stop bits
TXDi output
"H"
"L"
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP
ST D0
TXEPT bit in the UiC0 register
1
0
IR bit in the SiTIC register
1
0
Set to 0 by an interrupt request acknowledgement or by a program

The above applies under the following conditions:
- UiMR register: PRYE bit = 0 (parity disabled), STPS bit = 1 (2 stop bits)
- UiC0 register: CRD bit = 1 (CTS function disabled)
- UiC1 register: UiIRS bit = 0 (transmit interrupt is generated when no data in the UiTB register)

$$TC = \frac{16(m+1)}{fj}$$

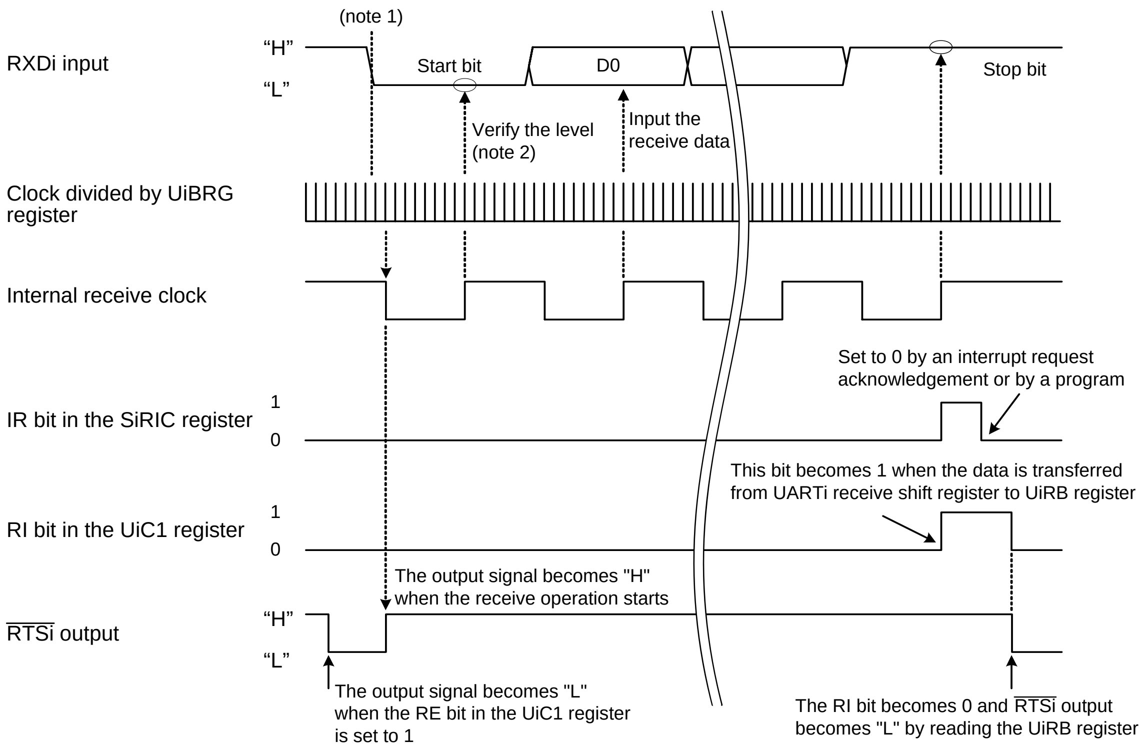
fj: f1, f8, f2n[1], fEXT
fEXT: clock input to the CLKi pin when the external clock is selected
m: setting value of the UiBRG register (00h to FFh)

i = 0 to 4

NOTE:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).

Figure 17.18 Transmit Operation in UART Mode

Example of the receive operation timing (1 stop bit)


(note 1)
RXDi input
"H"
"L"
Start bit
D0
Stop bit
Verify the level (note 2)
Input the receive data
Clock divided by UiBRG register
Internal receive clock
Set to 0 by an interrupt request acknowledgement or by a program
IR bit in the SiRIC register
1
0
This bit becomes 1 when the data is transferred from UARTi receive shift register to UiRB register
RI bit in the UiC1 register
1
0
The output signal becomes "H" when the receive operation starts
$\overline{\text{RTSi}}$ output
"H"
"L"
The output signal becomes "L" when the RE bit in the UiC1 register is set to 1
The RI bit becomes 0 and $\overline{\text{RTSi}}$ output becomes "L" by reading the UiRB register

i = 0 to 4

The above applies under the following conditions:

- UiMR register: STPS bit = 0 (1 stop bit)
- UiC0 register: CRS bit = 1 (CTS function not used)

NOTES:

1. RXDi input is sampled using the clock divided by the setting value of the UiBRG register. The internal receive clock is generated after detecting the falling edge of the start bit, and then the receive operation starts.
2. When "L" is detected, the receive operation continues. When "H" is detected, the receive operation is cancelled. When the receive operatin is cancelled, the $\overline{\text{RTSi}}$ output becomes "L".

Figure 17.19 Receive Operation in UART Mode

17.1.2.1 Baud Rate

In UART mode, the baud rate is the frequency of the clock divided by the setting value of the UiBRG register (i = 0 to 4) and again divided by 16. Table 17.5 lists an example of baud rate setting.

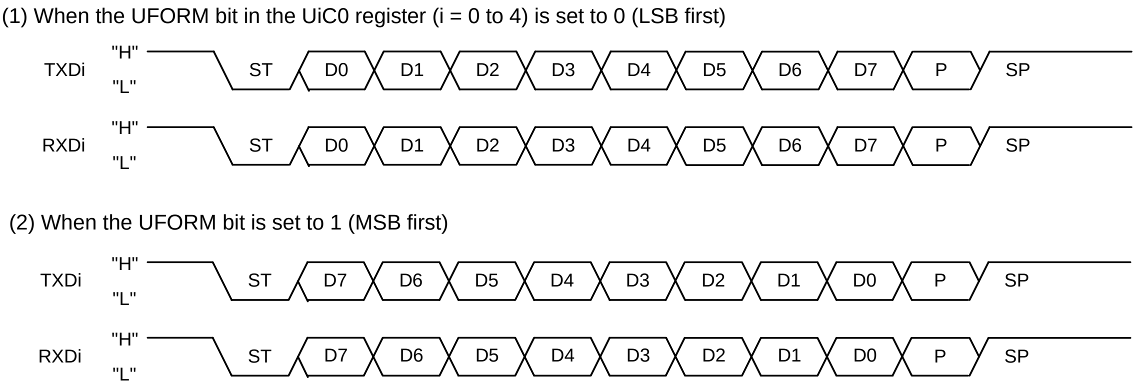
$$\text{Actual baud rate} = \frac{\text{UiBRG register count source}}{16 \times (\text{UiBRG register setting value} + 1)}$$

Table 17.5 Baud Rate

Target Baud Rate (bps)	UiBRG Count Source	Peripheral Clock: 16MHz		Peripheral Clock: 24MHz		Peripheral Clock: 32MHz	
		UiBRG Setting Value: n	Actual Baud Rate (bps)	UiBRG Setting Value: n	Actual Baud Rate (bps)	UiBRG Setting Value: n	Actual Baud Rate (bps)
1200	f8	103(67h)	1202	155(9Bh)	1202	207(CFh)	1202
2400	f8	51(33h)	2404	77(4Dh)	2404	103(67h)	2404
4800	f8	25(19h)	4808	38(26h)	4808	51(33h)	4808
9600	f1	103(67h)	9615	155(9Bh)	9615	207(CFh)	9615
14400	f1	68(44h)	14493	103(67h)	14423	138(8Ah)	14388
19200	f1	51(33h)	19231	77(4Dh)	19231	103(67h)	19231
28800	f1	34(22h)	28571	51(33h)	28846	68(44h)	28986
31250	f1	31(1Fh)	31250	47(2Fh)	31250	63(3Fh)	31250
38400	f1	25(19h)	38462	38(26h)	38462	51(33h)	38462
51200	f1	19(13h)	50000	28(1Ch)	51724	38(26h)	51282

17.1.2.2 LSB First or MSB First

As shown in Figure 17.20, the UFORM bit in the UiC0 register (i = 0 to 4) determines a bit order. This function can be used when data length is 8 bits long.



(1) When the UFORM bit in the UiC0 register (i = 0 to 4) is set to 0 (LSB first)

(2) When the UFORM bit is set to 1 (MSB first)

TXDi
"H"
"L"
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
RXDi
"H"
"L"
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
TXDi
"H"
"L"
ST D7 D6 D5 D4 D3 D2 D1 D0 P SP
RXDi
"H"
"L"
ST D7 D6 D5 D4 D3 D2 D1 D0 P SP

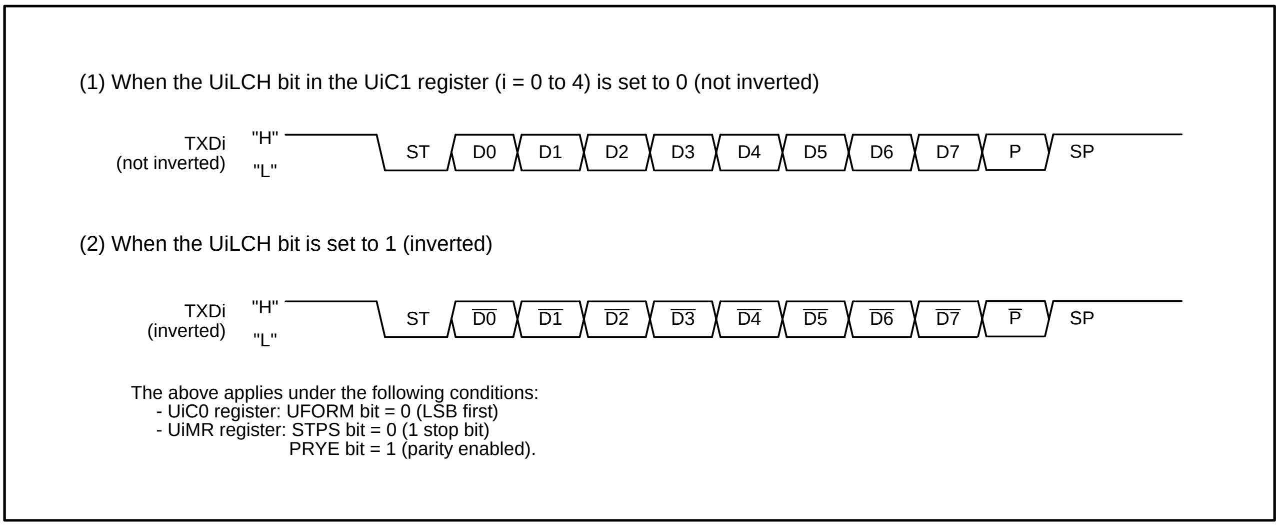
The above applies under the following conditions:
- UiC0 register: CKPOL bit = 0 (transmit data output at the falling edge and receive data input at the rising edge of the serial clock)
- UiC1 register: UiLCH bit = 0 (not inverted) and the UiLCH bit in the UiC1 register is set to 0 (not inverted).

ST: Start bit
P: Parity bit
SP: Stop bit

Figure 17.20 Bit Order

17.1.2.3 Serial Data Logic Inverse

When the UiLCH bit in the UiC1 register is set to 1 (inverted), data logic written in the UiTB register is inverted for transmit operation. A read from the UiRB register returns the inverted logic of receive data. This function can be used when data length is 7 bits or 8 bits long. Figure 17.21 shows an example of serial data logic inverse operation.

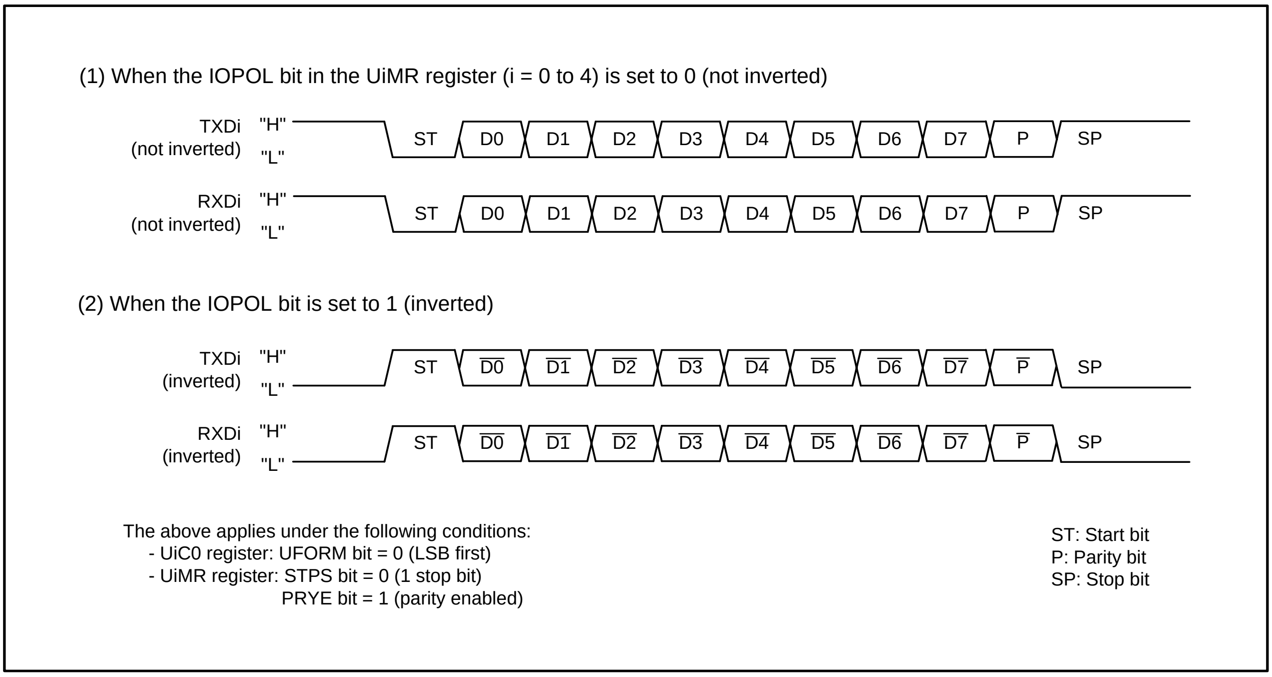


The above applies under the following conditions:
- UiC0 register: UFORM bit = 0 (LSB first)
- UiMR register: STPS bit = 0 (1 stop bit)
PRYE bit = 1 (parity enabled).

Figure 17.21 Serial Data Logic Inverse

17.1.2.4 TXD and RXD I/O Polarity Inverse

The level output from the TXD pin and the level applied to the RXD pin are inverted with this function. When the IOPOL bit in the UiMR register (i = 0 to 4) is set to 1 (inverted), all the input/output data levels, including the start bit, stop bit and parity bit, are inverted. Figure 17.22 shows TXD and RXD I/O polarity inverse.



The above applies under the following conditions:
- UiC0 register: UFORM bit = 0 (LSB first)
- UiMR register: STPS bit = 0 (1 stop bit)
PRYE bit = 1 (parity enabled)

ST: Start bit
P: Parity bit
SP: Stop bit

Figure 17.22 TXD and RXD I/O Polarity Inverse

17.1.2.5 CTS/RTS Function

- CTS Function

  Transmit operation is controlled by using the input signal to the $\overline{\text{CTSi}}$ pin . To use the CTS function, select the I/O port in the Function Select Register, set the CRD bit in the UiC0 register to 0 (CTS function enabled), and the CRS bit to 0 (CTS function selected).

  With the CTS function used, the transmit operation starts when all the following conditions are met and an "L" signal is applied to the $\overline{\text{CTSi}}$ pin (i = 0 to 4).

  -The TE bit in the UiC1 register is set to 1 (transmit operation enabled)

  -The TI bit in the UiC1 register is 0 (data in the UiTB register)

  When a high-level ("H") signal is applied to the $\overline{\text{CTSi}}$ pin during transmitting, the transmit operation is disabled after the transmit operation in progress is completed.

- RTS Function

  The MCU can inform the external device that it is ready for a receive operation by using the output signal from the $\overline{\text{RTSi}}$ pin. To use the RTS function, select the $\overline{\text{RTSi}}$ pin in the Function Select Register.

  With the RTS function used, the $\overline{\text{RTSi}}$ pin outputs an "L" signal when all the following conditions are met, and outputs an "H" when the start bit is detected.

  -The RI bit in the UiC1 register is 0 (no data in the UiRB register)

  -The RE bit is set to 1 (receive operation enabled)

17.1.2.6 Procedure When Communication is Aborted or Communication Error is Occurred

Follow the procedure below when a communication is aborted or a communication error is occurred in UART mode.

(1) Set the TE bit in the UiC1 register (i = 0 to 4) to 0 (transmit operation disabled) and the RE bit to 0 (receive operation 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 100b (UART mode, 7-bit data length), 101b (UART mode, 8-bit data length), or 110b (UART mode, 9-bit data length).

(4) Set the TE bit to 1 (transmit operation enabled) and the RE bit to 1 (receive operation enabled).

17.1.3 Special Mode 1 (I^2C Mode)

In I^2C mode, the simplified I^2C helps to communicate with external devices.

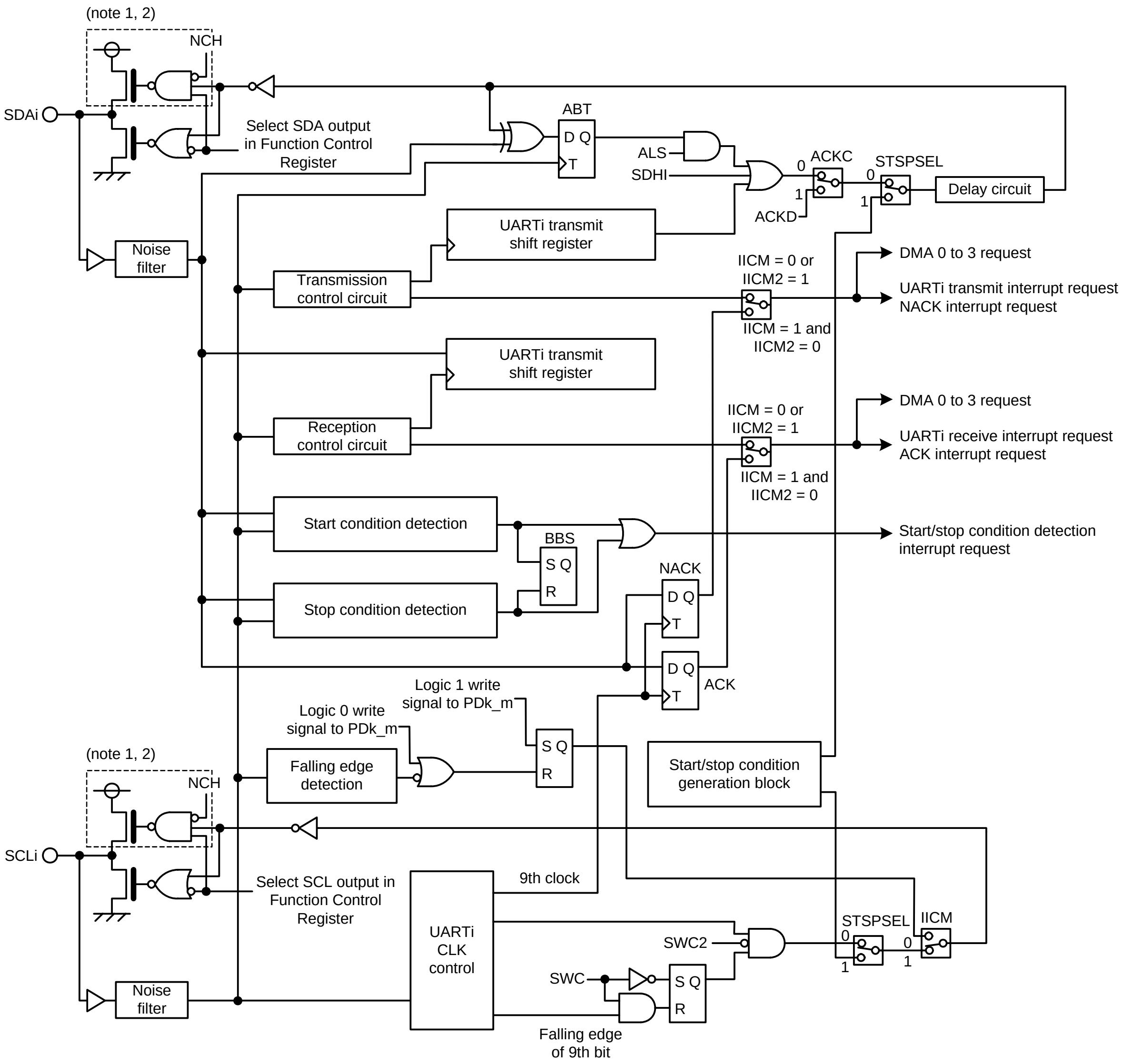
Table 17.6 lists specifications of I^2C mode. Tables 17.7 and 17.8 list register settings. Tables 17.9 and 17.10 list individual functions in I^2C mode. Table 17.11 lists pin settings. Figure 17.23 shows a block diagram of I^2C mode. Figure 17.24 shows a transfer timing to the UiRB register (i = 0 to 4) and interrupt timing.

Table 17.6 I^2C Mode Specifications

Item	Specification
Data format	• Data length: 8 bits long
Baud rate	• In master mode When the CKDIR bit in the UiMR register (i = 0 to 4) is set to 0 (internal clock): fj / (2 (m + 1)) fj = f1, f8, f2n[1] m: setting value of the UiBRG register (00h to FFh) • In slave mode When the CKDIR bit is set to 1 (external clock): input from the SCLi pin
Transmit start condition	To start transmit operation, all of the following must be met[2]: • Set the TE bit in the UiC1 register to 1 (transmit operation enabled) • The TI bit in the UiC1 register is 0 (data in the UiTB register)
Receive start condition	To start receive operation, all of the following must be met[2]: • Set the TE bit to 1 (transmit operation enabled) • The TI bit is 0 (data in the UiTB register) • Set the RE bit in the UiC1 register to 1 (receive operation enabled)
Interrupt request generation timing	• Start condition detection • Stop condition detection • ACK (Acknowledge) detection • NACK (Not-Acknowledge) detection
Error detection	• Overrun error[3] Overrun error occurs when the 8th bit of the next data is received before reading the UiRB register
Selectable function	• Arbitration lost detect timing Update timing of the ABT bit in the UiRB register (i = 0 to 4) can be selected. • SDAi digital delay No digital delay or 2 to 8 cycle delay of the UiBRG count source can be selected. • Clock phase setting Clock delay or no clock delay can be selected.

NOTES:

1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. If an external clock is selected, satisfy the conditions while an “H” signal is applied to the SCLi pin.
3. If an overrun error occurs, a read from the UiRB register returns undefined values.


(note 1, 2)
NCH
SDAi
Select SDA output in Function Control Register
ABT
D Q
T
ALS
SDHI
ACKC
0
1
ACKD
STSPSEL
0
1
Delay circuit
Noise filter
UARTi transmit shift register
Transmission control circuit
IICM = 0 or IICM2 = 1
IICM = 1 and IICM2 = 0
DMA 0 to 3 request
UARTi transmit interrupt request
NACK interrupt request
UARTi transmit shift register
Reception control circuit
DMA 0 to 3 request
UARTi receive interrupt request
ACK interrupt request
Start condition detection
BBS
S Q
R
Start/stop condition detection interrupt request
Stop condition detection
NACK
D Q
T
ACK
Logic 1 write signal to PDk_m
Logic 0 write signal to PDk_m
Falling edge detection
S Q
R
Start/stop condition generation block
(note 1, 2)
NCH
SCLi
Select SCL output in Function Control Register
UARTi CLK control
9th clock
STSPSEL
IICM
SWC2
SWC
S Q
R
Falling edge of 9th bit
Noise filter

i = 0 to 4

IICM, BBS: bits in the UiSMR register
IICM2, SWC, ALS, SWC2, SDHI: bits in the UiSMR2 register
STSPSEL, ACKD, ACKC: bits in the UiSMR4 register
NCH: bit in the UiC0 register
ABT: UiRB register
PDk_m: bit in the Port Pk Direction Register corresponding to the SCLi pin

NOTES:

1. P7_0 and P7_1 do not have the dotted rectangular portion of the circuit. The absolute maximum rating of the input voltage for P7_0 and P7_1 is from -0.3 V to 6.0 V.
2. P6_2, P6_3, P6_6, P6_7, P9_1, P9_2, P9_6, and P9_7 are used with turning off the P channel of the CMOS port all the time. The absolute maximum rating of the input voltage for these ports is from - 0.3 V to VCC1 + 0.3 V.

Figure 17.23 I^{2}C Mode Block Diagram

Table 17.7 Register Settings in I^2C Mode (1/2)

<table>
<tr><th rowspan="2">Register</th><th rowspan="2">Bit</th><th colspan="2">Setting Value</th></tr>
<tr><th>Master</th><th>Slave</th></tr>
<tr><td rowspan="3">UiMR</td><td>SMD2 to SMD0</td><td colspan="2">Set to 010b</td></tr>
<tr><td>CKDIR</td><td>Set to 0</td><td>Set to 1</td></tr>
<tr><td>IOPOL</td><td colspan="2">Set to 0</td></tr>
<tr><td rowspan="4">UiSMR</td><td>IICM</td><td colspan="2">Set to 1</td></tr>
<tr><td>ABC</td><td>Select an arbitration lost detect timing</td><td>Disabled</td></tr>
<tr><td>BBS</td><td colspan="2">Bus busy flag</td></tr>
<tr><td>7 to 3</td><td colspan="2">Set to 00000b</td></tr>
<tr><td rowspan="8">UiSMR2</td><td>IICM2</td><td colspan="2">See **Tables 17.9 and 17.10 Functions in I^2C Mode**</td></tr>
<tr><td>CSC</td><td>Set to 1 to enable clock synchronization</td><td>Set to 0</td></tr>
<tr><td>SWC</td><td colspan="2">Set to 1 to hold an “L” signal output from SCLi at the falling edge of the ninth bit of the serial clock</td></tr>
<tr><td>ALS</td><td>Set to 1 to abort an SDAi output when detecting the arbitration lost</td><td>Set to 0</td></tr>
<tr><td>STC</td><td>Set to 0</td><td>Set to 1 to initialize UARTi by detecting the start condition</td></tr>
<tr><td>SWC2</td><td colspan="2">Set to 1 to forcibly make a signal output from SCL an “L”</td></tr>
<tr><td>SDHI</td><td colspan="2">Set to 1 to disable SDA output</td></tr>
<tr><td>SU1HIM</td><td colspan="2">Set to 0</td></tr>
<tr><td rowspan="4">UiSMR3</td><td>SSE</td><td colspan="2">Set to 0</td></tr>
<tr><td>CKPH</td><td colspan="2">See **Tables 17.9 and 17.10 Functions in I^2C Mode**</td></tr>
<tr><td>DINC, NODC, ERR</td><td colspan="2">Set to 0</td></tr>
<tr><td>DL2 to DL0</td><td colspan="2">Set SDAi digital delay value</td></tr>
<tr><td rowspan="8">UiSMR4</td><td>STAREQ</td><td>Set to 1 to generate the start condition</td><td rowspan="4">Set to 0</td></tr>
<tr><td>RSTAREQ</td><td>Set to 1 to generate the restart condition</td></tr>
<tr><td>STPREQ</td><td>Set to 1 to generate the stop condition</td></tr>
<tr><td>STSPSEL</td><td>Set to 1 when using a condition generation function</td></tr>
<tr><td>ACKD</td><td colspan="2">Select ACK or NACK</td></tr>
<tr><td>ACKC</td><td colspan="2">Set to 1 to output ACK data</td></tr>
<tr><td>SCLHI</td><td>Set to 1 to enable SCL output stop when detecting the stop condition</td><td>Set to 0</td></tr>
<tr><td>SWC9</td><td>Set to 0</td><td>Set to 1 to hold an “L” signal output from SCLi at the falling edge of the ninth bit of the serial clock</td></tr>
</table>

i = 0 to 4

Table 17.8 Register Settings in I2C Mode (2/2)

Register	Bit	Setting Value	
		Master	Slave
UiC0	CLK1, CLK0	Select the count source of the UiBRG register	Disabled
	CRS	Disabled because the CRD bit is set to 1	
	TXEPT	Transmit shift register empty flag	
	CRD, NCH	Set to 1	
	CKPOL	Set to 0	
	UFORM	Set to 1	
UiC1	TE	Set to 1 to enable transmit operation	
	TI	UiTB register empty flag	
	RE	Set to 1 to enable receive operation	
	RI	Receive operation complete flag	
	UiLCH, UiERE	Set to 0	
UiBRG	7 to 0	Set baud rate	Disabled
IFSR	IFSR7, IFSR6	Select the UARTi interrupt source	
UiTB	7 to 0	Set transmit data	
UiRB	7 to 0	Receive data can be read	
	8	ACK or NACK is received	
	ABT	Arbitration lost detect flag	Disabled
	OER	Overrun error flag	

i = 0 to 4

As shown in Tables 17.9 and 17.10, I^{2}C mode is entered when bits SMD2 to SMD0 in the UiMR register are set to 010b (I^{2}C mode) and the IICM bit in the UiSMR register to 1 (I^{2}C mode). Because an SDAi transmit output passes through a delay circuit, output signal from the SDAi pin changes after the SCLi pin level becomes low (“L”) and the “L” output stabilizes.

Table 17.9 Functions in I^{2}C Mode (1/2)

Function	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)
Interrupt source for numbers 39 to 41[1] (See **Figure 17.24**)	Start condition or stop condition detection (See Table 17.12 STSPSEL Bit Function)			
Interrupt source for numbers 17, 19, 33, 35, 37[1] (See **Figure 17.24**)	No acknowledgement detection (NACKi) - at the rising edge of 9th bit of SCLi		UARTi transmit operation - at the rising edge of 9th bit of SCLi	UARTi transmit operation - at the next falling edge after the 9th bit of SCLi
Interrupt source for numbers 18, 20, 34, 36, 38[1] (See **Figure 17.24**)	Acknowledgement detection (ACKi) - at the rising edge of 9th bit of SCLi		UARTi receive operation - at the falling edge of 9th bit of SCLi	
Data transfer timing from the UART receive shift register to the UiRB register	At rising edge of 9th bit of SCLi		Falling edge of 9th bit of SCLi	Falling edge and rising edge of 9th bit of SCLi
UARTi transmit output delay	Delay			
Functions of P6_3, P6_7, P7_0, P9_2, P9_6	SDAi input and output			
Functions of P6_2, P6_6, P7_1, P9_1, P9_7	SCLi input and output			
Noise filter width	200 ns			

i = 0 to 4

NOTE:

1. Use the following procedures to change an interrupt source.
   (a) Disable an interrupt of the corresponding interrupt number.
   (b) Change an interrupt source.
   (c) Set the IR bit of a corresponding interrupt number to 0 (interrupt not requested).
   (d) Set bits ILVL2 to ILVL0 of the corresponding interrupt number.

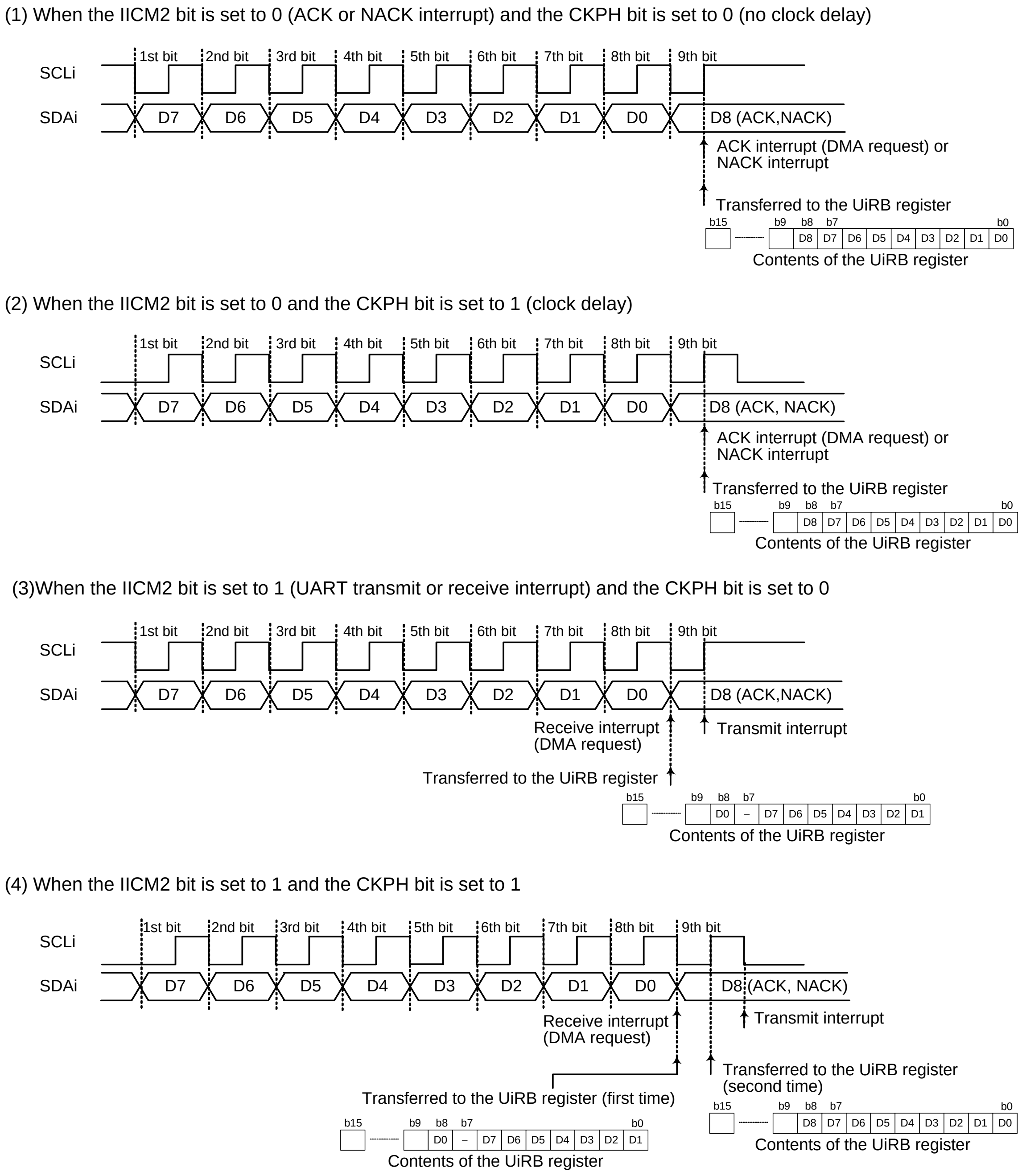
Table 17.10 Functions in I^{2}C Mode (2/2)

Function	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)
Reading RXDi, SCLi pin levels	Can be read regardless of the corresponding port direction bit			
Default value of TXDi, SDAi output	Value set in the port register before entering I^{2}C mode(1)			
SCLi default and end values	H	L	H	L
DMA source (See **Figure 17.24**)	Acknowledgement detection (ACKi)		UARTi receive operation - at the falling edge of 9th bit of SCLi	
Storing receive data	1st to 8th bit of the receive data are stored into bits 7 to 0 in the UiRB register		1st to 7th bits of the receive 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 bits 7 to 0 in the UiRB register(2)
Reading receive data	The value in the UiRB register is read as it is			Bits 6 to 0 in the UiRB register are read as bits 7 to 1. Bit 8 in the UiRB register is read as bit 0(3)

i = 0 to 4

NOTES:

1. Set default value of the SDAi output while bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled).
2. Second data transfer to the UiRB register (at the rising edge of the ninth bit of SCLi).
3. First data transfer to the UiRB register (at the falling edge of the ninth bit of SCLi).


(1) When the IICM2 bit is set to 0 (ACK or NACK interrupt) and the CKPH bit is set to 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 (DMA request) or NACK interrupt
Transferred to the UiRB register
b15
b9
b8
b7
b0
D8 D7 D6 D5 D4 D3 D2 D1 D0
Contents of the UiRB register
(2) When the IICM2 bit is set to 0 and the CKPH bit is set to 1 (clock delay)
D8 (ACK, NACK)
ACK interrupt (DMA request) or NACK interrupt
Transferred to the UiRB register
Contents of the UiRB register
(3)When the IICM2 bit is set to 1 (UART transmit or receive interrupt) and the CKPH bit is set to 0
D8 (ACK,NACK)
Receive interrupt (DMA request)
Transmit interrupt
Transferred to the UiRB register
D0 – D7 D6 D5 D4 D3 D2 D1
Contents of the UiRB register
(4) When the IICM2 bit is set to 1 and the CKPH bit is set to 1
D8 (ACK, NACK)
Receive interrupt (DMA request)
Transmit interrupt
Transferred to the UiRB register (first time)
Transferred to the UiRB register (second time)
D0 – D7 D6 D5 D4 D3 D2 D1
D8 D7 D6 D5 D4 D3 D2 D1 D0
Contents of the UiRB register

i = 0 to 4

The above applies when the CKDIR bit in UiMR register is set to 1 (external clock).

Figure 17.24 Transfer Timing to the UiRB Register and Interrupt Timing

Table 17.11 Pin Settings in I²C Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC Register	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers(1)(2)
P6_2	SCL0 output	–	–	PSL0_2 = 0	PS0_2 = 1
	SCL0 input	PD6_2 = 0	–	–	PS0_2 = 0
P6_3	SDA0 output	–	–	PSL0_3 = 0	PS0_3 = 1
	SDA0 input	PD6_3 = 0	–	–	PS0_3 = 0
P6_6	SCL1 output	–	–	PSL0_6 = 0	PS0_6 = 1
	SCL1 input	PD6_6 = 0	–	–	PS0_6 = 0
P6_7	SDA1 output	–	–	PSL0_7 = 0	PS0_7 = 1
	SDA1 input	PD6_7 = 0	–	–	PS0_7 = 0
P7_0(3)	SDA2 output	–	PSC_0 = 0	PSL1_0 = 0	PS1_0 = 1
	SDA2 input	PD7_0 = 0	–	–	PS1_0 = 0
P7_1(3)	SCL2 output	–	PSC_1 = 0	PSL1_1 = 0	PS1_1 = 1
	SCL2 input	PD7_1 = 0	–	–	PS1_1 = 0
P9_1	SCL3 output	–	–	PSL3_1 = 0	PS3_1 = 1
	SCL3 input	PD9_1 = 0	–	–	PS3_1 = 0
P9_2	SDA3 output	–	–	PSL3_2 = 0	PS3_2 = 1
	SDA3 input	PD9_2 = 0	–	–	PS3_2 = 0
P9_6	SDA4 output	–	–	–	PS3_6 = 1
	SDA4 input	PD9_6 = 0	–	–	PS3_6 = 0
P9_7	SCL4 output	–	–	PSL3_7 = 0	PS3_7 = 1
	SCL4 input	PD9_7 = 0	–	–	PS3_7 = 0

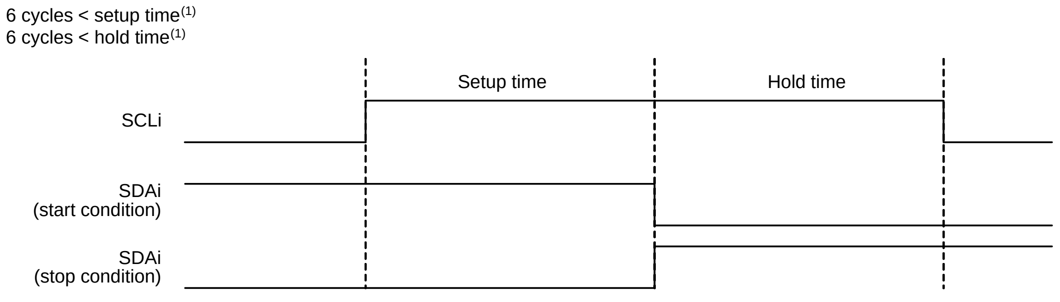
NOTES:

1. Set registers PS0, PS1, and PS3 after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
3. P7_0 and P7_1 are N-channel open drain output ports.

17.1.3.1 Detecting Start Condition and Stop Condition

The MCU detects the start condition and stop condition. The start condition detection interrupt request is generated when the SDAi (i = 0 to 4) pin level changes from high (“H”) to low (“L”) while the SCLi pin level is held “H”. The stop condition detection interrupt request is generated when the SDAi pin level changes from “L” to “H” while the SCLi pin level is held “H”.

The start condition detection interrupt shares the Interrupt Control Register and interrupt vector with the stop condition detection interrupt. The BBS bit in the UiSMR register determines which interrupt is requested.



i=0 to 4

NOTE:

1. These are cycles of the main clock oscillation frequency *f(XIN)*.

Figure 17.25 Start Condition or Stop Condition Detection

17.1.3.2 Start Condition or Stop Condition Output

The start condition is generated when the STAREQ bit in the UiSMR4 register (i = 0 to 4) is set to 1 (start).
The restart condition is generated when the RSTAREQ bit in the UiSMR4 register is set to 1 (start).
The stop condition is generated when the STPREQ bit in the UiSMR4 is set to 1 (start).
The following is the procedure to output the start condition, restart condition, or stop condition.

(1) Set the STAREQ bit, RSTAREQ bit, or STPREQ bit to 1 (start).
(2) Set the STSPSEL bit in the UiSMR4 register to 1 (start/stop condition generation circuit selected).

Table 17.12 and Figure 17.26 show functions of the STSPSEL bit.

Table 17.12 STSPSEL Bit Function

Function	STSPSEL = 0	STSPSEL = 1
Output from pins SCLi and SDAi	Output the serial clock and data. Output of the start condition or stop condition is controlled by software utilizing port functions. (The start condition and stop condition are not automatically generated by hardware)	Output of the start condition or stop condition is controlled by the status of bits STAREQ, RSTAREQ, and STPREQ.
Timing to generate start condition and stop condition interrupt requests	When start condition and stop condition are detected	When start condition and stop condition generation are completed

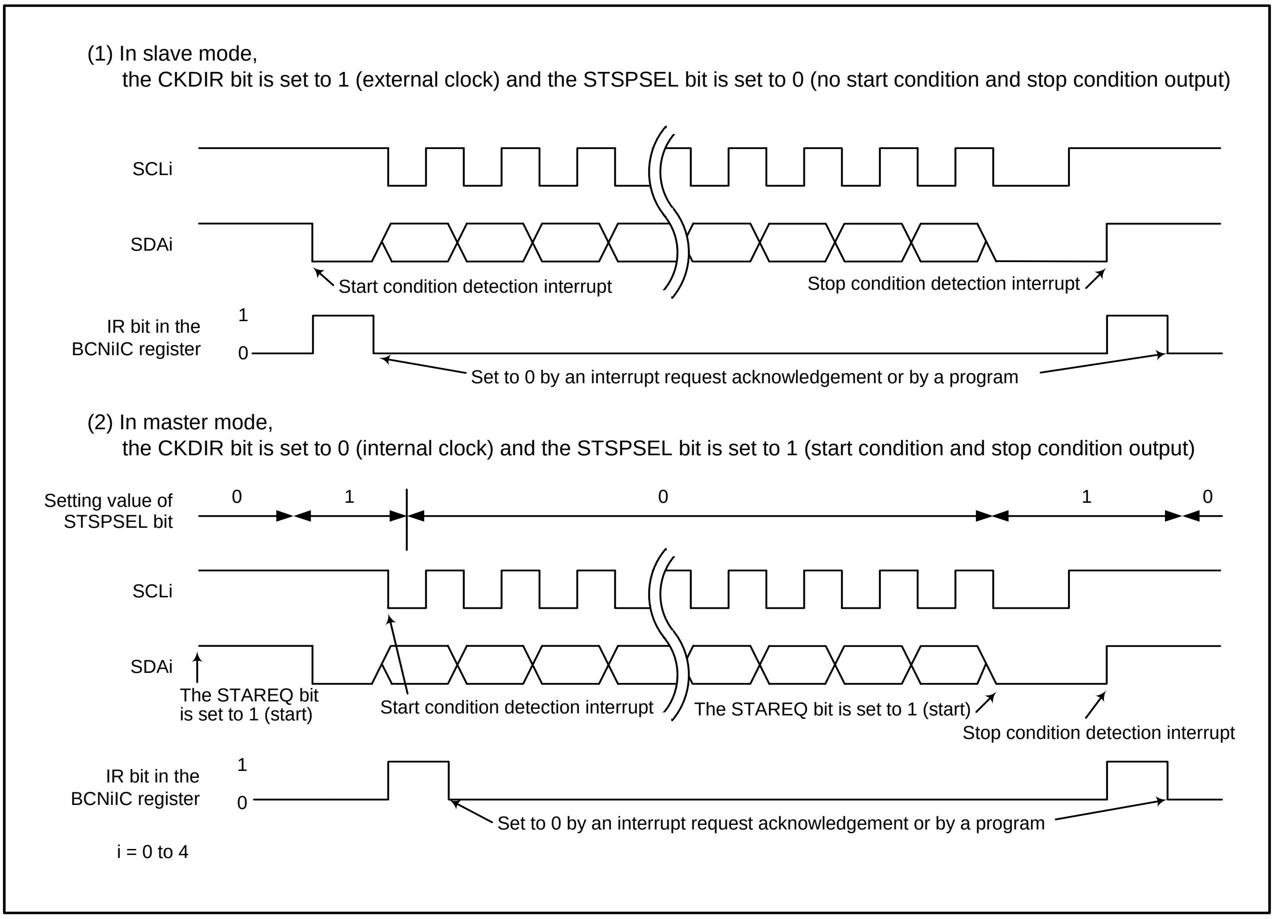

(1) In slave mode,
the CKDIR bit is set to 1 (external clock) and the STSPSEL bit is set to 0 (no start condition and stop condition output)
SCLi
SDAi
Start condition detection interrupt
Stop condition detection interrupt
IR bit in the
BCNiIC register
1
0
Set to 0 by an interrupt request acknowledgement or by a program
(2) In master mode,
the CKDIR bit is set to 0 (internal clock) and the STSPSEL bit is set to 1 (start condition and stop condition output)
Setting value of
STSPSEL bit
0
1
0
1
0
SCLi
SDAi
The STAREQ bit
is set to 1 (start)
Start condition detection interrupt
The STAREQ bit is set to 1 (start)
Stop condition detection interrupt
IR bit in the
BCNiIC register
1
0
Set to 0 by an interrupt request acknowledgement or by a program
i = 0 to 4

Figure 17.26 STSPSEL Bit Function

17.1.3.3 Arbitration

The ABC bit in the UiSMR register (i = 0 to 4) determines an update timing of the ABT bit in the UiRB register. At the rising edge of the clock input to the SCLi pin, the MCU determines whether a transmit data matches data input to the SDAi pin.
When the ABC bit is set to 0 (update per bit), the ABT bit becomes 1 (detected - arbitration is lost) as soon as a data discrepancy is detected. The ABT bit remains 0 (not detected - arbitration is won) if not detected. When the ABC bit is set to 1 (update per byte), the ABT bit becomes 1 at the falling edge of the ninth cycle of the serial clock if discrepancy is ever detected. When the ABT bit is updated per byte, set the ABT bit to 0 after an ACK detection in the first byte data is completed. Then the next byte data transfer can be started.
When the ALS bit in the UiSMR2 register is set to 1 (SDAi output stopped) and the ABT bit becomes 1 (detected - arbitration is lost), the SDAi pin is placed in a high-impedance state simultaneously.

17.1.3.4 Serial Clock

The serial clock is used to transmit and receive data as is shown in Figure 17.24.
By setting the CSC bit in the UiSMR2 register to 1 (clock synchronized), an internally generated clock (internal SCLi) is synchronized with the external clock applied to the SCLi pin. If the CSC bit is set to 1, the internal SCLi becomes low (“L”) when the internal SCLi is held high (“H”) and the external clock applied to the SCLi pin is at the falling edge. The contents of the UiBRG register are reloaded and a counting for “L” period is started. When the external clock applied to SCLi pin is held “L” and then the internal SCLi changes “L” to “H”, the UiBRG counter stops. The counting is resumed when the clock applied to SCLi pin becomes “H”. The UARTi serial clock is equivalent to logical AND operation of the internal SCLi and the clock signal applied to the SCLi pin.
The serial clock is synchronized between a half cycle before the falling edge of the first bit and the rising edge of the ninth bit of the internal SCLi. Select the internal clock as the serial clock while the CSC bit is set to 1.
The SWC bit in the UiSMR2 register determines whether an output signal from the SCLi pin is held “L” at the falling edge of the ninth cycle of the serial clock or not.
When the SCLHI bit in the UiSMR4 register is set to 1 (SCLi output stopped), a SCLi output stops as soon as the stop condition is detected (the SCLi pin is in a high-impedance state).
When the SWC2 bit in the UiSMR2 register is set to 1 (SCLi pin is held “L”), the SCLi pin forcibly outputs an “L” even in the middle of transmitting and receiving. The fixed “L” output from the SCLi pin is cancelled by setting the SWC2 bit to 0 (serial clock), and then the serial clock inputs to or outputs from the SCLi pin.
When the CKPH bit in the UiSMR3 register is set to 1 (clock delay) and the SWC9 bit in the UiSMR4 register is set to 1 (SCLi pin is held “L” after receiving 9th bit), an output signal from the SCLi pin is held “L” at the next falling edge to the ninth bit of the clock. The fixed “L” output from the SCLi pin is cancelled by setting the SWC9 bit to 0 (no wait state/release wait state).

17.1.3.5 SDA Output

Values set in bits 7 to 0 (D7 to D0) in the UiTB register are output in descending order from D7. The ninth bit (D8) is ACK or NACK.
Set the default value of SDAi transmit output, while the IICM bit in the UiSMR register is set to 1 (I^2C mode) and bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled).
Bits DL2 to DL0 in the UiSMR3 register determine no delay or delay of 2 to 8 UiBRG register count source cycles are added to an SDAi output.
When the SDHI bit in the UiSMR2 register is set to 1 (SDA output stopped), the SDAi pin is forcibly placed in a high-impedance state. Do not write to the SDHI bit at the rising edge of the UARTi serial clock. The ABT bit in the UiRB register may become 1 (detected).

17.1.3.6 SDA Input

When the IICM2 bit in the UiSMR2 register (i = 0 to 4) is set to 0, the first eight bits of received data are stored into bits 7 to 0 (D7 to D0) in the UiRB register. The ninth bit (D8) is ACK or NACK.
When the IICM2 bit is set to 1, the first seven bits (D7 to D1) of received data are stored into bits 6 to 0 in the UiRB register. The eighth bit (D0) is stored into bit 8 in the UiRB register.
If the IICM2 bit is set to 1 and the CKPH bit in the UiSMR3 register is set to 1 (clock delay), the same data as that of when setting the IICM2 bit to 0 can be returned, by reading the UiRB register after the rising edge of the ninth bit of the serial clock.

17.1.3.7 ACK, NACK

When the STSPSEL bit in the UiSMR4 register is set to 0 (start/stop condition not output) and the ACKC bit in the UiSMR4 register is set to 1 (ACK data output), the SDAi pin outputs the setting value, ACK or NACK, of the ACKD bit in the UiSMR4 register.
If the IICM2 bit is set to 0, the NACK interrupt request is generated when the SDAi pin is held high (“H”) at the rising edge of the ninth bit of the serial clock. The ACK interrupt request is generated when the SDAi pin is held low (“L”) at the rising edge of the ninth bit of the serial clock.
When ACK is selected to generate a DMA request source, the DMA transfer is activated by an ACK detection.

17.1.3.8 Transmit and Receive Operation Initialization

The following occurs when the STC bit in the UiSMR2 register is set to 1 (UARTi initialized) and the start condition is detected:

- The UARTi transmit shift register is initialized and the contents of the UiTB register are transferred to the UARTi transmit shift register. Then, the transmit operation is started at the next serial clock input to the SCLi pin. UARTi output value remains the same as when the start condition was detected until the first bit data is output.
- The UARTi receive shift register is initialized and the receive operation is started at the next serial clock input to the SCLi pin.
- The SWC bit in the UiSMR2 register becomes 1 (SCLi pin is held “L” after receiving 8th bit). An output from the SCLi pin becomes “L” at the falling edge of the ninth bit of the serial clock.

When UARTi transmit/receive operation is started with setting the STC bit to 1, the TI bit in the UiC1 register remains unchanged. Also, select the external clock as the serial clock to start UARTi transmit/receive operation with setting the STC bit to 1.

17.1.4 Special Mode 2

Full-duplex clock synchronous serial communications are allowed in this mode. SS function is used for transmit and receive control. The input signal to the $\overline{\text{SSi}}$ pin (i = 0 to 4) determines whether the transmit and receive operation is enabled or disabled. When it is disabled, the output pin is placed in a high-impedance state. Table 17.13 lists specifications of special mode 2. Table 17.14 lists pin settings. Figure 17.27 shows register settings.

Table 17.13 Special Mode 2 Specifications

Item	Specification
Data format	Data length: 8 bits long
Baud rate	• The CKDiR bit in the UiMR register (i = 0 to 4) is set to 0 (internal clock): fj / (2 (m + 1)) fj = f1, f8, f2n[1] m: setting value of the UiBRG register (00h to FFh) • The CKDIR bit to 1 (external clock): input from the CLKi pin
Transmit/receive control	• SS function Output pin is placed in a high-impedance state to avoid data conflict between a master and other masters, or a slave and other slaves.
Transmit and receive start condition	Internal clock is selected (master mode): • Set the TE bit in the UiC1 register to 1 (transmit operation enabled) • The TI bit in the UiC1 register is 0 (data in the UiTB register) • Set the RE bit in the UiC1 register to 1 (receive operation enabled) • “H” signal is applied to the $\overline{\text{SSi}}$ pin when the SS function is used External clock is selected (slave mode)[2]: • Set the TE bit to 1 • The TI bit is 0 • Set the RE bit to 1 • “L” signal is applied to the $\overline{\text{SSi}}$ pin If transmit-only operation is performed, the RE bit setting is not required in both cases.
Interrupt request generation timing	Transmit interrupt (The UiIRS bit in the UiC1 register selects one of the following): • The UiIRS bit is set to 0 (no data in the UiTB register): when data is transferred from the UiTB register to the UARTi transmit shift register (transmit operation started) • The UiIRS bit is set to 1 (transmit operation completed): when data transmit operation from the UARTi transmit shift register is completed Receive interrupt: • When data is transferred from the UARTi receive shift register to the UiRB register (receive operation completed)
Error detection	• Overrun error[3] Overrun error occurs when the 7th bit of the next data is received before reading the UiRB register • Mode error Mode error occurs when an “L” signal is applied to the $\overline{\text{SSi}}$ pin in master mode
Selectable function	• CLK polarity Transmit data output timing and receive data input timing can be selected • LSB first or MSB first Data is transmitted or received from either bit 0 or bit 7 • Serial data logic inverse Transmit and receive data are logically inverted • TXD and RXD I/O polarity Inverse The level output from the TXD pin and the level applied to the RXD pin are inverted. • Clock phase One of four combinations of serial clock polarity and phase can be selected

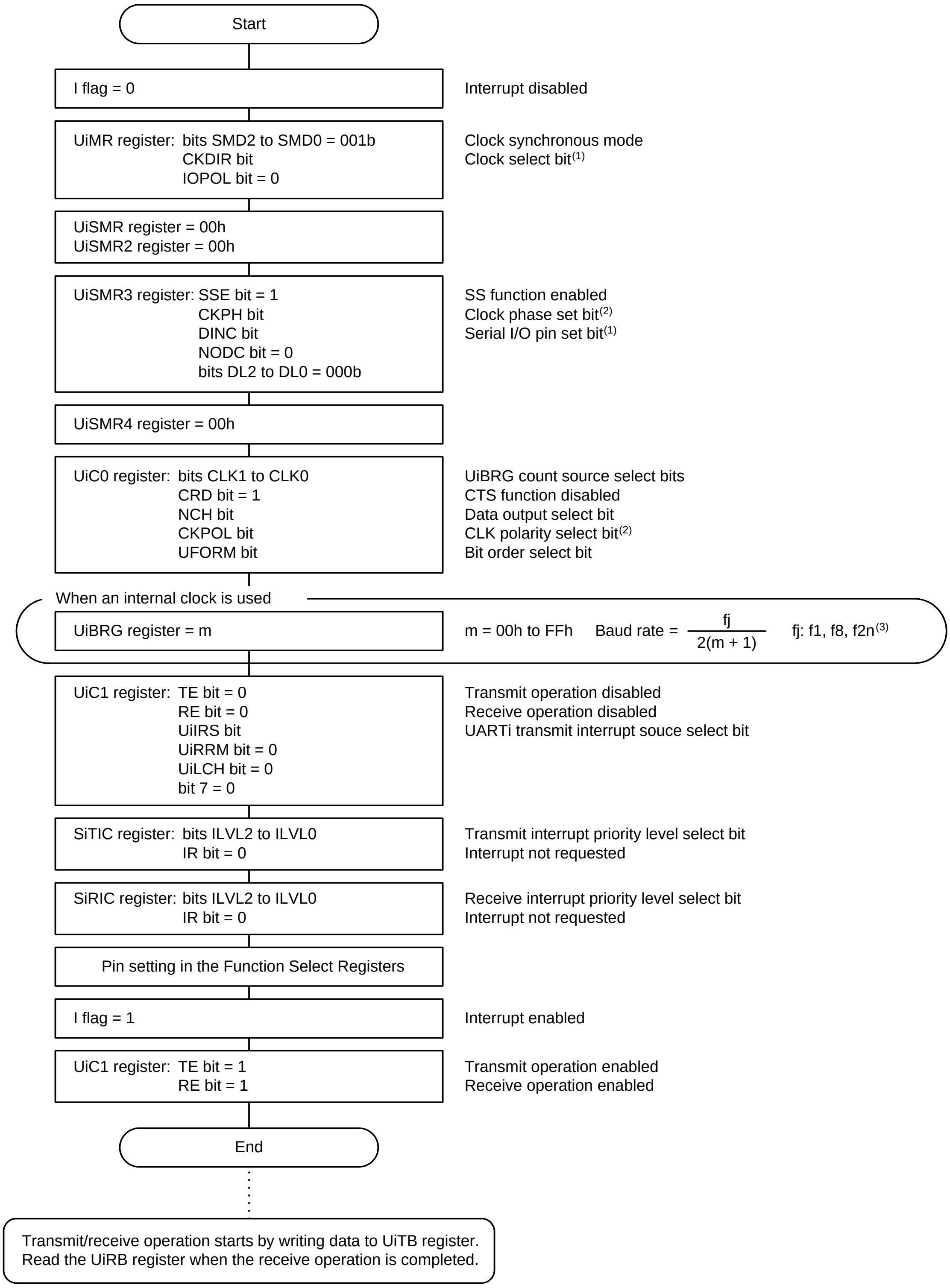
NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. If an external clock is selected, ensure that an “H” signal is applied to the CLKi pin when the CKPOL bit in the UiC0 register is set to 0, and that an “L” signal is applied when the CKPOL bit is set to 1.
3. If an overrun error occurs, a read from the UiRB register returns undefined values. The IR bit in the SiRIC register remains unchanged as 0 (interrupt not requested).

Table 17.14 Pin Settings in Special Mode 2

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC Register	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers(1)(2)
P6_0	$\overline{\text{SS0}}$ input	PD6_0 = 0	–	–	PS0_0 = 0
P6_1	CLK0 output (master)	–	–	PSL0_1 = 0	PS0_1 = 1
	CLK0 input (slave)	PD6_1 = 0	–	–	PS0_1 = 0
P6_2	RXD0 input (master)	PD6_2 = 0	–	–	PS0_2 = 0
	STXD0 output (slave)	–	–	PSL0_2 = 1	PS0_2 = 1
P6_3	TXD0 output (master)	–	–	PSL0_3 = 0	PS0_3 = 1
	SRXD0 input (slave)	PD6_3 = 0	–	–	PS0_3 = 0
P6_4	$\overline{\text{SS1}}$ input	PD6_4 = 0	–	–	PS0_4 = 0
P6_5	CLK1 output (master)	–	–	PSL0_5 = 0	PS0_5 = 1
	CLK1 input (slave)	PD6_5 = 0	–	–	PS0_5 = 0
P6_6	RXD1 input (master)	PD6_6 = 0	–	–	PS0_6 = 0
	STXD1 output (slave)	–	–	PSL0_6 = 1	PS0_6 = 1
P6_7	TXD1 output (master)	–	–	PSL0_7 = 0	PS0_7 = 1
	SRXD1 input (slave)	PD6_7 = 0	–	–	PS0_7 = 0
P7_0(3)	TXD2 output (master)	–	PSC_0 = 0	PSL1_0 = 0	PS1_0 = 1
	SRXD2 input (slave)	PD7_0 = 0	–	–	PS1_0 = 0
P7_1(3)	RXD2 input (master)	PD7_1 = 0	–	–	PS1_1 = 0
	STXD2 output (slave)	–	–	PSL1_1 = 1	PS1_1 = 1
P7_2	CLK2 output (master)	–	PSC_2 = 0	PSL1_2 = 0	PS1_2 = 1
	CLK2 input (slave)	PD7_2 = 0	–	–	PS1_2 = 0
P7_3	$\overline{\text{SS2}}$ input	PD7_3 = 0	–	–	PS1_3 = 0
P9_0	CLK3 output (master)	–	–	PSL3_0 = 0	PS3_0 = 1
	CLK3 input (slave)	PD9_0 = 0	–	–	PS3_0 = 0
P9_1	RXD3 input (master)	PD9_1 = 0	–	–	PS3_1 = 0
	STXD3 output (slave)	–	–	PSL3_1 = 1	PS3_1 = 1
P9_2	TXD3 output (master)	–	–	PSL3_2 = 0	PS3_2 = 1
	SRXD3 input (slave)	PD9_2 = 0	–	–	PS3_2 = 0
P9_3	$\overline{\text{SS3}}$ input	PD9_3 = 0	–	PSL3_3 = 0	PS3_3 = 0
P9_4	$\overline{\text{SS4}}$ input	PD9_4 = 0	–	PSL3_4 = 0	PS3_4 = 0
P9_5	CLK4 output (master)	–	–	–	PS3_5 = 1
	CLK4 input (slave)	PD9_5 = 0	–	PSL3_5 = 0	PS3_5 = 0
P9_6	TXD4 output (master)	–	–	–	PS3_6 = 1
	SRXD4 input (slave)	PD9_6 = 0	–	PSL3_6 = 0	PS3_6 = 0
P9_7	RXD4 input (master)	PD9_7 = 0	–	–	PS3_7 = 0
	STXD4 output (slave)	–	–	PSL3_7 = 1	PS3_7 = 1

NOTES:

1. Set registers PS0, PS1, and PS3 after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
3. P7_0 and P7_1 are N-channel open drain output ports.


Start
I flag = 0
Interrupt disabled
UiMR register: bits SMD2 to SMD0 = 001b
CKDIR bit
IOPOL bit = 0
Clock synchronous mode
Clock select bit(1)
UiSMR register = 00h
UiSMR2 register = 00h
UiSMR3 register: SSE bit = 1
CKPH bit
DINC bit
NODC bit = 0
bits DL2 to DL0 = 000b
SS function enabled
Clock phase set bit(2)
Serial I/O pin set bit(1)
UiSMR4 register = 00h
UiC0 register: bits CLK1 to CLK0
CRD bit = 1
NCH bit
CKPOL bit
UFORM bit
UiBRG count source select bits
CTS function disabled
Data output select bit
CLK polarity select bit(2)
Bit order select bit
When an internal clock is used
UiBRG register = m
m = 00h to FFh
Baud rate = $\frac{fj}{2(m + 1)}$
fj: f1, f8, f2n(3)
UiC1 register: TE bit = 0
RE bit = 0
UiIRS bit
UiRRM bit = 0
UiLCH bit = 0
bit 7 = 0
Transmit operation disabled
Receive operation disabled
UARTi transmit interrupt souce select bit
SiTIC register: bits ILVL2 to ILVL0
IR bit = 0
Transmit interrupt priority level select bit
Interrupt not requested
SiRIC register: bits ILVL2 to ILVL0
IR bit = 0
Receive interrupt priority level select bit
Interrupt not requested
Pin setting in the Function Select Registers
I flag = 1
Interrupt enabled
UiC1 register: TE bit = 1
RE bit = 1
Transmit operation enabled
Receive operation enabled
End
Transmit/receive operation starts by writing data to UiTB register.
Read the UiRB register when the receive operation is completed.

i = 0 to 4

NOTES:

1. Set to 0 in master mode, and set to 1 in slave mode.
2. The clock phase is determined by the combination of the CKPH and CKPOL bits in the UiSMR3 register.
3. Bits CNT3 to CNT0 select no division (n = 0) or divide-by-2n (n = 1 to 15).

Figure 17.27 Register Settings in Special Mode 2

17.1.4.1 Master Mode

Master mode is entered when the DINC bit in the UiSMR3 register (i = 0 to 4) is set to 0. The following pins are used in master mode.

- TXDi: transmit data output
- RXDi: receive data input
- CLKi: serial clock output

To use the SS function, set the SSE bit in the UiSMR3 register to 1. A transmit and receive operation is performed while an “H” is applied to the $\overline{\text{SSi}}$ pin. If an “L” is applied to the $\overline{\text{SSi}}$ pin, the ERR bit in the UiSMR3 register becomes 1 (mode error occurred) and pins CLKi and TXDi are placed in high-impedance states. Set the UiIRS bit in the UiC1 register to 1 (Transmit completion as interrupt source) to verify whether a mode error has occurred or not by checking the EER bit in the transmission complete interrupt routine. To resume serial communication after a mode error occurs, set the ERR bit to 0 (no mode error) while an “H” signal is applied to the $\overline{\text{SSi}}$ pin. Pins TXDi and CLKi become in output mode.

17.1.4.2 Slave Mode

Slave mode is entered when the DINC bit in the UiSMR3 register is set to 1. The following pins are used in slave mode.

- STXDi: transmit data output
- SRXDi: receive data input
- CLKi: serial clock input

To use the SS function, set the SSE bit in the UiSMR3 register to 1. When an “L” signal is applied to the $\overline{\text{SSi}}$ input pin, the serial clock input is enabled, and a transmit and receive operation becomes available. When an “H” signal is applied to the $\overline{\text{SSi}}$ pin, the serial clock input to the CLKi pin is ignored and the STXDi pin is placed in a high-impedance state.

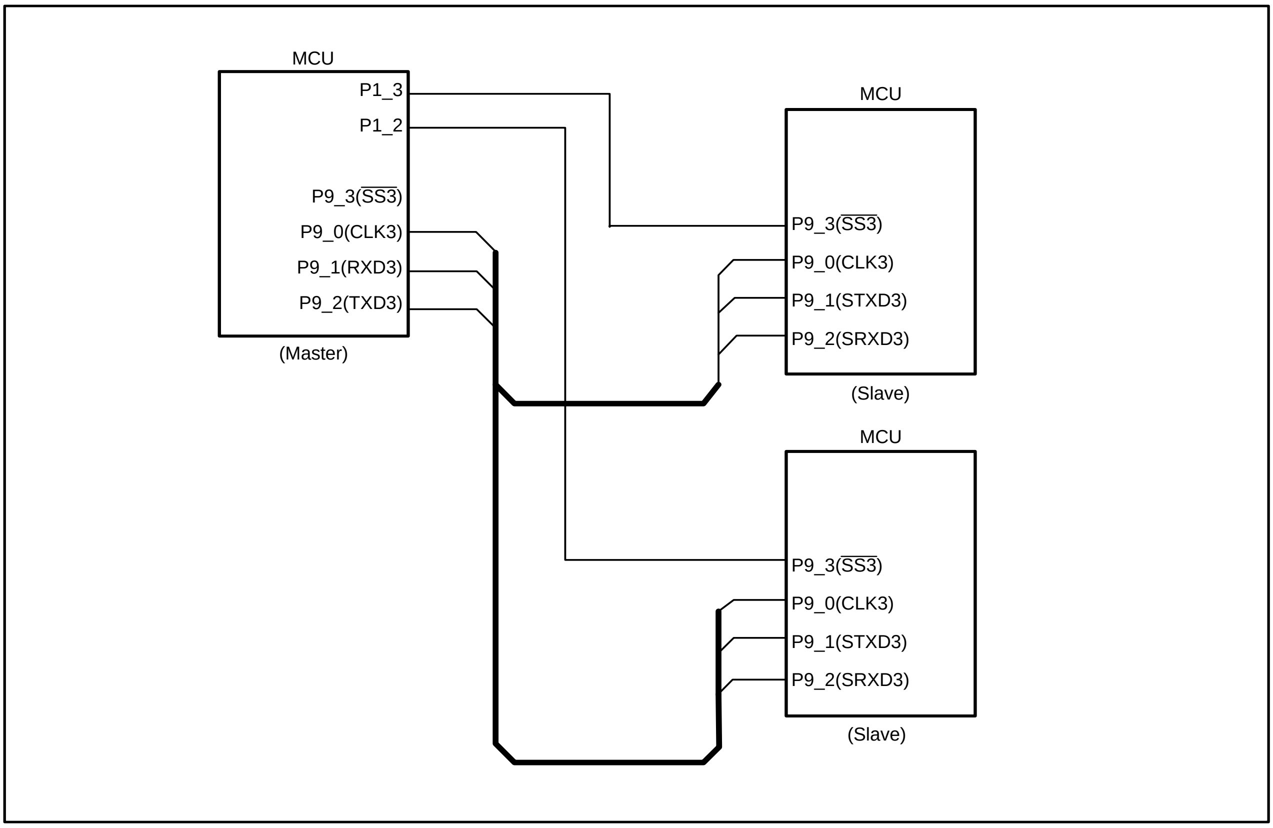
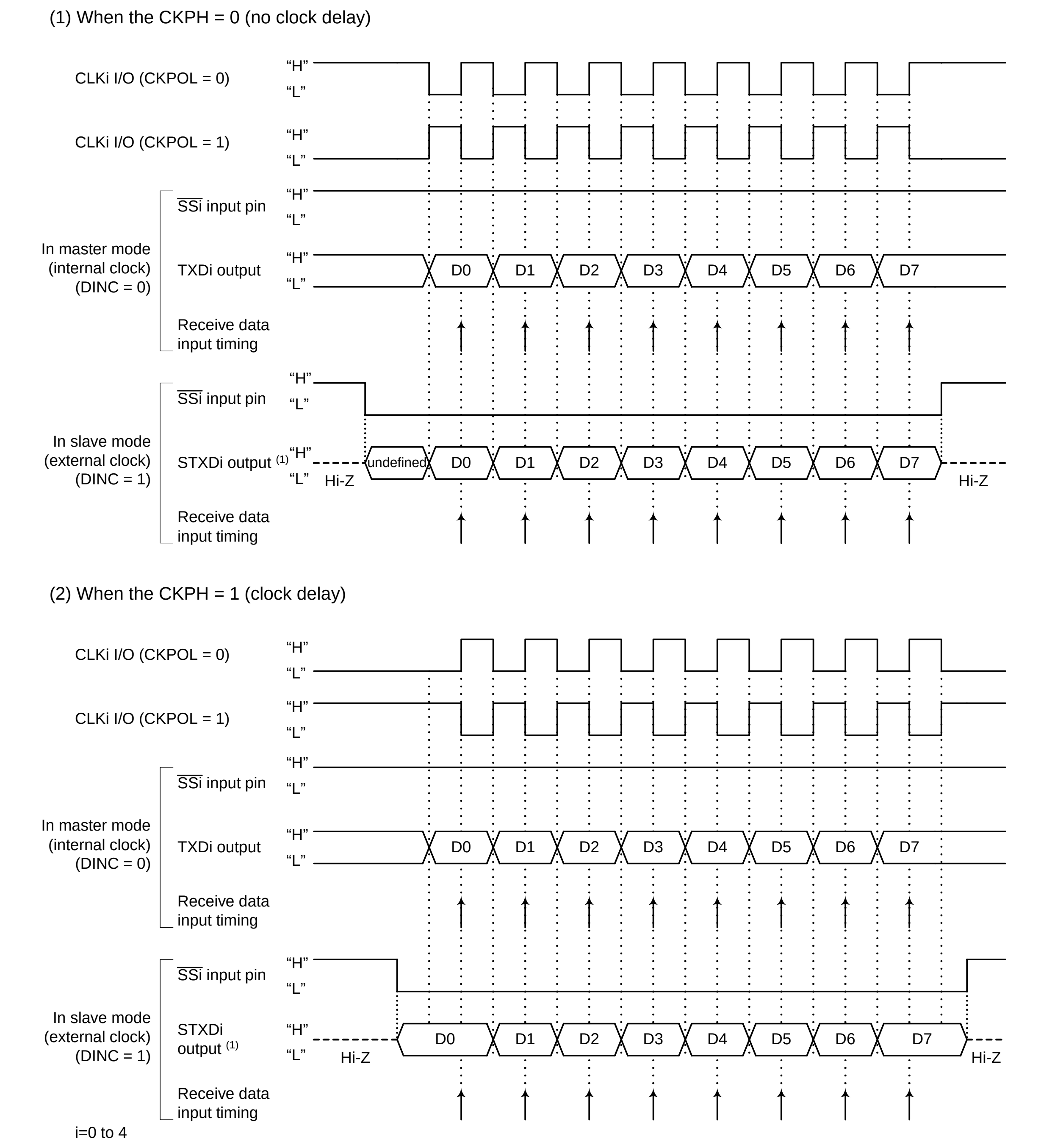


Figure 17.28 Serial Bus Communication Control with $\overline{\text{SSi}}$ Pin

17.1.4.3 Clock Phase Setting Function

The clock polarity and clock phase are selected from four combinations of the CKPH and CKPOL bits in the UiSMR3 register (i = 0 to 4). The master must have the same serial clock polarity and phase as the slaves involved in the communication. Figure 17.29 shows a transmit and receive operation timing.


(1) When the CKPH = 0 (no clock delay)
CLKi I/O (CKPOL = 0)
CLKi I/O (CKPOL = 1)
“H”
“L”
In master mode (internal clock) (DINC = 0)
SSi input pin
TXDi output
Receive data input timing
D0
D1
D2
D3
D4
D5
D6
D7
In slave mode (external clock) (DINC = 1)
STXDi output (1)
undefined
Hi-Z
(2) When the CKPH = 1 (clock delay)
STXDi output (1)
i=0 to 4

CKPH, DINC: bits in the UiSMR3 register
CKPOL: bit in the UiC0 register

NOTE:
1. P7_0 and P7_1 are N-channel open drain output ports. They must be pulled up externally to output data.

Figure 17.29 Transmit and Receive Operation Timing in Special Mode 2

17.1.5 Special Mode 3 (GCI Mode)

Full-duplex clock synchronous serial communications are allowed in this mode. When a trigger is input to the $\overline{\text{CTSi}}$ (i = 0 to 4) pin, the internal clock which is synchronized with the continuous external clock is generated, and a transmit and receive operation is started.

Table 17.15 lists specifications of GCI mode. Table 17.16 lists pin settings. Figure 17.30 shows register settings.

Table 17.15 GCI Mode Specifications

Item	Specification
Data format	Data length: 8 bits long
Serial clock	Select the external clock Set the CKDIR bit in the UiMR register (i = 0 to 4) to 1 (external clock). When a trigger is input, the external clock or the clock divided by 2 becomes the serial clock.
Transmit and receive start condition	A transmit and receive operation starts when a trigger is input to the $\overline{\text{CTSi}}$ pin after all the following are met: • Set the TE bit in the UiC1 register to 1 (transmit operation enabled) • The TI bit in the UiC1 register is 1 (data in the UiTB register) • Set the RE bit in the UiC1 register to 1 (receive operation enabled) • Set the SCLKSTPB bit in the UiC1 register is set to 0 (clock-divided synchronization stopped) The SCLKSTPB bit becomes 1 (clock-divided synchronization started) when a trigger is input to the $\overline{\text{CTSi}}$ pin
Transmit and receive stop condition	The SCLKSTPB bit in the UiC1 register is set to 0
Interrupt request generation timing	Transmit interrupt (The UiIRS bit in the UiC1 register selects one of the following): • The UiIRS bit is set to 0 (no data in the UiTB register): when data is transferred from the UiTB register to the UARTi transmit shift register (transmit operation started) • The UiIRS bit is set to 1 (transmit operation completed): when data transmit operation from the UARTi transmit shift register is completed Receive interrupt: • When data is transferred from the UARTi receive shift register to the UiRB register (receive operation completed)
Error detection	Overrun error[1] Overrun error occurs when the 7th bit of the next data is received before reading the UiRB register

NOTE:

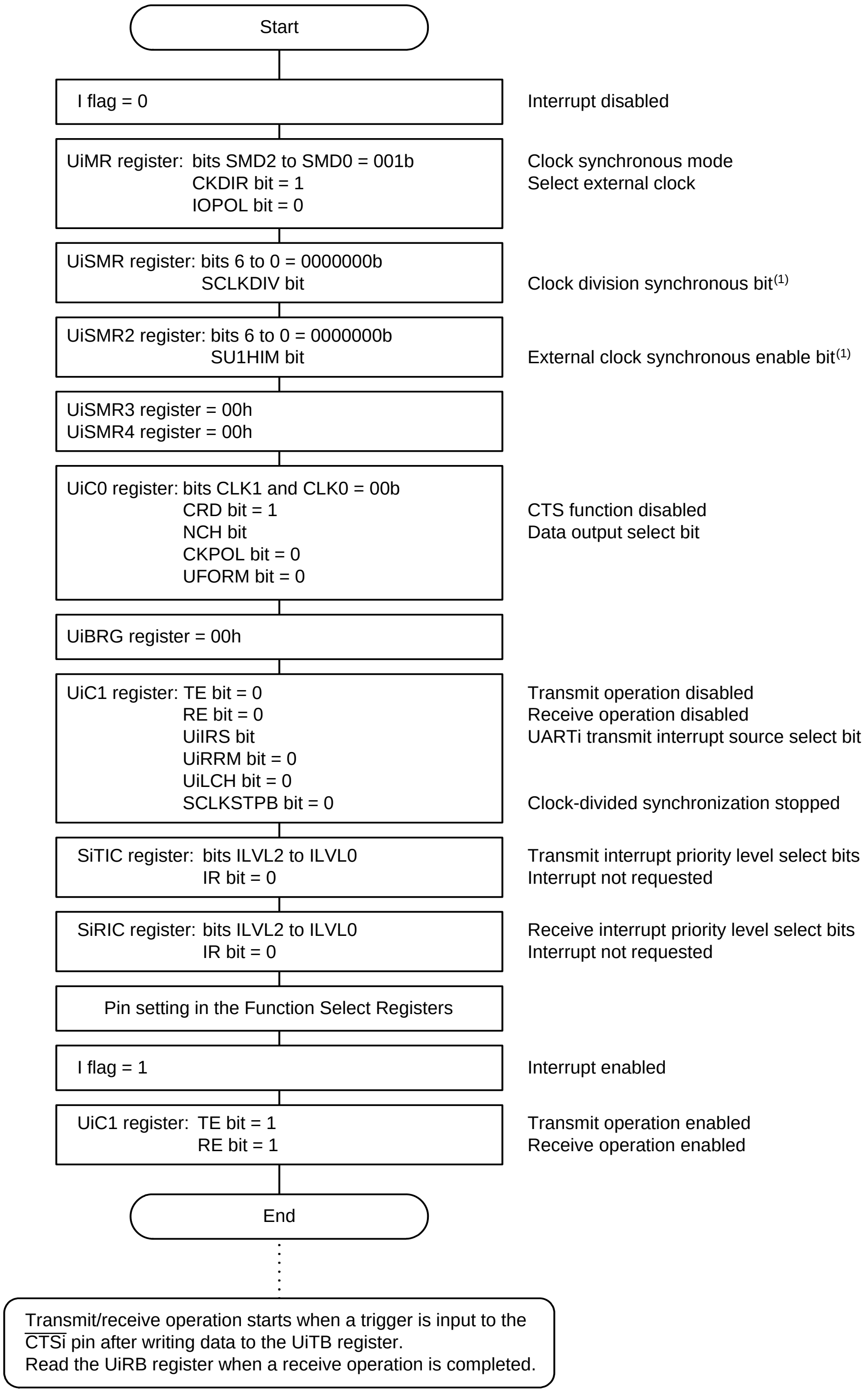
1. If an overrun error occurs, a read from the UiRB register returns undefined values. The IR bit in the SiRIC register remains unchanged as 0 (interrupt not requested).

Table 17.16 Pin Settings in GCI Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers[2]	PSC Register	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers[1][2]
P6_0	$\overline{\text{CTS0}}$ input[3]	PD6_0 = 0	–	–	PS0_0 = 0
P6_1	CLK0 input	PD6_1 = 0	–	–	PS0_1 = 0
P6_2	RXD0 input	PD6_2 = 0	–	–	PS0_2 = 0
P6_3	TXD0 output	–	–	PSL0_3 = 0	PS0_3 = 1
P6_4	$\overline{\text{CTS1}}$ input[3]	PD6_4 = 0	–	–	PS0_4 = 0
P6_5	CLK1 input	PD6_5 = 0	–	–	PS0_5 = 0
P6_6	RXD1 input	PD6_6 = 0	–	–	PS0_6 = 0
P6_7	TXD1 output	–	–	PSL0_7 = 0	PS0_7 = 1
P7_0[4]	TXD2 output	–	PSC_0 = 0	PSL1_0 = 0	PS1_0 = 1
P7_1	RXD2 input	PD7_1 = 0	–	–	PS1_1 = 0
P7_2	CLK2 input	PD7_2 = 0	–	–	PS1_2 = 0
P7_3	$\overline{\text{CTS2}}$ input[3]	PD7_3 = 0	–	–	PS1_3 = 0
P9_0	CLK3 input	PD9_0 = 0	–	–	PS3_0 = 0
P9_1	RXD3 input	PD9_1 = 0	–	–	PS3_1 = 0
P9_2	TXD3 output	–	–	PSL3_2 = 0	PS3_2 = 1
P9_3	$\overline{\text{CTS3}}$ input[3]	PD9_3 = 0	–	PSL3_3 = 0	PS3_3 = 0
P9_4	$\overline{\text{CTS4}}$ input[3]	PD9_4 = 0	–	PSL3_4 = 0	PS3_4 = 0
P9_5	CLK4 input	PD9_5 = 0	–	PSL3_5 = 0	PS3_5 = 0
P9_6	TXD4 output	–	–	–	PS3_6 = 1
P9_7	RXD4 input	PD9_7 = 0	–	–	PS3_7 = 0

NOTES:

1. Set registers PS0, PS1, and PS3 after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
3. $\overline{\text{CTS}}$ input is used as a trigger signal input.
4. P7_0 is an N-channel open drain output port.


Start
I flag = 0
Interrupt disabled
UiMR register: bits SMD2 to SMD0 = 001b
CKDIR bit = 1
IOPOL bit = 0
Clock synchronous mode
Select external clock
UiSMR register: bits 6 to 0 = 0000000b
SCLKDIV bit
Clock division synchronous bit(1)
UiSMR2 register: bits 6 to 0 = 0000000b
SU1HIM bit
External clock synchronous enable bit(1)
UiSMR3 register = 00h
UiSMR4 register = 00h
UiC0 register: bits CLK1 and CLK0 = 00b
CRD bit = 1
NCH bit
CKPOL bit = 0
UFORM bit = 0
CTS function disabled
Data output select bit
UiBRG register = 00h
UiC1 register: TE bit = 0
RE bit = 0
UiIRS bit
UiRRM bit = 0
UiLCH bit = 0
SCLKSTPB bit = 0
Transmit operation disabled
Receive operation disabled
UARTi transmit interrupt source select bit
Clock-divided synchronization stopped
SiTIC register: bits ILVL2 to ILVL0
IR bit = 0
Transmit interrupt priority level select bits
Interrupt not requested
SiRIC register: bits ILVL2 to ILVL0
IR bit = 0
Receive interrupt priority level select bits
Interrupt not requested
Pin setting in the Function Select Registers
I flag = 1
Interrupt enabled
UiC1 register: TE bit = 1
RE bit = 1
Transmit operation enabled
Receive operation enabled
End
Transmit/receive operation starts when a trigger is input to the CTSi pin after writing data to the UiTB register.
Read the UiRB register when a receive operation is completed.

i = 0 to 4

NOTE:

1. The external clock synchronization function is determined by the combination of the SCLKDIV bit in the UiSMR register and the SU1HIM bit in the UiSMR2 register. Refer to the table "**Clock-Divided Synchronous Function Select**" for details.

Figure 17.30 Register Settings in GCI Mode

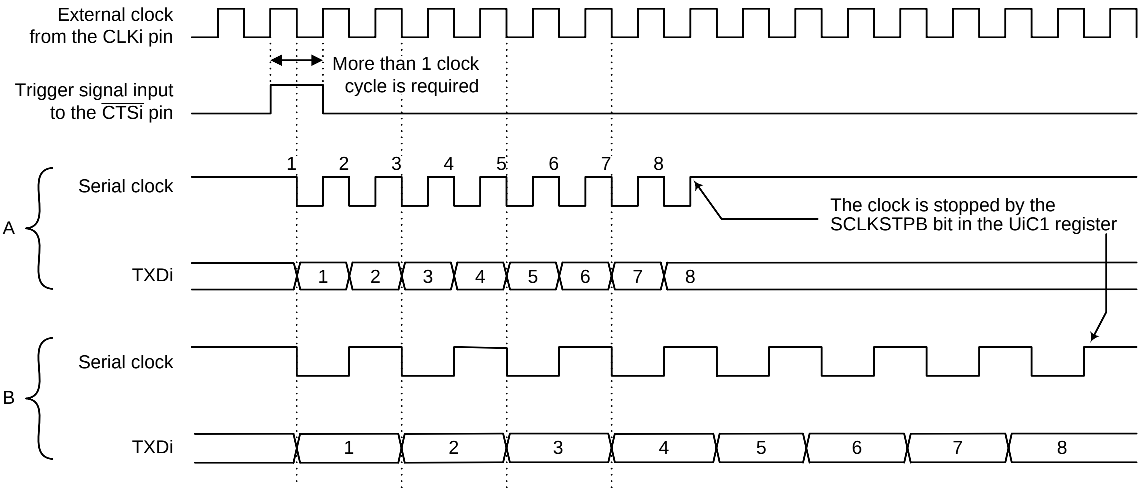
Set the SU1HIM bit in the UiSMR2 register (i = 0 to 4) and the SCLKDIV bit in the UiSMR register to values shown in Table 17.17, and apply a trigger signal to the $\overline{\text{CTSi}}$ pin. Then, the SCLKSTPB bit becomes 1 and a transmit and receive operation starts. Either the same clock cycle as the external clock or the external clock cycle divided by two can be selected for the serial clock.

When the SCLKSTPB bit in the UiC1 register is set to 0, a transmission and reception in progress stops immediately.

Figure 17.31 shows an example of the clock-divided synchronous function.

Table 17.17 Clock-Divided Synchronous Function Select

SCLKDIV bit in the UiSMR register	SU1HIM bit in the UiSMR2 register	Clock-Divided Synchronous Function
0	0	Not synchronized
0	1	Same clock cycle as the external clock
1	0 or 1	External clock cycle divided by 2


External clock from the CLKi pin
More than 1 clock cycle is required
Trigger signal input to the $\overline{\text{CTSi}}$ pin
1 2 3 4 5 6 7 8
Serial clock
A
TXDi
1 2 3 4 5 6 7 8
The clock is stopped by the SCLKSTPB bit in the UiC1 register
Serial clock
B
TXDi
1 2 3 4 5 6 7 8

i = 0 to 4

A: When the SCLKDIV bit in the UiSMR register is set to 0, and the SU1HIM bit in the UiSMR2 register is set to 1
B: When the SCLKDIV bit is set to 1, and SU1HIM bit is set to either 0 or 1.

Figure 17.31 Clock-Divided Synchronous Function

17.1.6 Special Mode 4 (SIM Mode)

In SIM mode, the MCU can communicate with SIM interface devices using UART mode. Both direct and inverse formats are available. The TXDi pin (i = 0 to 4) outputs a low-level ("L") signal when a parity error is detected.

Table 17.18 lists specifications of SIM mode. Table 17.19 list pin settings. Figure 17.32 lists register settings. Figure 17.33 shows an example of SIM interface operation. Figure 17.34 shows an example of SIM interface connection.

Table 17.18 SIM Mode Specifications

Item	Specification
Data format	• Data length 8-bit UART mode • One stop bit • Direct format: Parity: even Data logic: direct (not inverted) Bit order: LSB first • Inverse format: Parity: odd Data logic: inverse (inverted) Bit order: MSB first
Baud rate	Set the CKDIR bit in the UiMR register is 0 (internal clock): fj / (16 (m + 1)) fj = f1, f8, f2n(1) m: setting value of the UiBRG register (00h to FFh)
Transmit/receive control	CTS/RTS function disabled
Transmit start condition	To start transmit operation, all of the following must be met: • Set the TE bit in the UiC1 register to 1 (transmit operation enabled) • The TI bit in the UiC1 register is 0 (data in the UiTB register)
Receive start condition	To start receive operation, all of the following must be met: • Set the RE bit in the UiC1 register to 1 (receive operation enabled) • The start bit is detected
Interrupt request generation timing	Transmit interrupt: • Set the UiIRS bit in the UiC1 register to 1 (transmit operation completed) when the stop bit is output from the UARTi transmit shift register Receive interrupt: • when data is transferred from the UARTi receive shift register to the UiRB register (receive operation completed)
Error detection	• Overrun error(2) Overrun error occurs when the preceding bit of the stop bit of the next data is received before reading the UiRB register • Framing error Framing error occurs when the number of the stop bits set using the STPS bit in the UiMR register is not detected • Parity error Parity error occurs when parity is enabled and the received data does not have the correct even or odd parity set with the PRY bit in the UiMR register. • Error sum flag Error sum flag becomes 1 when an overrun, framing, or parity error occurs

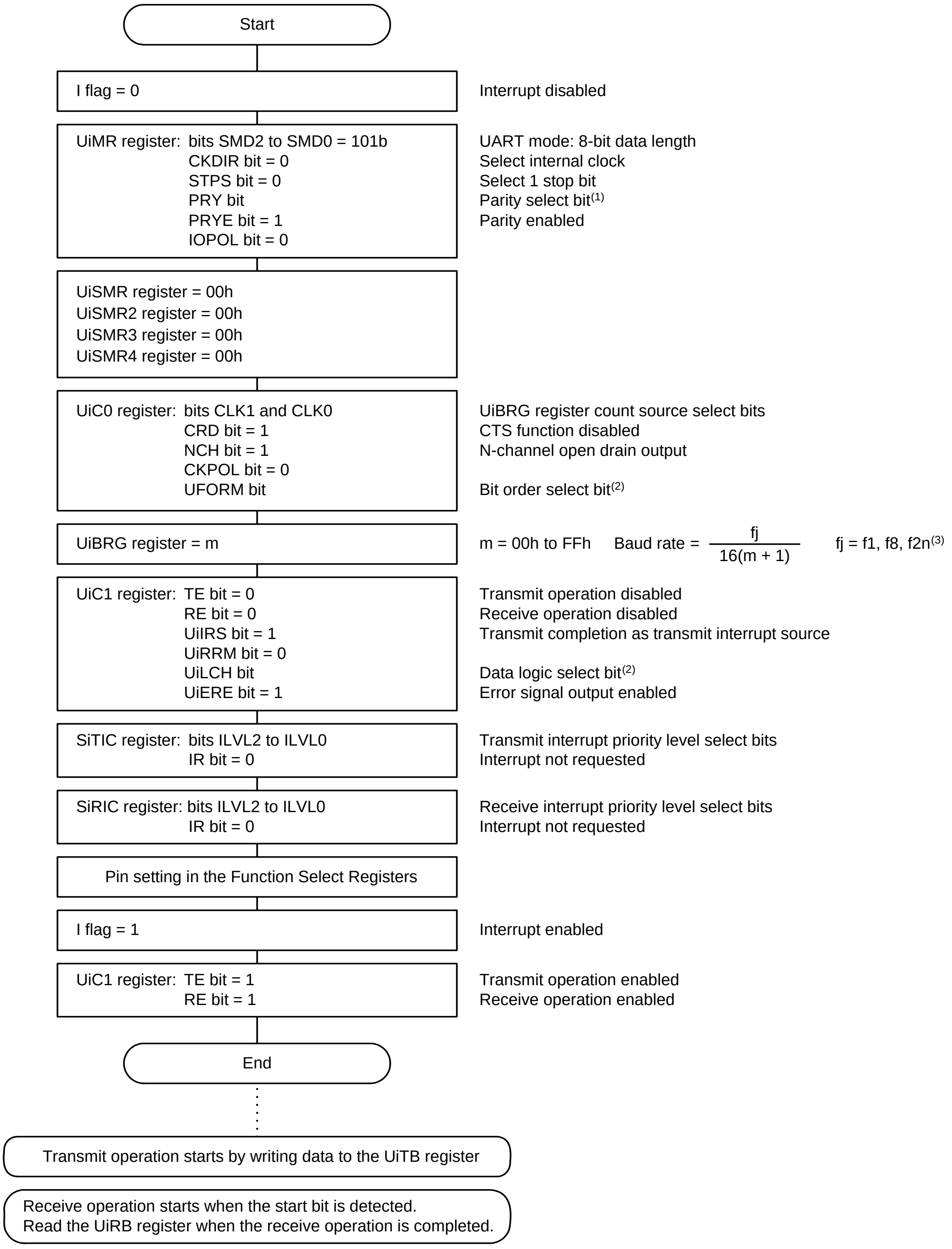
NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. If an overrun error occurs, a read from the UiRB register returns undefined values. The IR bit in the SiRIC register remains unchanged as 0 (interrupt not requested).

Table 17.19 Pin Settings in SIM Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC Register	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers(1)(2)
P6_2	RXD0 input	PD6_2 = 0	–	–	PS0_2 = 0
P6_3	TXD0 output	–	–	PSL0_3 = 0	PS0_3 = 1
P6_6	RXD1 input	PD6_6 = 0	–	–	PS0_6 = 0
P6_7	TXD1 output	–	–	PSL0_7 = 0	PS0_7 = 1
P7_0(3)	TXD2 output	–	PSC_0 = 0	PSL1_0 = 0	PS1_0 = 1
P7_1	RXD2 input	PD7_1 = 0	–	–	PS1_1 = 0
P9_1	RXD3 input	PD9_1 = 0	–	–	PS3_1 = 0
P9_2	TXD3 output	–	–	PSL3_2 = 0	PS3_2 = 1
P9_6	TXD4 output	–	–	–	PS3_6 = 1
P9_7	RXD4 input	PD9_7 = 0	–	–	PS3_7 = 0

NOTES:

1. Set registers PS0, PS1, and PS3 after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
3. P7_0 is an N-channel open drain output port.

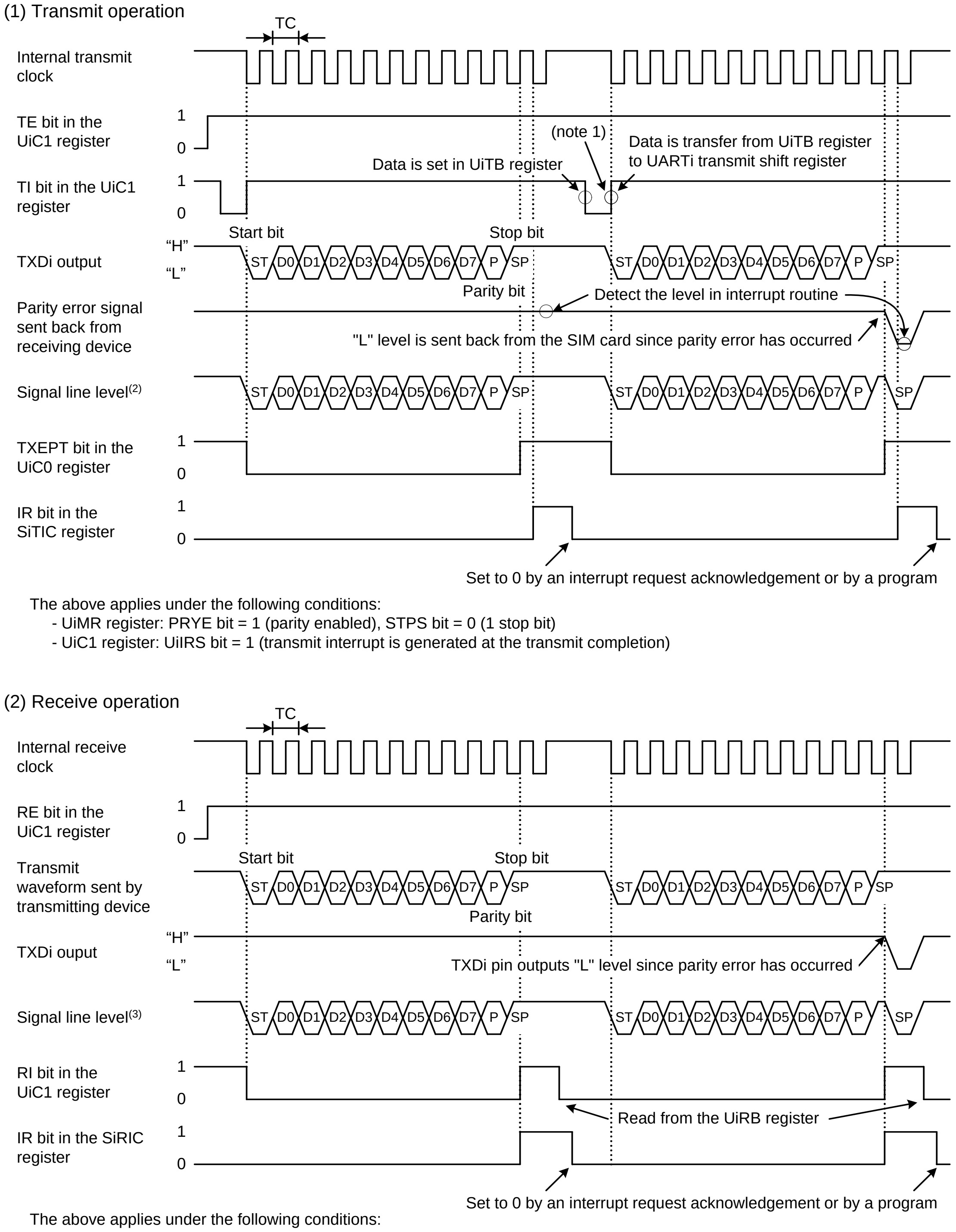

Start
I flag = 0
Interrupt disabled
UiMR register: bits SMD2 to SMD0 = 101b
CKDIR bit = 0
STPS bit = 0
PRY bit
PRYE bit = 1
IOPOL bit = 0
UART mode: 8-bit data length
Select internal clock
Select 1 stop bit
Parity select bit(1)
Parity enabled
UiSMR register = 00h
UiSMR2 register = 00h
UiSMR3 register = 00h
UiSMR4 register = 00h
UiC0 register: bits CLK1 and CLK0
CRD bit = 1
NCH bit = 1
CKPOL bit = 0
UFORM bit
UiBRG register count source select bits
CTS function disabled
N-channel open drain output
Bit order select bit(2)
UiBRG register = m
m = 00h to FFh
Baud rate = fj / 16(m + 1)
fj = f1, f8, f2n(3)
UiC1 register: TE bit = 0
RE bit = 0
UiIRS bit = 1
UiRRM bit = 0
UiLCH bit
UiERE bit = 1
Transmit operation disabled
Receive operation disabled
Transmit completion as transmit interrupt source
Data logic select bit(2)
Error signal output enabled
SiTIC register: bits ILVL2 to ILVL0
IR bit = 0
Transmit interrupt priority level select bits
Interrupt not requested
SiRIC register: bits ILVL2 to ILVL0
IR bit = 0
Receive interrupt priority level select bits
Interrupt not requested
Pin setting in the Function Select Registers
I flag = 1
Interrupt enabled
UiC1 register: TE bit = 1
RE bit = 1
Transmit operation enabled
Receive operation enabled
End
Transmit operation starts by writing data to the UiTB register
Receive operation starts when the start bit is detected.
Read the UiRB register when the receive operation is completed.

i = 0 to 4

NOTES:

1. Set to 1 in the direct format, and set to 0 in the inverse format.
2. Set to 0 in the direct format, and set to 1 in the inverse format.
3. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
4. Determine whether an "L" is output from the TXDi pin by reading the port that shares a pin with the RXDi pin in the receive operation complete interrupt routine. When an "L" is output, wait for one clock cycle to read the UiRB register.

Figure 17.32 Register Settings in SIM Mode


(1) Transmit operation
TC
Internal transmit clock
TE bit in the UiC1 register
1
0
TI bit in the UiC1 register
1
0
(note 1)
Data is set in UiTB register
Data is transfer from UiTB register to UARTi transmit shift register
Start bit
Stop bit
TXDi output
"H"
"L"
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
Parity bit
Detect the level in interrupt routine
Parity error signal sent back from receiving device
"L" level is sent back from the SIM card since parity error has occurred
Signal line level(2)
TXEPT bit in the UiC0 register
IR bit in the SiTIC register
Set to 0 by an interrupt request acknowledgement or by a program
(2) Receive operation
Internal receive clock
RE bit in the UiC1 register
Transmit waveform sent by transmitting device
TXDi ouput
TXDi pin outputs "L" level since parity error has occurred
Signal line level(3)
RI bit in the UiC1 register
IR bit in the SiRIC register
Read from the UiRB register
Set to 0 by an interrupt request acknowledgement or by a program

The above applies under the following conditions:
- UiMR register: PRYE bit = 1 (parity enabled), STPS bit = 0 (1 stop bit)
- UiC1 register: UiIRS bit = 1 (transmit interrupt is generated at the transmit completion)

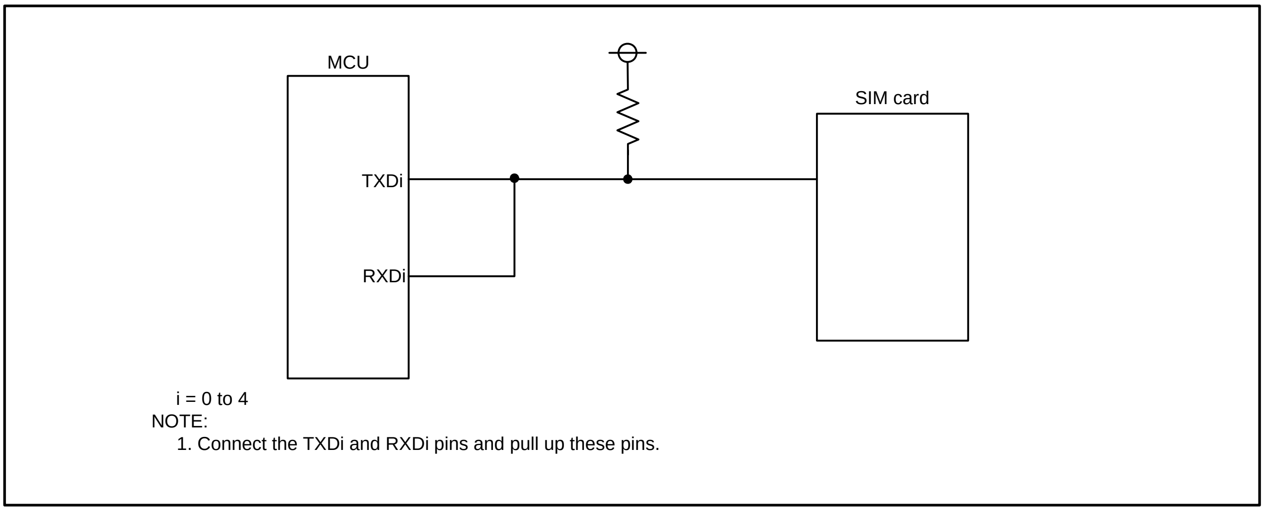
The above applies under the following conditions:
- UiMR register: PRYE bit = 1 (parity enabled), STPS bit = 0 (1 stop bit)

i = 0 to 4

$TC = \frac{16(m+ 1)}{fj}$ fj: f1, f8, f2n(4)
m: setting value of the UiBRG register (00 to FF)

NOTES:
1. Transmit operation is started when UiBRG overflows after data is set in the UiTB register in the indicated timing.
2. Because pins TXDi and RXDi are connected, a composite waveform, consisting of transmit waveform from the TXDi pin and parity error signal from the receiving device, is generated.
3. Because pins TXDi and RXDi are connected, a composite waveform consisting of transmit waveform from the transmitting device and parity error signal from the TXDi pin, is generated.
4. Bits CNT3 to CNT0 in the TCSPR register selects no division (n = 0) or divide-by-2n (n = 1 to 15).

Figure 17.33 SIM Interface Operation

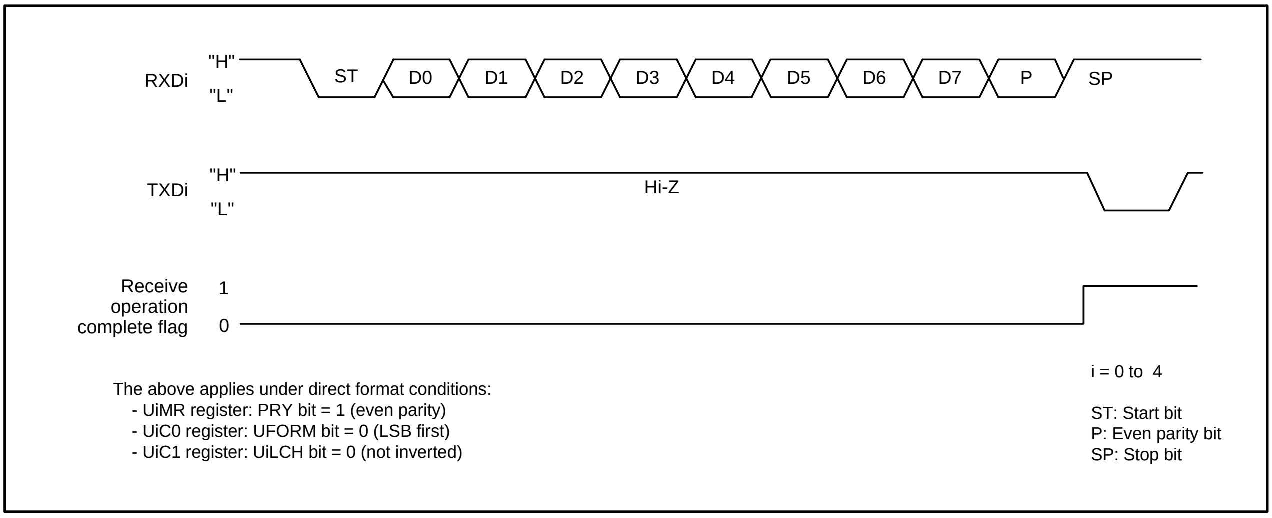


i = 0 to 4
NOTE:
1. Connect the TXDi and RXDi pins and pull up these pins.

Figure 17.34 SIM Interface Connection

17.1.6.1 Parity Error Signal Output Function

When the UiERE bit in the UiC1 register (i = 0 to 4) is set to 1 (error signal output), the parity error signal output is enabled. The parity error signal is output when a parity error is detected upon receiving data, and an "L" signal is output from the TXDi pin in the timing shown in Figure 17.35. If the UiRB register is read while a parity error signal is output, the PER bit in the UiRB register is set to 0 (no parity error) and the TXDi pin level becomes back to "H".
To determine whether the parity error signal is output or not, read the port that shares a pin with the RXDi pin in the transmission complete interrupt routine.



The above applies under direct format conditions:
- UiMR register: PRY bit = 1 (even parity)
- UiC0 register: UFORM bit = 0 (LSB first)
- UiC1 register: UiLCH bit = 0 (not inverted)

i = 0 to 4

ST: Start bit
P: Even parity bit
SP: Stop bit

Figure 17.35 Parity Error Signal Output Timing

17.1.6.2 Formats

17.1.6.2.1 Direct Format

When data is transmitted, data set in the UiTB register (i = 0 to 4) is transmitted with even parity, starting from D0. When data is received, received data is stored into the UiRB register, starting from D0. A parity error is determined with even parity.

Set the bits as follows to transmit or receive in the direct format.

- Set the PRYE bit in the UiMR register to 1 (parity enabled).
- Set the PRY bit in the UiMR register to 1 (even parity).
- Set the UFORM bit in the UiC0 register to 0 (LSB first).
- Set the UiLCH bit in the UiC1 register to 0 (not inverted).

17.1.6.2.2 Inverse Format

When data is transmitted, values set in the UiTB register are logically inverted. The data with the inverted values is transmitted with odd parity, starting from D7. When data is received, received data is logically inverted to be stored into the UiRB register, starting from D7. A parity error is determined with odd parity.

Set the bits as follows to transmit or receive in the inverse format.

- Set the PRYE bit to 1 (parity enabled).
- Set the PRY bit to 0 (odd parity).
- Set the UFORM bit to 1 (MSB first).
- Set the UiLCH bit to 1 (inverted).

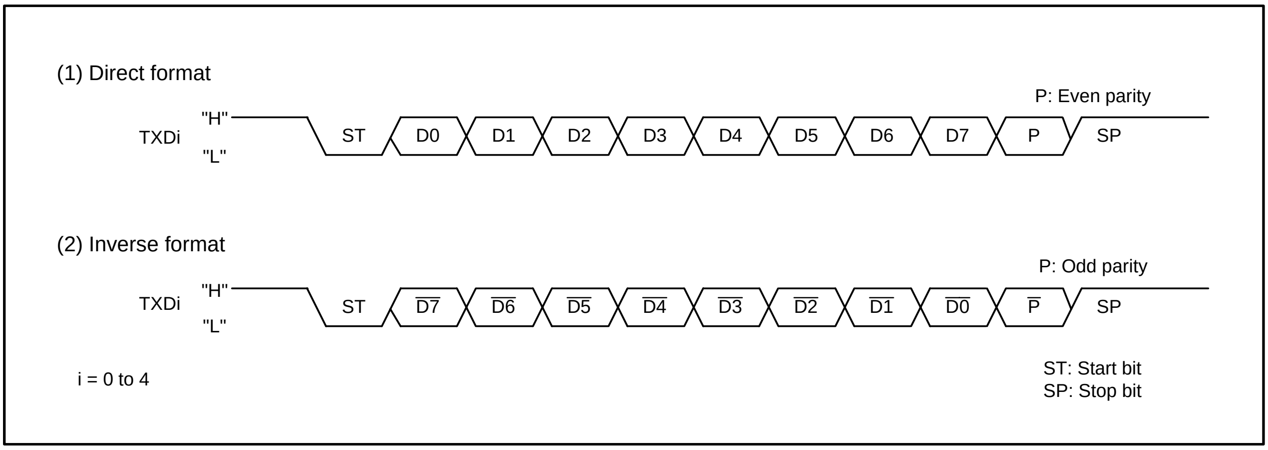


Figure 17.36 SIM Interface Formats

18. A/D Converter

NOTE

The 144-pin package is described as an example in this chapter.
Pins AN15_0 to AN15_7 are not provided in the 100-pin package.

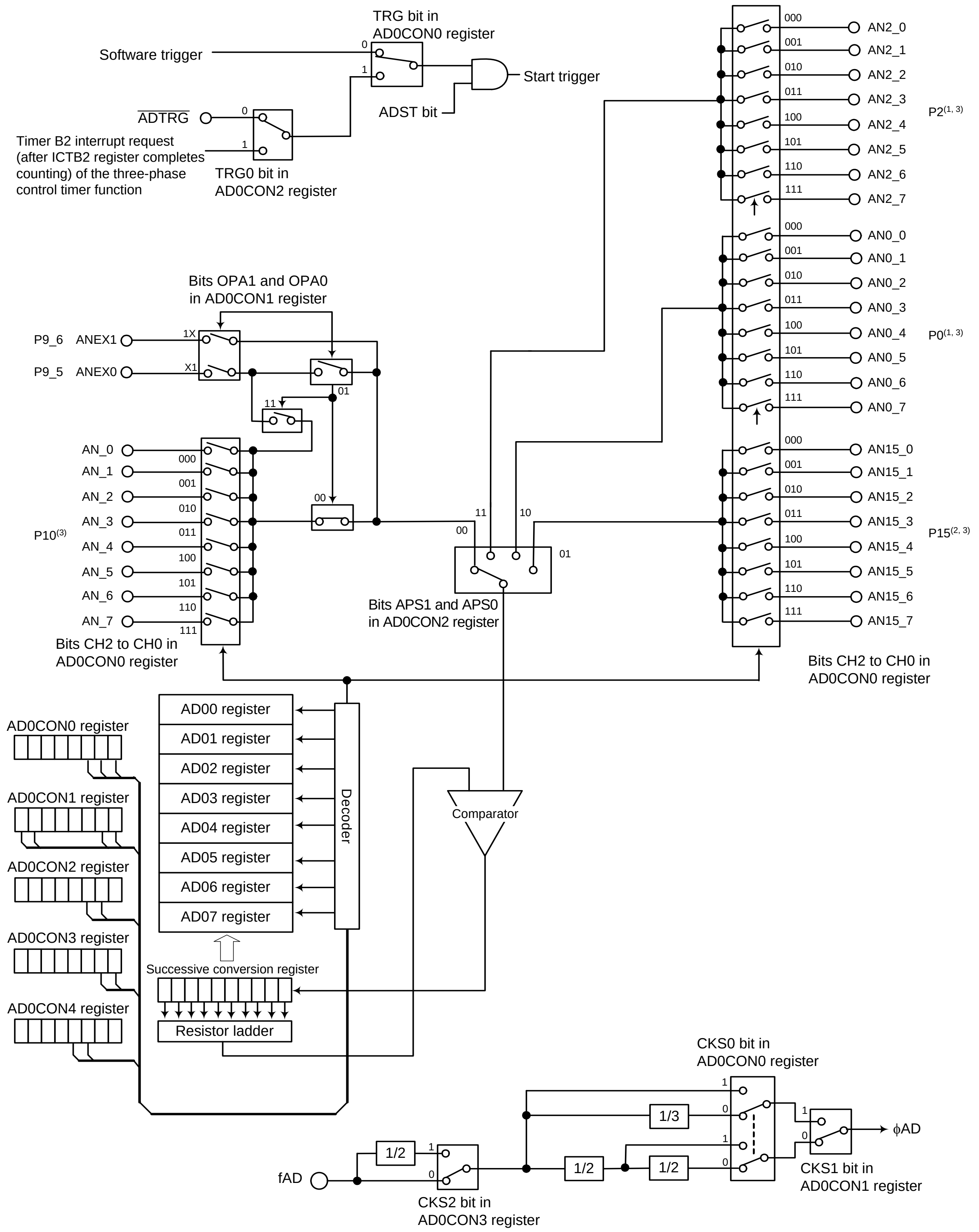
M32C/8B Group has one 10-bit successive approximation A/D converter with a capacitance coupled amplifier.
The results of A/D conversion are stored into the AD0i registers (i = 0 to 7) corresponding to the selected pins. When using DMAC operating mode, the conversion results are stored only into the AD00 register.
Table 18.1 lists specifications of the A/D converter. Figure 18.1 shows a block diagram of the A/D converter. Figures 18.2 to 18.6 show registers associated with the A/D converter.

Table 18.1 A/D Converter Specifications

Item	Specification
A/D conversion method	Successive approximation (with capacitance coupled amplifier)
Analog input voltage	0 V to AVCC (VCC1)
Operating clock ϕAD[1]	fAD, fAD/2, fAD/3, fAD/4, fAD/6, fAD/8
Resolution	Selectable from 8 bits or 10 bits
Operating modes	• One-shot mode • Repeat mode • Single sweep mode • Repeat sweep mode 0 • Repeat sweep mode 1 • Multi-port single sweep mode • Multi-port repeat sweep mode 0
Analog input pins[2]	144 pin package: 34 pins 8 pins each for AN (AN_0 to AN_7), AN0 (AN0_0 to AN0_7), AN2 (AN2_0 to AN2_7), and AN15 (AN15_0 to AN15_7) 2 extended input pins (ANEX0 and ANEX1) 100 pin package: 26 pins 8 pins each for AN (AN_0 to AN_7), AN0 (AN0_0 to AN0_7), AN2 (AN2_0 to AN2_7) 2 extended input pins (ANEX0 and ANEX1)
A/D conversion start condition	• Software trigger The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts). • External trigger (retrigger is enabled) When the falling edge is detected at the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1. • Hardware trigger (retrigger is enabled) Timer B2 interrupt request of the three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Conversion rate per pin	• Without sample and hold function 8-bit resolution: 49 ϕAD cycles, 10-bit resolution: 59 ϕAD cycles • With sample and hold function 8-bit resolution: 28 ϕAD cycles, 10-bit resolution: 33 ϕAD cycles

NOTES:

1. The ϕAD frequency must be 16 MHz or lower when VCC1 = 4.2 to 5.5 V.
   The ϕAD frequency must be 10 MHz or lower when VCC1 = 3.0 to 5.5 V.
   Without the sample and hold function, the ϕAD frequency must be 250 kHz or higher.
   With the sample and hold function, the ϕAD frequency must be 1 MHz or higher.
2. AVCC = VCC1 ≥ VCC2
   AD input (AN_0 to AN_7, AN15_0 to AN15_7, ANEX0, ANEX1) ≤ VCC1,
   AD input (AN0_0 to AN0_7, AN2_0 to AN2_7) ≤ VCC2



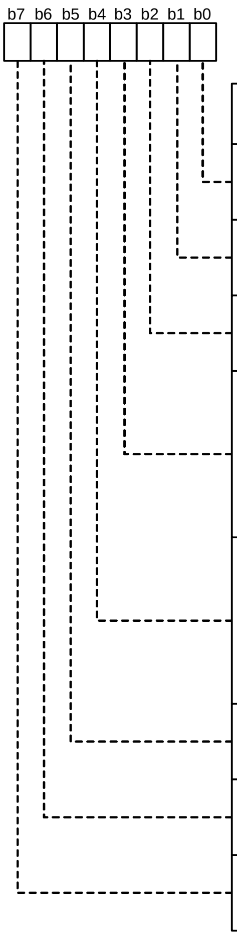
NOTES:

1. These pins can be used in single-chip mode only.
2. These pins are provided in the 144-pin package only.
3. AVCC = VCC1 ≥ VCC2, AD input (AN_0 to AN_7, AN15_0 to AN15_7, ANEX0, ANEX1) ≤ VCC1, AD input (AN0_0 to AN0_7, AN2_0 to AN2_7) ≤ VCC2

Figure 18.1 A/D Converter Block Diagram

A/D0 Control Register 0[1]

Symbol	Address	After Reset
AD0CON0	0396h	00h



Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bits[2, 3]	b2 b1 b0 0 0 0: ANi_0 0 0 1: ANi_1 0 1 0: ANi_2 0 1 1: ANi_3 1 0 0: ANi_4 1 0 1: ANi_5 1 1 0: ANi_6 1 1 1: ANi_7 (i = none, 0, 2, 15)	RW
CH1			RW
CH2			RW
MD0	A/D operating mode select bits 0[2]	When the MSS bit in the AD0CON3 register = 0 b4 b3 0 0: One-shot mode 0 1: Repeat mode 1 0: Single sweep mode 1 1: Repeat sweep mode 0, repeat sweep mode 1	RW
MD1		When the MSS bit in the AD0CON3 register = 1 b4 b3 0 0: } Do not set to these values. 0 1: } 1 0: Multi-port single sweep mode 1 1: Multi-port repeat sweep mode 0	RW
TRG	Trigger select bit	0: Software trigger 1: External trigger, hardware trigger[4]	RW
ADST	A/D conversion start bit	0: A/D conversion stops 1: A/D conversion starts[4]	RW
CKS0	Frequency select bit 0	(Note 5)	RW

NOTES:
1. If the AD0CON0 register is rewritten during A/D conversion, the conversion result will be incorrect.
2. Analog input pins must be configured again after an A/D operating mode is changed.
3. Bits CH2 to CH0 are enabled in one-shot mode and repeat mode.
4. To set the TRG bit to 1, select a trigger source using the TRG0 bit in the AD0CON2 register. Then, set the ADST bit to 1 after the TRG bit is set to 1.
5. ϕAD frequency must be 16 MHz or lower when VCC1 = 4.2 to 5.5V.
ϕAD frequency must be 10 MHz or lower when VCC1 = 3.0 to 5.5V.
ϕAD is selected by the combination of the CKS0 bit, the CKS1 in the AD0CON1 register, and the CKS2 bit in the AD0CON3 register.

CKS2 bit in AD0CON3 register	CKS0 bit in AD0CON0 register	CKS1 bit in AD0CON1 register	ϕAD
0	0	0	fAD divided by 4
		1	fAD divided by 3
	1	0	fAD divided by 2
		1	fAD
1	0	0	fAD divided by 8
		1	fAD divided by 6

Figure 18.2 AD0CON0 Register

A/D0 Control Register 1(1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
AD0CON1	0397h	00h

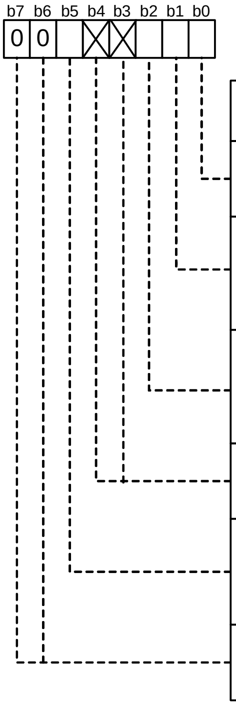
Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bits(2)	Single sweep mode and repeat sweep mode 0 b1 b0 0 0: ANi_0, ANi_1 (i = none, 0, 2, 15) 0 1: ANi_0 to ANi_3 1 0: ANi_0 to ANi_5 1 1: ANi_0 to ANi_7 Repeat sweep mode 1(3) b1 b0 0 0: ANi_0 0 1: ANi_0, ANi_1 1 0: ANi_0 to ANi_2 1 1: ANi_0 to ANi_3 Multi-port single sweep mode and multi-port repeat sweep mode 0(4) Set to 11b.	RW
SCAN1			RW
MD2	A/D operating mode select bit 1(4)	0: Other than repeat sweep mode 1 1: Repeat sweep mode 1	RW
BITS	Resolution select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	(Note 5)	RW
VCUT	VREF connection bit(8)	0: VREF not connected(7) 1: VREF connected	RW
OPA0	Extended input pin function select bits(4, 6)	b7 b6 0 0: ANEX0 and ANEX1 are not used 0 1: Signal applied to ANEX0 is A/D converted 1 0: Signal applied to ANEX1 is A/D converted 1 1: External op-amp connection	RW
OPA1			RW

NOTES:

1. If the AD0CON1 register is rewritten during A/D conversion, the conversion result will be incorrect.
2. Bits SCAN1 and SCAN0 are enabled in single sweep mode, repeat sweep mode 0, 1, multi-port single sweep mode, and multi-port repeat sweep mode 0.
3. These are prioritized pins used for A/D conversion when the MD2 bit is set to 1.
4. When the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode used);
   -set bits SCAN1 and SCAN0 to 11b
   -set the MD2 bit to 0
   -set bits OPA1 and OPA0 to 00b.
5. Refer to the note for the CKS0 bit in the AD0CON0 register.
6. Bits OPA1 and OPA0 can be set to 01b or 10b in one-shot mode and repeat mode. Set these bits to 00b or 11b in other modes.
7. Do not set the VCUT bit to 0 during A/D conversion. Even if the VCUT bit is set to 0, VREF remains connected to the D/A converter.
8. When the VCUT bit is set to 1 from 0, wait for 1 µs or more to start the A/D conversion.

Figure 18.3 AD0CON1 Register

A/D0 Control Register 2[1]



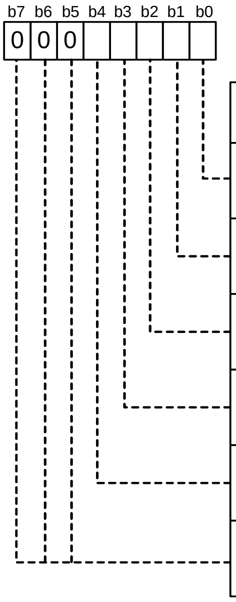
Symbol	Address	After Reset
AD0CON2	0394h	XX0X X000b

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
APS0	Analog input port select bits[3]	When the MSS bit in the AD0CON3 register = 0 b2 b1 0 0: AN_0 to AN_7, ANEX0, ANEX1 0 1: AN15_0 to AN15_7[2] 1 0: AN0_0 to AN0_7 1 1: AN2_0 to AN2_7 When the MSS bit in the AD0CON3 register = 1 Set to 01b.	RW
APS1			RW
– (b4-b3)	Unimplemented. Write 0. Read as undefined value.		–
TRG0	External trigger source select bit	0: $\overline{\text{ADTRG}}$ selected 1: Timer B2 interrupt request of the three-phase motor control timer function (after the ICTB2 register completes counting) selected	RW
– (b7-b6)	Reserved bits	Set to 0. Read as undefined value.	RW

NOTES:
1. If the AD0CON2 register is rewritten during A/D conversion, the conversion result will be incorrect.
2. In the 100-pin package, do not set to 01b.
3. Set to 00b or 01b in memory expansion mode and microprocessor mode.

Figure 18.4 AD0CON2 Register

A/D0 Control Register 3[1, 2]



Symbol	Address	After Reset
AD0CON3	0395h	XXXX X000b

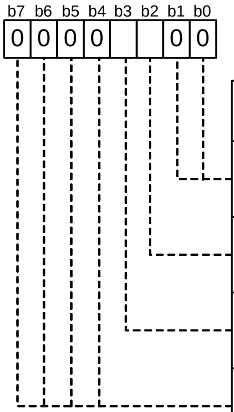
Bit Symbol	Bit Name	Function	RW
DUS	DMAC operating mode select bit	0: DMAC operating mode not used 1: DMAC operating mode used	RW
MSS	Multi-port sweep mode select bit	0: Multi-port sweep mode not used 1: Multi-port sweep mode used[3]	RW
CKS2	Frequency select bit 2	(Note 4)	RW
MSF0	Multi-port sweep status flags[5]	b4 b3 0 0: AN_0 to AN_7 0 1: AN15_0 to AN15_7 1 0: AN0_0 to AN0_7 1 1: AN2_0 to AN2_7	RO
MSF1			RO
– (b7-b5)	Reserved bits	Set to 0. Read as undefined value.	RW

NOTES:

1. If the AD0CON3 register is rewritten during A/D conversion, the conversion result will be incorrect.
2. The AD0CON3 register may return an incorrect value if read during A/D conversion. It must be read or written after the A/D conversion stops.
3. When the MSS bit is set to 1;
   -set the DUS bit to 1 and configure DMAC.
   -set bits MD1 and MD0 in the AD0CON0 register to 10b or 11b.
   -set bits SCAN1 and SCAN0 in the AD0CON1 register to 11b, the MD2 bit to 0, bits OPA1 and OPA0 to 00b.
   -set bits APS1 and APS0 in the AD0CON2 register to 01b.
   -set bits MPS11 and MPS10 to 01b, 10b, or 11b.
4. Refer to the note for the CKS0 bit in the AD0CON0 register.
5. Bits MSF1 and MSF0 are enabled when the MSS bit is set to 1. When the MSS bit is set to 0, a read from these bits returns an undefined value.

Figure 18.5 AD0CON3 Register

A/D0 Control Register 4[1]

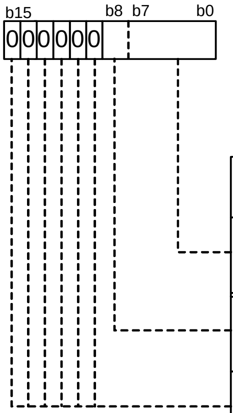


Symbol	Address	After Reset
AD0CON4	0392h	XXXX 00XXb

Bit Symbol	Bit Name	Function	RW
– (b1-b0)	Reserved bits	Set to 0. Read as undefined value.	RW
MPS10	Multi-port sweep port select bits[2, 3]	b3 b2 0 0: (Note 4) 0 1: AN_0 to AN_7, AN15_0 to AN15_7 1 0: AN_0 to AN_7, AN0_0 to AN0_7 1 1: AN_0 to AN_7, AN2_0 to AN2_7	RW
MPS11			RW
– (b7-b4)	Reserved bits	Set to 0. Read as undefined value.	RW

NOTES:
1. If the AD0CON4 register is rewritten during A/D conversion, the conversion result will be incorrect.
2. Do not set bits MPS11 and MPS10 to 01b in the 100-pin package.
3. Do not set bits MPS11 and MPS10 to 10b or 11b in memory expansion mode or microprocessor mode.
4. When the MSS bit in the AD0CON3 register is set to 0 (multi-port sweep mode not used), set bits MPS11 and MPS10 to 00b. When the MSS bit is set to 1 (multi-port sweep mode used), set bits MPS11 and MPS10 to other than 00b.

A/D0 Register i[1, 2, 3, 4] (i = 0 to 7)



Symbol	Address	After Reset
AD00	0381h - 0380h	00000000 XXXXXXXXb
AD01 to AD03	0383h - 0382h, 0385h - 0384h, 0387h - 0386h	00000000 XXXXXXXXb
AD04 to AD06	0389h - 0388h, 038Bh - 038Ah, 038Dh - 038Ch	00000000 XXXXXXXXb
AD07	038Fh - 038Eh	00000000 XXXXXXXXb

Function	RW
8 low-order bits of A/D conversion result	RO
In 10-bit mode: 2 high-order bits of A/D conversion result In 8-bit mode: Read as 0.	RO
Reserved bits. Read as 0.	RO

NOTES:
1. When the AD0i register is read by a program in DMAC operating mode, the conversion result is incorrect.
2. If the next A/D conversion result is stored before reading the previous result in the AD0i register, the result will be incorrect.
3. Only AD00 register is enabled in DMAC operating mode. The contents of other registers are undefined.
4. When using both DMAC operating mode and 10-bit mode, select a 16-bit transfer for DMAC.

Figure 18.6 AD0CON4 Register, AD00 to AD07 Registers

If analog input shares the pin with other peripheral function inputs, a through current may flow to the peripheral function inputs when an intermediate voltage is applied to the pin. To prevent through current, set the control bit for the corresponding pin to 1, and other peripheral inputs are disconnected. Table 18.2 lists settings of an analog input pin.

Table 18.2 Analog Input Pin Setting

Port	Function	Control Bit	
		PSC Register	PSL3 Register
P9_5	ANEX0	–	PSL3_5 = 1
P9_6	ANEX1	–	PSL3_6 = 1
P10_4	AN_4	PSC_7 = 1	–
P10_5	AN_5		–
P10_6	AN_6		–
P10_7	AN_7		–

18.1 Mode Descriptions

The A/D converter has seven different modes. Table 18.3 lists settings for these modes.

Table 18.3 Mode Settings

Mode	AD0CON0 register		AD0CON1 register	AD0CON3 register	
	MD1 bit	MD0 bit	MD2 bit	MSS bit	DUS bit
One-shot mode	0	0	0	0	0 or 1
Repeat mode	0	1	0	0	0 or 1
Single sweep mode	1	0	0	0	0 or 1
Repeat sweep mode 0	1	1	0	0	0 or 1
Repeat sweep mode 1	1	1	1	0	0 or 1
Multi-port single sweep mode	1	0	0	1	1
Multi-port repeat sweep mode 0	1	1	0	1	1

18.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is converted to a digital code once. Table 18.4 lists specifications of one-shot mode.

Table 18.4 One-Shot Mode Specifications

Item	Specification
Function	Analog voltage applied to a selected pin is converted once
Analog input pins	Select one pin from AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, or ANEX1. The following register settings determine which pin is used: • Bits CH2 to CH0 in the AD0CON0 register • Bits OPA1 and OPA0 in the AD0CON1 register • Bits APS1 and APS0 in the AD0CON2 register
Start Condition	Software trigger is selected (TRG bit in the AD0CON0 register = 0): • The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts) External trigger, hardware trigger is selected (TRG bit = 1): • TRG0 bit in the AD0CON2 register = 0 The falling edge is detected on the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1 • TRG0 bit = 1 Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Stop condition	• A/D conversion is completed (the ADST bit becomes 0 when software trigger is selected). • Set the ADST bit to 0 by a program (A/D conversion stops).
Interrupt request generation timing	When the A/D conversion is completed
Reading A/D conversion result	• DMAC operating mode is not used (DUS bit in the AD0CON3 register = 0): Read the AD0j register (j = 0 to 7) corresponding to a selected pin by a program. • DMAC operating mode is used (DUS bit = 1): A/D conversion result is stored into the AD00 register after A/D conversion is completed. Then, DMAC transfers the data from the AD00 register to a given memory space. (Refer to **13. DMAC** for DMAC settings)

18.1.2 Repeat Mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code.
Table 18.5 lists specifications of repeat mode.

Table 18.5 Repeat Mode Specifications

Item	Specification
Function	Analog voltage applied to a selected pin is repeatedly converted
Analog input pins	Select one pin from AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, or ANEX1 The following register settings determine which pin is used: • Bits CH2 to CH0 in the AD0CON0 register • Bits OPA1 and OPA0 in the AD0CON1 register • Bits APS1 and APS0 in the AD0CON2 register
Start condition	Software trigger is selected (TRG bit in the AD0CON0 register = 0): • The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts) External trigger, hardware trigger is selected (TRG bit = 1): • TRG0 bit in the AD0CON2 register = 0 The falling edge is detected on the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1 • TRG0 bit = 1 Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Stop condition	Set the ADST bit to 0 (A/D conversion stops)
Interrupt request generation timing	• DMAC operating mode is not used (DUS bit in the AD0CON3 register = 0): Interrupt request is not generated. • DMAC operating mode is used (DUS bit = 1): Interrupt request is generated every time each A/D conversion is completed.
Reading A/D conversion result	• DMAC operating mode is not used (DUS bit = 0): Read the AD0j register (j = 0 to 7) corresponding to a selected pin by a program. • DMAC operating mode is used (DUS bit = 1): A/D conversion result is stored into the AD00 register after A/D conversion is completed. Then, DMAC transfers the data from the AD00 register to a given memory space. (Refer to **13. DMAC** for DMAC settings)

18.1.3 Single Sweep Mode

In single sweep mode, analog voltage applied to multiple selected pins is converted to a digital code once for each pin.
Table 18.6 lists specifications of single sweep mode.

Table 18.6 Single Sweep Mode Specifications

Item	Specification
Function	Analog voltage applied to selected pins is converted once for each pin
Analog input pins	Select one of the following. • 2 pins (ANi_0 and ANi_1) (i = none, 0, 2, 15) • 4 pins (ANi_0 to ANi_3) • 6 pins (ANi_0 to ANi_5) • 8 pins (ANi_0 to ANi_7) The following register settings determine which pins are used: • Bits SCAN1 and SCAN0 in the AD0CON1 register • Bits APS1 and APS0 in the AD0CON2 register
Start condition	Software trigger is selected (TRG bit in the AD0CON0 register = 0): • The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts) External trigger, hardware trigger is selected (TRG bit = 1): • TRG0 bit in the AD0CON2 register = 0 The falling edge is detected on the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1 • TRG0 bit = 1 Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Stop condition	• A sequence of A/D conversions is completed (the ADST bit becomes 0 when software trigger is selected) • Set the ADST bit to 0 by a program (A/D conversion stops)
Interrupt request generation timing	• DMAC operating mode is not used (DUS bit in the AD0CON3 register = 0): Interrupt request is generated after a sequence of A/D conversions is completed. • DMAC operating mode is used (DUS bit = 1): Interrupt request is generated every time each A/D conversion is completed
Reading A/D conversion result	• DMAC operating mode is not used (DUS bit = 0): Read the AD0j register (j = 0 to 7) corresponding to a selected pin by a program. • DMAC operating mode is used (DUS bit = 1): A/D conversion result is stored into the AD00 register after A/D conversion is completed. Then, DMAC transfers the data from the AD00 register to a given memory space. (Refer to **13. DMAC** for DMAC settings)

18.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltage applied to multiple selected pins is repeatedly converted to a digital code.

Table 18.7 lists specifications of repeat sweep mode 0.

Table 18.7 Repeat Sweep Mode 0 Specifications

Item	Specification
Function	Analog voltage applied to selected pins is repeatedly converted
Analog input pins	Select one of the following. • 2 pins (ANi_0 and ANi_1) (i = none, 0, 2, 15) • 4 pins (ANi_0 to ANi_3) • 6 pins (ANi_0 to ANi_5) • 8 pins (ANi_0 to ANi_7) The following register settings determine which pins are used: • Bits SCAN1 and SCAN0 in the AD0CON1 register • Bits APS1 and APS0 in the AD0CON2 register
Start condition	Software trigger is selected (TRG bit in the AD0CON0 register = 0): • The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts) External trigger, hardware trigger is selected (TRG bit = 1): • TRG0 bit in the AD0CON2 register = 0 The falling edge is detected on the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1 • TRG0 bit = 1 Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Stop condition	Set the ADST bit to 0 (A/D conversion stops)
Interrupt request generation timing	• DMAC operating mode is not used (DUS bit in the AD0CON3 register = 0): Interrupt request is not generated • DMAC operating mode is used (DUS bit = 1): Interrupt request is generated every time each A/D conversion is completed
Reading A/D conversion result	• DMAC operating mode is not used (DUS bit = 0): Read the AD0j register (j = 0 to 7) corresponding to a selected pin by a program. • DMAC operating mode is used (DUS bit = 1): A/D conversion result is stored into the AD00 register after A/D conversion is completed. Then, DMAC transfers the data from the AD00 register to a given memory space. (Refer to **13. DMAC** for DMAC settings)

18.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage applied to eight pins, prioritizing one to four pins, is repeatedly converted to a digital code.

Table 18.8 lists specifications of repeat sweep mode 1.

Table 18.8 Repeat Sweep Mode 1 Specification

Item	Specification
Function	Analog voltage applied to 8 selected pins, prioritizing one to four pins, is repeatedly converted.
Analog input pins	ANi_0 to ANi_7 (8 pins are selected from these pins) (i = none, 0, 2, 15)
Prioritized pins	Select one of the following. • single pin (ANi_0) • 2 pins (ANi_0 and ANi_1) • 3 pins (ANi_0 to ANi_2) • 4 pins (ANi_0 to ANi_3) The following register settings determine which pins are used: • Bits SCAN1 and SCAN0 in the AD0CON1 register • Bits APS1 and APS0 in the AD0CON2 register
Start condition	Software trigger is selected (TRG bit in the AD0CON0 register = 0): • The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts) External trigger, hardware trigger is selected (TRG bit = 1): • TRG0 bit in the AD0CON2 register = 0 The falling edge is detected on the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1 • TRG0 bit = 1 Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Stop condition	Set the ADST bit is set to 0 (A/D conversion stops)
Interrupt request generation timing	• DMAC operating mode is not used (DUS bit in the AD0CON3 register = 0): Interrupt request is not generated. • DMAC operating mode is used (DUS bit = 1): Interrupt request is generated every time each A/D conversion is completed.
Reading A/D conversion result	• DMAC operating mode is not used (DUS bit = 0): Read the AD0j register (j = 0 to 7) corresponding to a selected pin by a program. • DMAC operating mode is used (DUS bit = 1): A/D conversion result is stored into the AD00 register after A/D conversion is completed. Then, DMAC transfers the data from the AD00 register to a given memory space. (Refer to **13. DMAC** for DMAC settings)

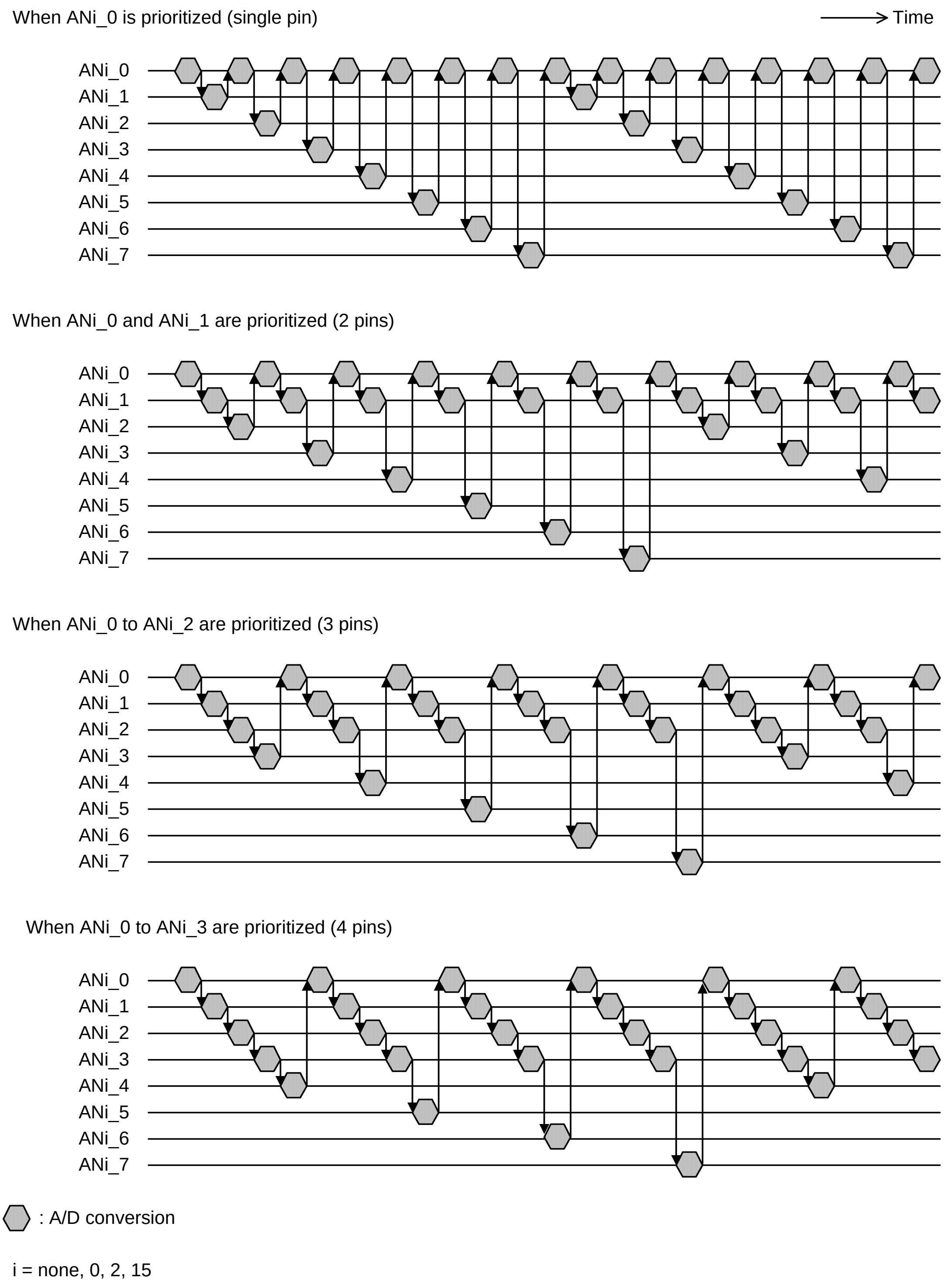

When ANi_0 is prioritized (single pin)
Time
ANi_0
ANi_1
ANi_2
ANi_3
ANi_4
ANi_5
ANi_6
ANi_7
When ANi_0 and ANi_1 are prioritized (2 pins)
When ANi_0 to ANi_2 are prioritized (3 pins)
When ANi_0 to ANi_3 are prioritized (4 pins)
: A/D conversion
i = none, 0, 2, 15

Figure 18.7 Transition Diagram of Pins used in A/D Conversion in Repeat Sweep Mode 1

18.1.6 Multi-Port Single Sweep Mode

In multi-port single sweep mode, analog voltage applied to 16 selected pins is converted to a digital code once for each pin. Set the DUS bit in the AD0CON3 register to 1 (DMAC operating mode used).
Table 18.9 lists specifications of multi-port single sweep mode.

Table 18.9 Multi-Port Single Sweep Mode Specifications

Item	Specification
Function	Analog voltage applied to the 16 selected pins is repeatedly converted once for each pin in the following order: AN_0 to AN_7 → ANi_0 to ANi_7 (i = 0, 2, 15)
Analog input pins	Select one of the following. • AN_0 → AN_1 → · · · → AN_7 → AN0_0 → AN0_1→ · · · → AN0_7 • AN_0 → AN_1 → · · · → AN_7 → AN2_0 → AN2_1→ · · · → AN2_7 • AN_0 → AN_1 → · · · → AN_7 → AN15_0 → AN15_1→ · · · → AN15_7 The following register settings determine which pins are used: Bits MPS11 and MPS10 in the AD0CON4 register
Start condition	Software trigger is selected (TRG bit in the AD0CON0 register = 0): • The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts) External trigger, hardware trigger is selected (TRG bit = 1): • TRG0 bit in the AD0CON2 register = 0 The falling edge is detected on the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1 • TRG0 bit = 1 Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Stop condition	• A sequence of A/D conversions is completed (the ADST bit becomes 0 when software trigger is selected) • Set the ADST bit to 0 by a program (A/D conversion stops)
Interrupt request generation timing	An interrupt request is generated every time each A/D conversion is completed (Set the DUS bit in the AD0CON3 register to 1)
Reading A/D conversion result	A/D conversion result is stored into the AD00 register after A/D conversion is completed. Then, DMAC transfers the data from the AD00 register to a given memory space. Refer to **13. DMAC** for DMAC settings. (Set the DUS bit in the AD0CON3 register to 1)

18.1.7 Multi-Port Repeat Sweep Mode 0

In multi-port repeat sweep mode 0, analog voltage applied to 16 selected pins is repeatedly converted to a digital code. Set the DUS bit in the AD0CON3 register to 1 (DMAC operating mode used).
Table 18.10 lists specifications of multi-port repeat sweep mode 0.

Table 18.10 Multi-Port Repeat Sweep Mode 0 Specifications

Item	Specification
Function	Analog voltage applied to the 16 selected pins is repeatedly converted in the following order: AN_0 to AN_7 → ANi_0 to ANi_7 (i = 0, 2, 15)
Analog input pins	Select one of the following. • AN_0 → AN_1 → ··· → AN_7 → AN0_0 → AN0_1→ ··· → AN0_7 • AN_0 → AN_1 → ··· → AN_7 → AN2_0 → AN2_1→ ··· → AN2_7 • AN_0 → AN_1 → ··· → AN_7 → AN15_0 → AN15_1→ ··· → AN15_7 The following register settings determine which pins are used: Bits MPS11 and MPS10 in the AD0CON4 register
Start condition	Software trigger is selected (TRG bit in the AD0CON0 register = 0): • The ADST bit in the AD0CON0 register is set to 1 (A/D conversion starts) External trigger, hardware trigger is selected (TRG bit = 1): • TRG0 bit in the AD0CON2 register = 0 The falling edge is detected on the $\overline{\text{ADTRG}}$ pin after the ADST bit is set to 1 • TRG0 bit = 1 Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting) is generated after the ADST bit is set to 1.
Stop condition	Set the ADST bit is set to 0 (A/D conversion stops)
Interrupt request generation timing	An interrupt request is generated every time each A/D conversion is completed (Set the DUS bit in the AD0CON3 register to 1)
Reading A/D conversion result	A/D conversion result is stored into the AD00 register after A/D conversion is completed. Then, DMAC transfers the data from the AD00 register to a given memory space. Refer to **13. DMAC** for DMAC settings (Set the DUS bit in the AD0CON3 register to 1)

18.2 Functions

18.2.1 Resolution

The BITS bit in the AD0CON1 register determines the resolution. When the BITS bit is set to 1 (10-bit mode), the A/D conversion result is stored into bits 9 to 0 in the AD0i register (i = 0 to 7). When the BITS bit is set to 0 (8-bit mode), the A/D conversion result is stored into bits 7 to 0 in the AD0i register.

18.2.2 Sample and Hold

When the SMP bit in the AD0CON2 register is set to 1 (with sample and hold), the A/D conversion rate per pin increases to 28 ϕAD cycles for 8-bit resolution and 33 ϕAD cycles for 10-bit resolution. The sample and hold function is available in all operating modes. Start A/D conversion after selecting whether the sample and hold circuit is used or not.

18.2.3 Trigger Select Function

The TRG bit in the AD0CON0 register and the TRG0 bit in the AD0CON2 register determine a trigger to start A/D conversion. Table 18.11 lists setting values for the trigger select function.

Table 18.11 Trigger Select Function Setting Values

Bit and Setting		Trigger
AD0CON0 Register	AD0CON2 Register	
TRG = 0	–	Software trigger A/D conversion starts when the ADST bit in the AD0CON0 register is set to 1 by a program
TRG = 1[1]	TRG0 = 0	External trigger[2] Falling edge of a signal applied to $\overline{\text{ADTRG}}$
	TRG0 = 1	Hardware trigger[2] Timer B2 interrupt request of three-phase motor control timer function (after the ICTB2 register completes counting)

NOTES:

1. A/D conversion starts when the ADST bit is set to 1 (A/D conversion starts) and a trigger is input.
2. If an external trigger or a hardware trigger (retrigger) is input during A/D conversion, the sequence of A/D conversions in progress is aborted and starts over from the beginning.

18.2.4 DMAC Operating Mode

DMAC operating mode is available in all operating modes. To select multi-port single sweep mode or multi-port repeat sweep mode 0, DMAC operating mode must be used. When the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode used), all A/D conversion results are stored into the AD00 register. DMAC transfers the result from the AD00 register to a given memory space every time A/D conversion on a single pin is completed. 8-bit DMA transfer must be selected for 8-bit resolution and 16-bit DMA transfer for 10-bit resolution. Refer to **13. DMAC** for DMAC instructions.

When using DMAC operating mode in single sweep mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, or multi-port repeat sweep mode 0, do not input an external retrigger or hardware retrigger. If a retrigger is input, the sequence of A/D conversions in progress is aborted and starts over from the ANi_0 pin (i = none, 0, 2, 15). As a result, a pin and the conversion result of the pin transferred to the RAM do not correspond to each other.

18.2.5 Extended Analog Input Pins

In one-shot mode and repeat mode, the ANEX0 pin or ANEX1 pin can be used as the analog input pin. These pins can be selected using bits OPA1 and OPA0 in the AD0CON1 register. The A/D conversion result for ANEX0 input is stored into the AD00 register, and for ANEX1 input into the AD01 register. Both results are stored into the AD00 register when the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode used).

Set bits APS1 and APS0 in the AD0CON2 register to 00b (AN_0 to AN_7, ANEX0, ANEX1) and the MSS bit in the AD0CON3 register to 0 (multi-port sweep mode not used).

18.2.6 External Operating Amplifier (Op-Amp) Connection Mode

In external op-amp connection mode, multiple analog voltage can be amplified by one external op-amp using extended analog input pins, ANEX0 and ANEX1.
When bits OPA1 and OPA0 are set to 11b (external op-amp connection), voltage applied to pins AN_0 to AN_7 are output from the ANEX0. Amplify this output signal by external op-amp and apply it to the ANEX1.
Analog voltage applied to ANEX1 is converted to a digital code and the A/D conversion result is stored into the corresponding AD0i register (i = 0 to 7). The A/D conversion rate varies depending on the response characteristics of the external op-amp. Do not connect the ANEX0 pin to the ANEX1 pin directly.
Set bits APS1 and APS0 in the AD0CON2 register to 00b (AN_0 to AN_7, ANEX0, ANEX1).
Figure 18.8 shows a connection example of external op-amp connection mode.

Table 18.12 Extended Analog Input Pin Settings

AD0CON1 Register		ANEX0 Function	ANEX1 Function
OPA1 Bit	OPA0 Bit		
0	0	Not used	Not used
0	1	P9_5 as an analog input	Not used
1	0	Not used	P9_6 as an analog input
1	1	Output to external op-amp	Input from external op-amp

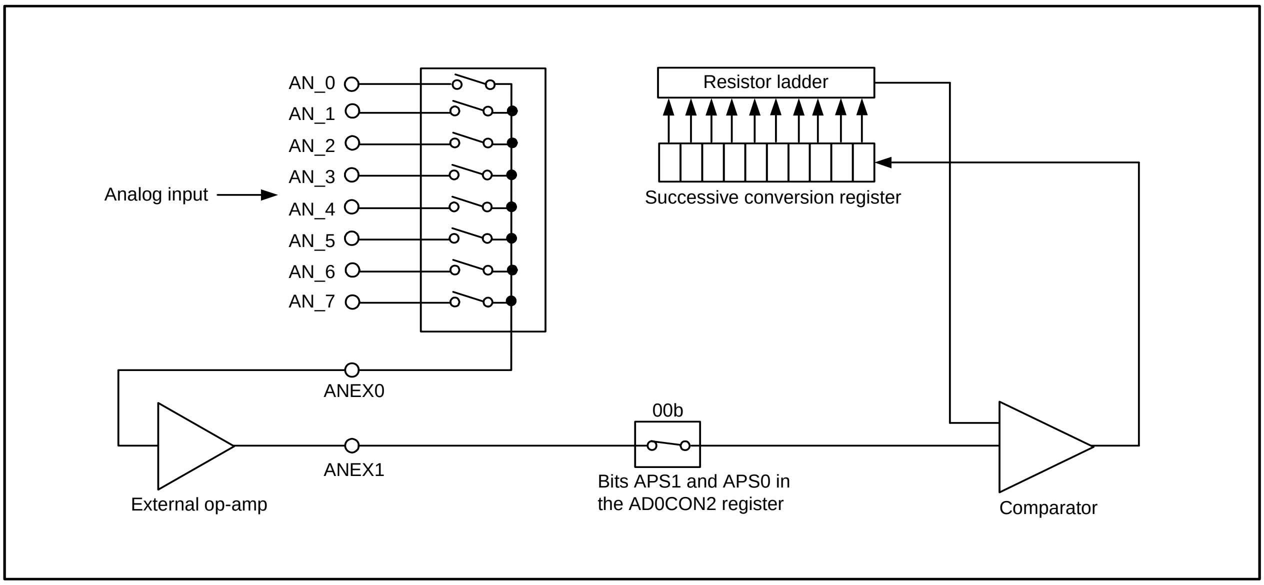


Figure 18.8 Connection Example in External Op-Amp Connection Mode

18.2.7 Power Consumption Reduce Function

When not using the A/D converter, the VCUT bit in the AD0CON1 register can disconnect the resistor ladder of the A/D converter from the reference voltage input pin (VREF). As a result, power consumption can be reduced by shutting off any current flow into the resistor ladder from the VREF pin.
When using the A/D converter, set the VCUT bit to 1 (VREF connected) prior to setting the ADST bit in the AD0CON0 register to 1 (A/D conversion starts).
Do not set the VCUT bit to 0 (VREF not connected) during A/D conversion.
Even if the VCUT bit is set to 0, VREF remains connected to the D/A converter.

18.3 Read from the AD0i Register (i = 0 to 7)

Use the following procedure to read the AD0i register by a program.

- In one-shot mode and single sweep mode:
  Ensure that the A/D conversion is completed before reading the corresponding AD0i register. The IR bit in the AD0IC register becomes 1 when the A/D conversion is completed.
- In repeat mode, repeat sweep mode 0, and repeat sweep mode 1:
  Read the AD0i register after setting the CPU clock as follows.
  (1) Set the CM07 bit in the CM0 register to 0 (clock selected by the CM21 bit divided by the MCD register).
  (2) Set the MCD register to 12h (no division).

18.4 Output Impedance of Sensor Equivalent Circuit under A/D Conversion

To take full advantage of the A/D converter performance, Internal capacitor (C) charge shown in Figure 18.9 must be completed within the specified period (T) as sampling time. Output impedance of the sensor equivalent circuit (R0) is determined by the following equation:

$$\mathrm{VC} = \mathrm{VIN}\left\{1 - e^{-\frac{1}{C(R0+R)}t}\right\}$$

When t = T,

$$\mathrm{VC} = \mathrm{VIN} - \frac{X}{Y}\mathrm{VIN} = \mathrm{VIN}\left(1 - \frac{X}{Y}\right)$$

$$e^{-\frac{1}{C(R0+R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)}T = \ln\frac{X}{Y}$$

$$R0 = -\frac{T}{C\ln\frac{X}{Y}} - R$$

where:

VC = Internal capacitor voltage
R = Internal resistance of the MCU
X = Accuracy (error) of the A/D converter
Y = Resolution (1024 in 10-bit mode, and 256 in 8-bit mode)

Figure 18.9 shows a connection example of analog input pin and external sensor equivalent circuit.
In the following example, the impedance R0 is obtained from the equation above when VC changes from 0 to VIN-(1/1024)VIN within the time (T), if the difference between VIN and VC becomes 1LSB. (1/1024) means that A/D accuracy drop, due to insufficient capacitor charge, is held to 1LSB at time of A/D conversion in the 10-bit mode. Actual error, however, is the value of absolute accuracy added to 1LSB.
When ϕAD = 10 MHz, T = 0.3 μs in A/D conversion with the sample and hold function. Output impedance (R0) enough to complete charging the capacitor (C) within the time (T) is determined by the following equation:

Using T = 0.3 μs, R = 2.0 kΩ, C = 8.6 pF, X = 1, Y = 1024,

$$R0 = -\frac{0.3\times10^{-6}}{8.6\times10^{-12}\cdot\ln\frac{1}{1024}} - 2.0\times10^{3} \cong 3.0\times10^{3}\Omega$$

Thus, the allowable output impedance R0 of the sensor equivalent circuit, making the accuracy (error) 1LSB or less, is approximately 3.0 kΩ maximum.

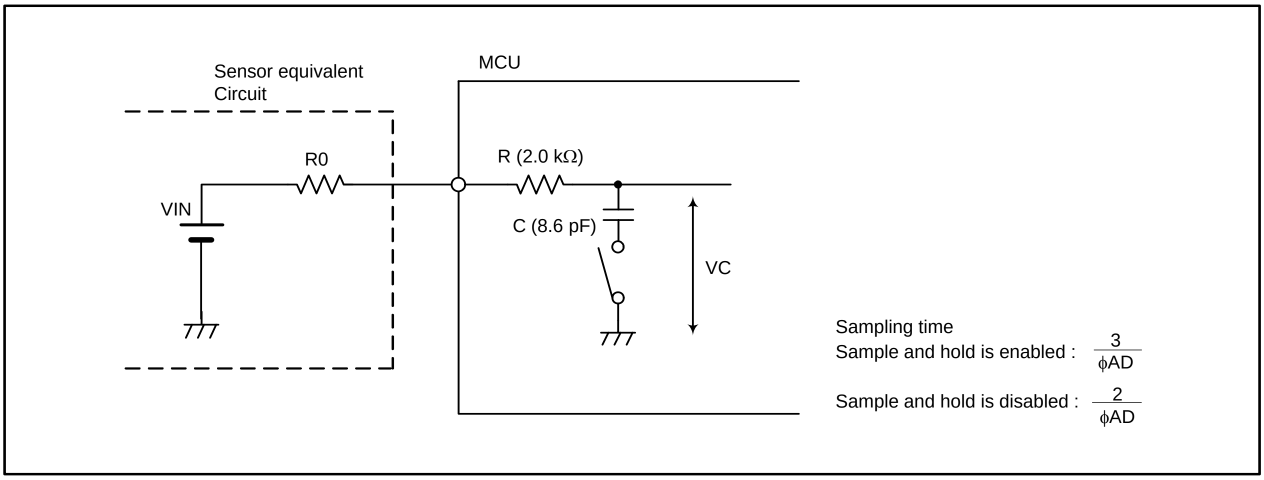

Sensor equivalent
Circuit
MCU
R0
R (2.0 kΩ)
VIN
C (8.6 pF)
VC
Sampling time
Sample and hold is enabled : $\frac{3}{\phi AD}$
Sample and hold is disabled : $\frac{2}{\phi AD}$

Figure 18.9 Analog Input Pin and External Sensor Equivalent Circuit

19. D/A Converter

The D/A converter consists of two independent 8-bit R-2R ladder D/A converter circuits.

Digital code is converted to analog voltage every time a value to be converted is written to the corresponding DAi register (i = 0, 1).

The DAiE bit in the DACON register determines whether the D/A conversion result is output or not. When the DAiE bit is set to 1 (output enabled), pull-up for the corresponding port is disabled.

When the D/A converter is not used, set the DAi register to 00h and the DAiE bit to 0 (output disabled).

Output analog voltage (V) is obtained from the following equation using the value n (n = decimal) set in the DAi register.

$$V = \frac{\text{VREF x n}}{256} \quad (n = 0 \text{ to } 255)$$

VREF: Reference voltage (VREF remains connected even if the VCUT bit in the AD0CON1 register is set to 0)

Table 19.1 lists specifications of the D/A converter. Figure 19.1 shows a block diagram of the D/A converter. Table 19.2 lists pin settings of DA0 and DA1. Figure 19.2 shows registers associated with the D/A converter. Figure 19.3 shows a D/A converter equivalent circuit.

Table 19.1 D/A Converter Specifications

Item	Specification
D/A conversion method	R-2R
Resolution	8 bits
Analog output pin	2 channels

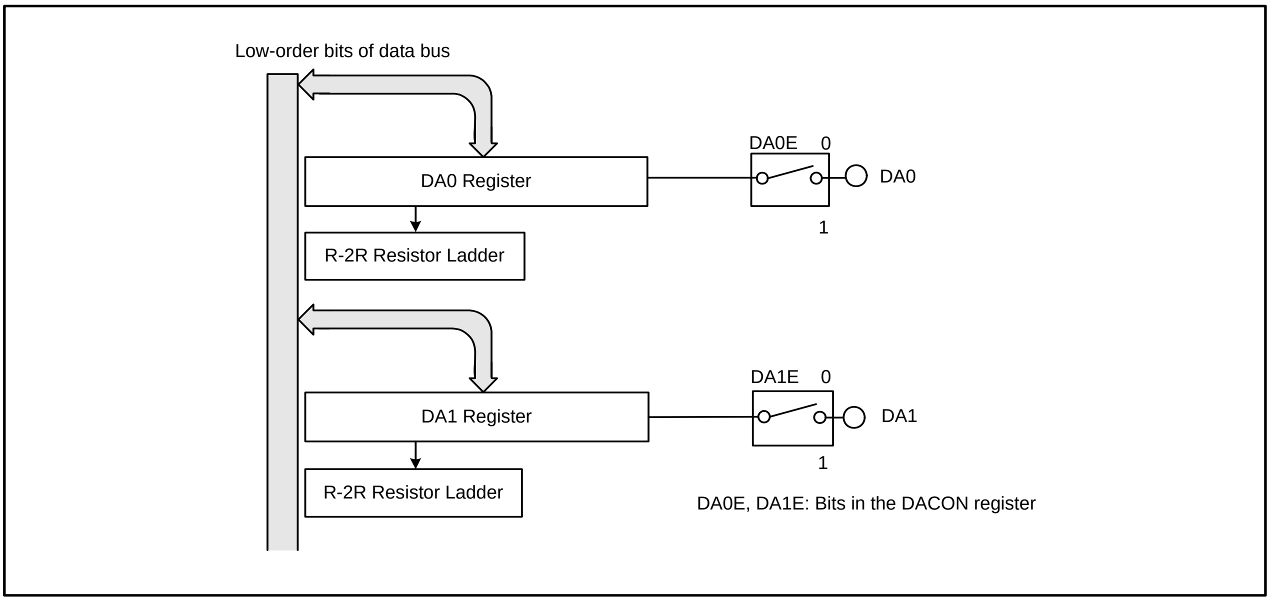


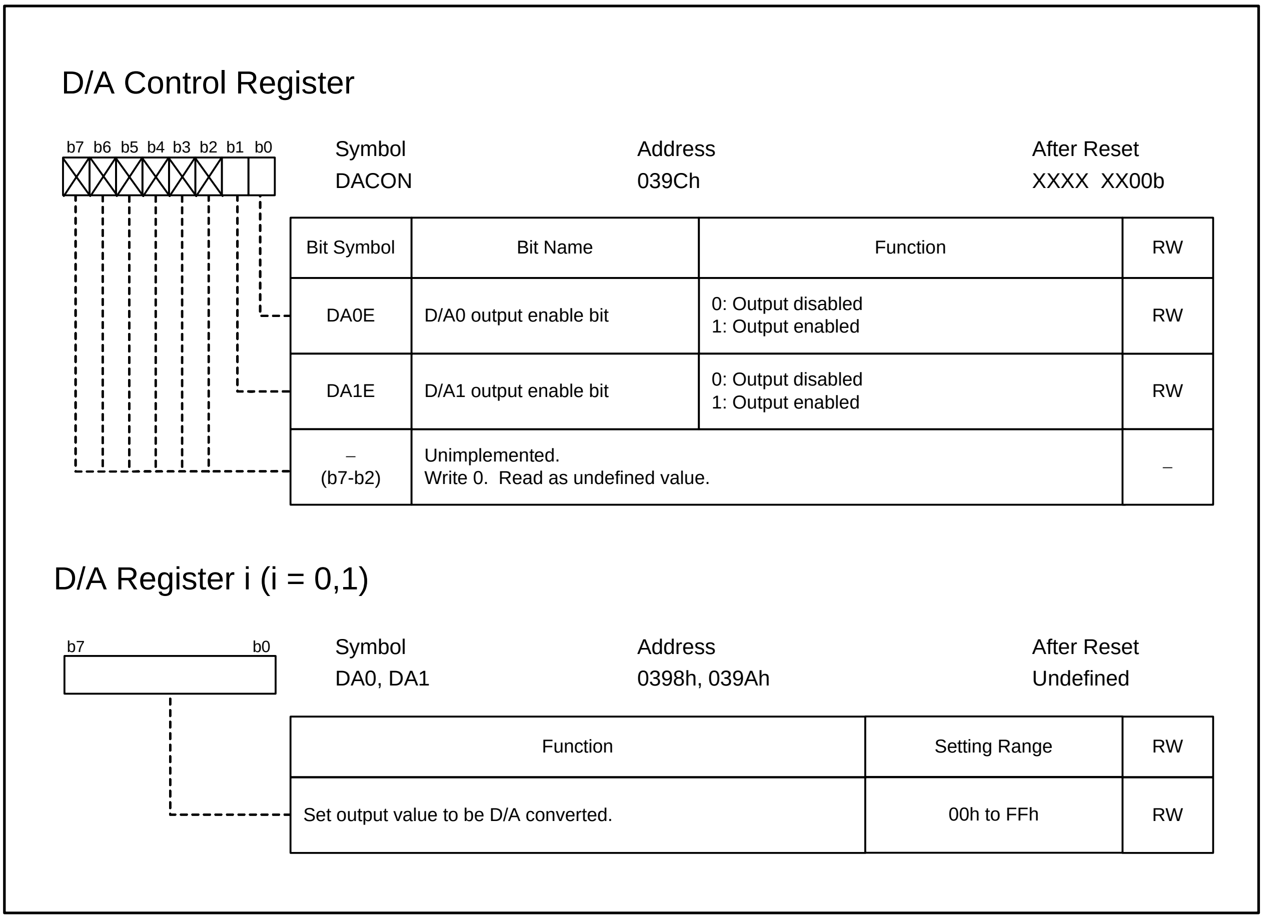
Figure 19.1 D/A Converter Block Diagram

Table 19.2 Pin Settings

Port	Function	Bit Setting		
		PD9 Register[2]	PSL3 Register	PS3 Register[1][2]
P9_3	DA0 output	PD9_3=0	PSL3_3=1	PS3_3=0
P9_4	DA1 output	PD9_4=0	PSL3_4=1	PS3_4=0

NOTES:

1. Set the PS3 register after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.



D/A Control Register

Symbol	Address	After Reset
DACON	039Ch	XXXX XX00b

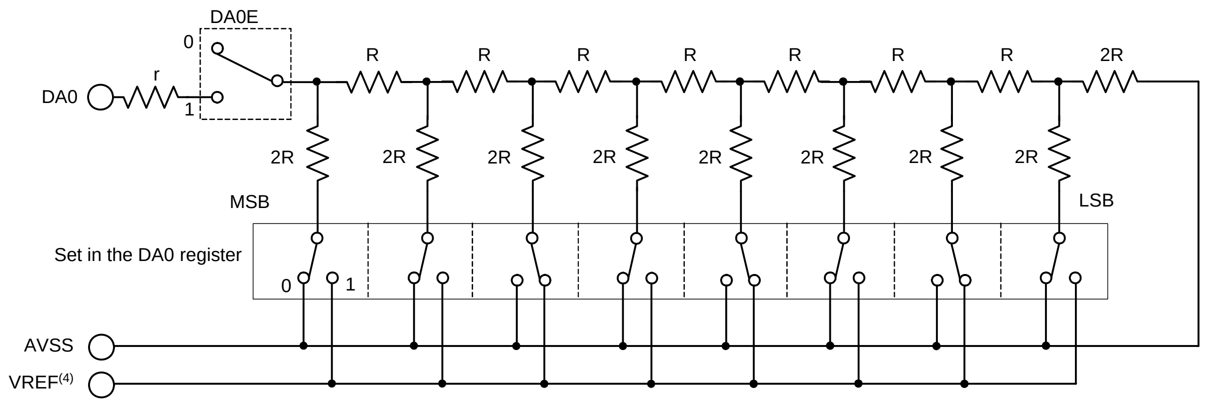
Bit Symbol	Bit Name	Function	RW
DA0E	D/A0 output enable bit	0: Output disabled 1: Output enabled	RW
DA1E	D/A1 output enable bit	0: Output disabled 1: Output enabled	RW
– (b7-b2)	Unimplemented. Write 0. Read as undefined value.		–

D/A Register i (i = 0,1)

Symbol	Address	After Reset
DA0, DA1	0398h, 039Ah	Undefined

Function	Setting Range	RW
Set output value to be D/A converted.	00h to FFh	RW

Figure 19.2 DACON Register, DA0 and DA1 Registers



NOTES:
1. The above applies when the DA0 register is set to 2Ah.
2. D/A1 has the same circuitry as the avove.
3. When the D/A converter is not used, set the DAiE bit (i = 0,1) in the DACON register to 0 (output disabled) and the DAi register to 00h to stop current from flowing into the R-2R resistor to reduce unnecessary power consumption.
4. VREF remains connected even if the VCUT bit in the AD0CON1 register is set to 0 (VREF not connected).

Figure 19.3 D/A Converter Equivalent Circuit

20. CRC Calculation

The CRC (Cyclic Redundancy Check) calculation detects an error in data blocks. A generator polynomial of CRC - CCITT ($X^{16} + X^{12} + X^5 + 1$) generates CRC code.

The CRC code is a 16-bit code generated for a given length of the data block in bytes. The CRC code is stored in the CRCD register every time one-byte data is transferred to the CRCIN register after a default value is written to the CRCD register. CRC code generation for one-byte data is completed in two bus clock cycles.

Figure 20.1 shows a block diagram of the CRC circuit. Figure 20.2 shows CRC-associated registers. Figure 20.3 shows an example of the CRC calculation.

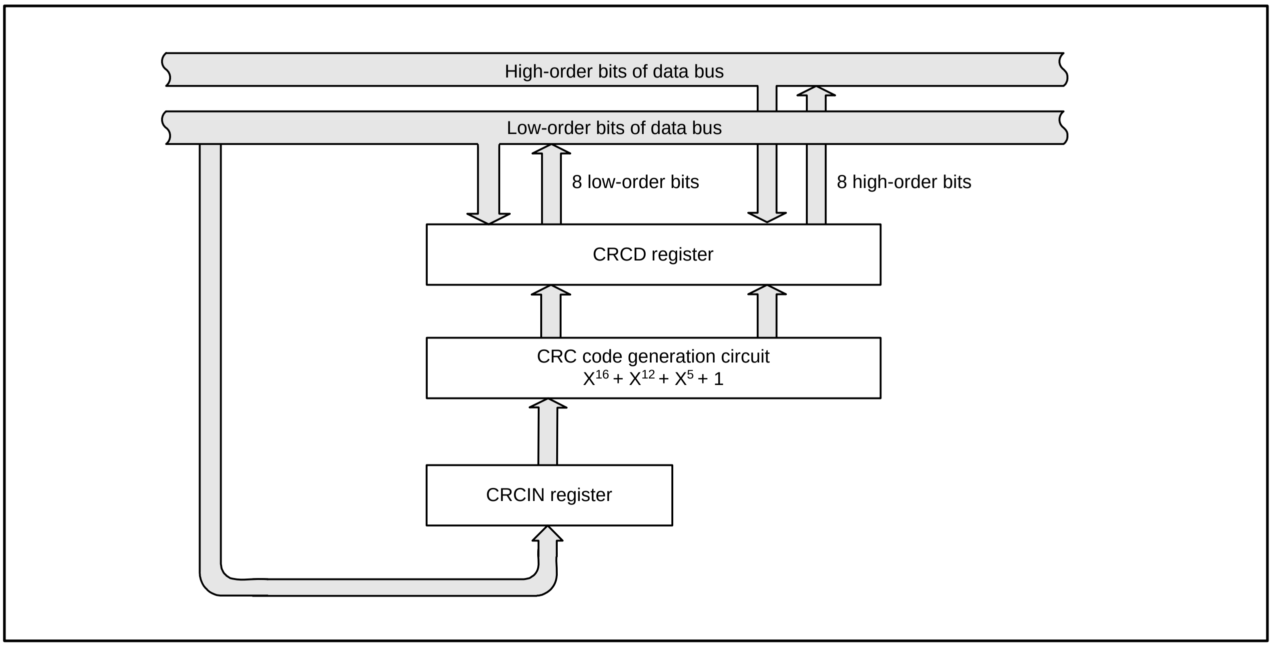

High-order bits of data bus
Low-order bits of data bus
8 low-order bits
8 high-order bits
CRCD register
CRC code generation circuit
$X^{16} + X^{12} + X^5 + 1$
CRCIN register

Figure 20.1 CRC Calculation Block Diagram

CRC Data Register

b15 b8 b7 b0

Symbol	Address	After Reset
CRCD	037Dh - 037Ch	Undefined

Function	Setting Range	RW
After an initial value is written to the CRCD register, the CRC code can be read from the CRCD register by writing data to the CRCIN register. Bit position of the initial value is inverted. The inverted value is read as the CRC code.	0000h to FFFFh	RW

CRC Input Register

b7 b0

Symbol	Adddress	After Reset
CRCIN	037Eh	Undefined

Function	Setting Range	RW
Data input. Inverse bit position of data	00h to FFh	RW

Figure 20.2 CRCD Register, CRCIN Register

CRC Calculation and Setup Procedure to Generate CRC Code for 80C4h

○ CRC Calculation for M32C

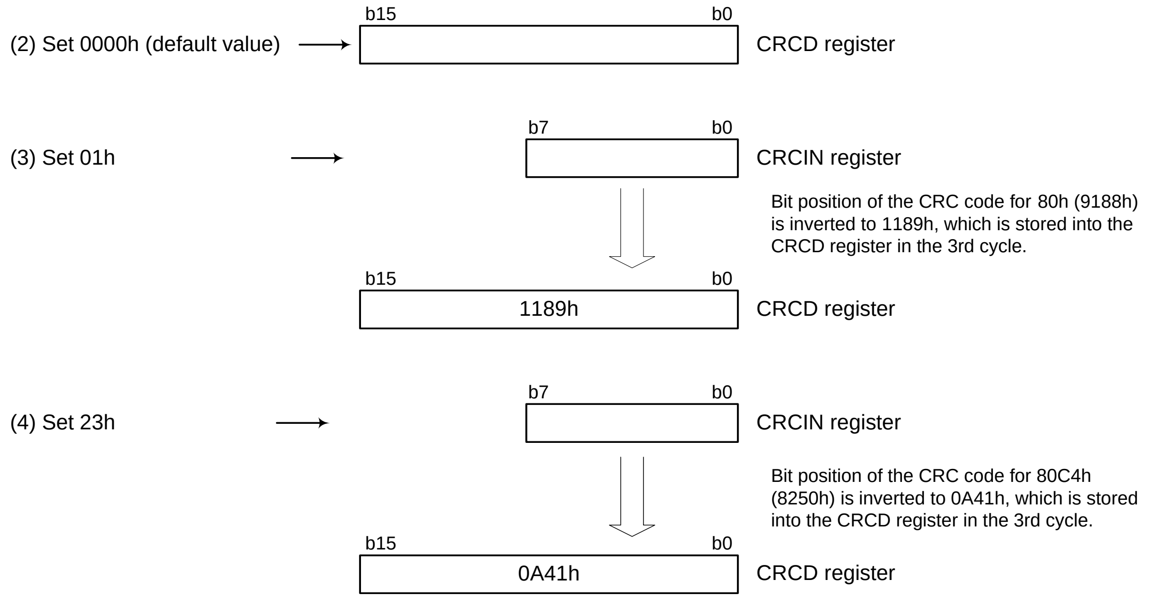
CRC code: a remainder of division, $\frac{\text{value of the CRCIN register with inversed bit position}}{\text{Generator polynomial}}$

Generator polynomial: $X^{16} + X^{12} + X^{5} + 1$ (1 0001 0000 0010 0001b)

○ Setting Steps

(1) Invert a bit position of 80C4h per byte by a program

80h → 01h, C4h → 23h

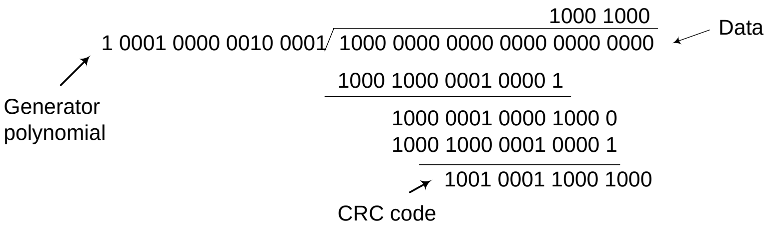


Bit position of the CRC code for 80h (9188h) is inverted to 1189h, which is stored into the CRCD register in the 3rd cycle.

Bit position of the CRC code for 80C4h (8250h) is inverted to 0A41h, which is stored into the CRCD register in the 3rd cycle.

○ Details of CRC Calculation

As shown in (3) above, bit position of 01h (00000001b) written to the CRCIN register is inverted to 10000000b. Add 1000 0000 0000 0000 0000 0000b, as 10000000b plus 16 digits, to 0000h as the initial value of the CRCD register to perform the modulo-2 division.



Modulo-2 Arithmetic is calculated on the law below

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
-1 = 1

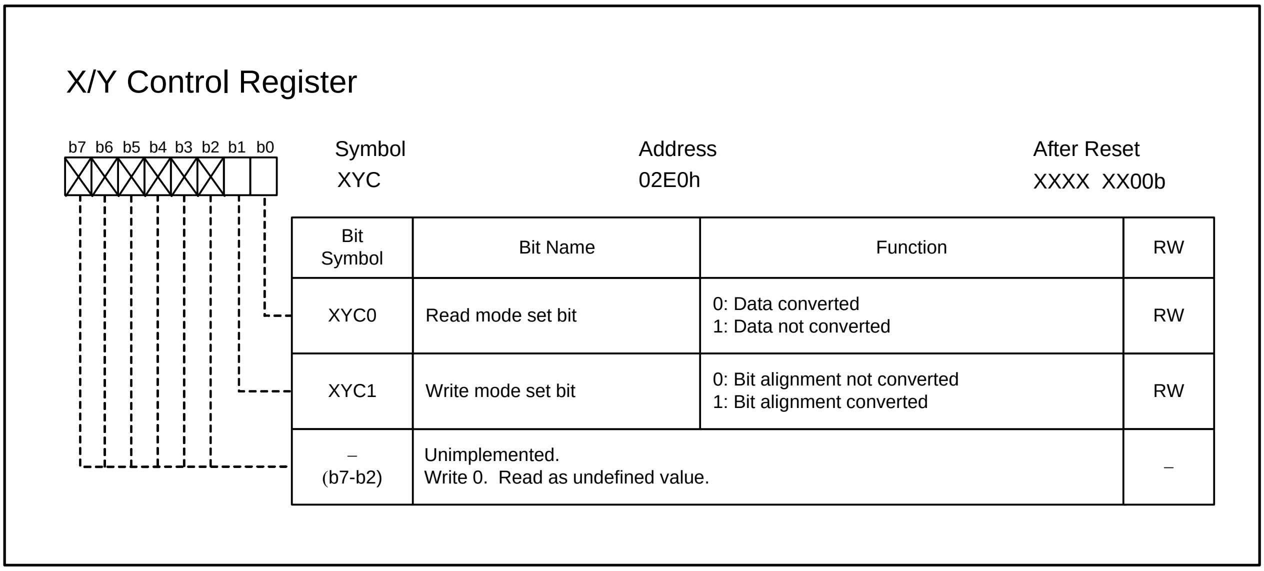
0001 0001 1000 1001b (1189h), the remainder 1001 0001 1000 1000b (9188h) with inversed bit position, can be read from the CRCD register.
When going on to (4) above, 23h (00100011b) written in the CRCIN register is inverted to 11000100b.
Add 1100 0100 0000 0000 0000 0000b plus 16 digits, to 1001 0001 1000 1000b as a remainder of (3) left in the CRCD register to perform the modulo-2 division.
0000 1010 0100 0001b (0A41h), the remainder with inverted bit position, can be read from CRCD register.

Figure 20.3 CRC Calculation

21. X/Y Conversion

The X/Y conversion rotates a 16 x 16 matrix data by 90 degrees and also inverts high-order bits and low-order bits of a 16-bit data. Figure 21.1 shows the XYC register.

The 16-bit XiR register (i = 0 to 15) and 16-bit YjR register (j = 0 to 15) are allocated to the same address. The XiR register is a write-only register, while the YjR register is a read-only register. Access registers XiR and YjR from an even address in 16-bit units. Performance cannot be guaranteed if registers XiR and YjR are accessed in 8-bit units.



X/Y Control Register

Symbol	Address	After Reset
XYC	02E0h	XXXX XX00b

Bit Symbol	Bit Name	Function	RW
XYC0	Read mode set bit	0: Data converted 1: Data not converted	RW
XYC1	Write mode set bit	0: Bit alignment not converted 1: Bit alignment converted	RW
– (b7-b2)	Unimplemented. Write 0. Read as undefined value.		–

Figure 21.1 XYC Register

The XYC0 bit in the XYC register determines how to read the YjR register.

When setting the XYC0 bit to 0 (data converted) and reading the YjR register, all the bits j in registers X0R to X15R can be read.

For example, bit 0 in the X0R register can be read when reading bit 0 in the Y0R register, bit 0 in the X1R register when reading bit 1 in the Y0R register..., bit 0 in the X14R register when reading bit 14 in the Y0R register, and bit 0 in the X15R register when reading bit 15 in the Y0R register.

Figure 21.2 shows a conversion table when the XYC0 bit is set to 0. Figure 21.3 shows an example of the X/Y conversion.

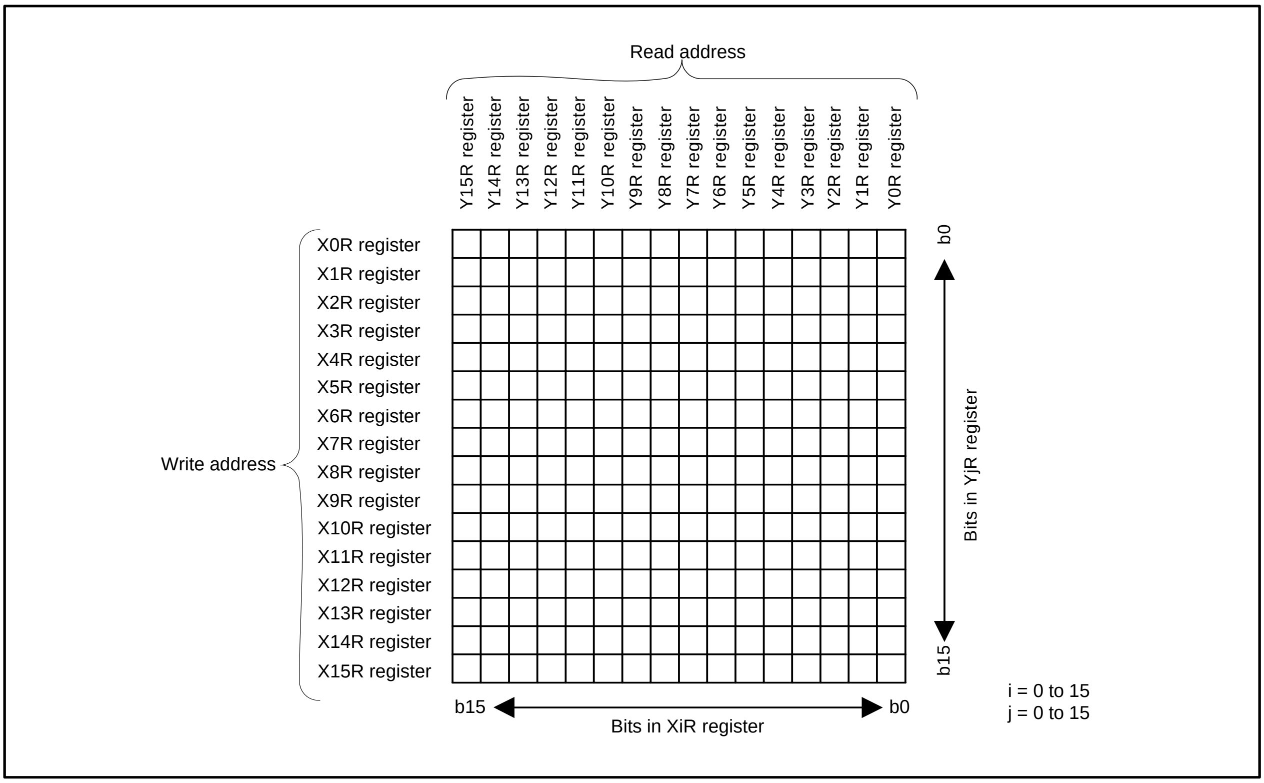


Figure 21.2 Conversion Table when the XYC0 Bit is Set to 0

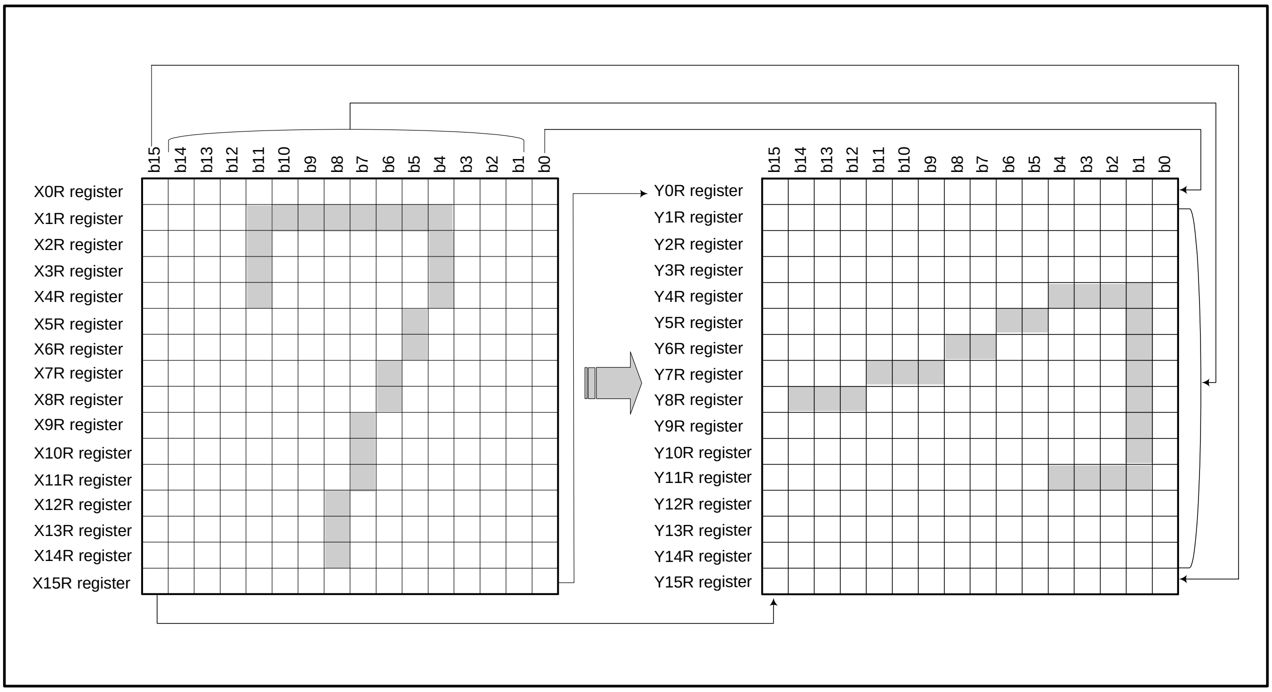


Figure 21.3 X/Y Conversion

When setting the XYC0 bit in the XYC register to 1 (data not converted) and reading the YjR register, the value written to the XiR register can be read. Figure 21.4 shows a conversion table when the XYC0 bit is set to 1.

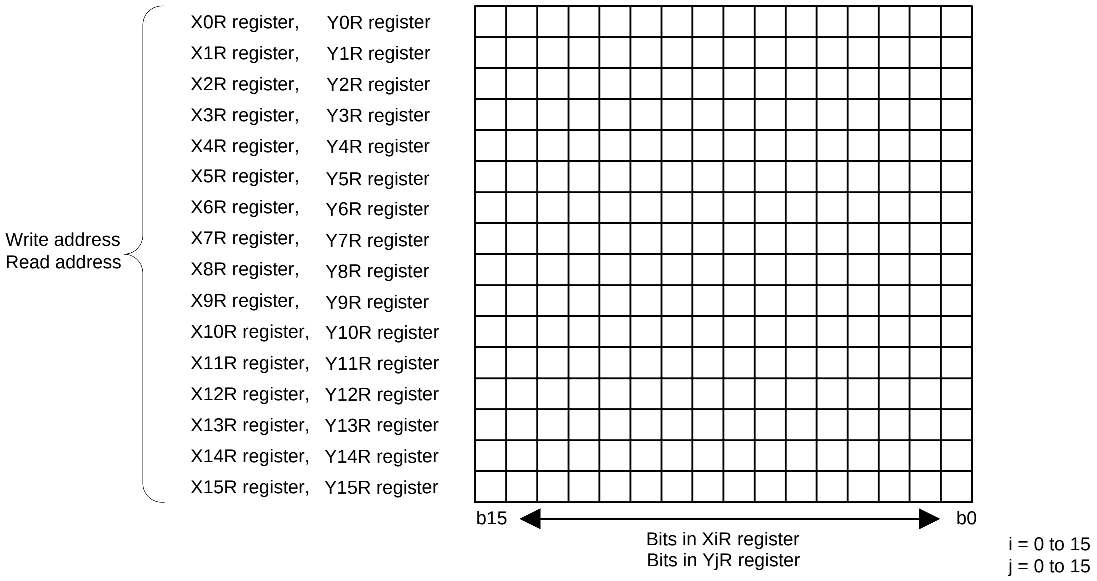

Write address
Read address
X0R register, Y0R register
X1R register, Y1R register
X2R register, Y2R register
X3R register, Y3R register
X4R register, Y4R register
X5R register, Y5R register
X6R register, Y6R register
X7R register, Y7R register
X8R register, Y8R register
X9R register, Y9R register
X10R register, Y10R register
X11R register, Y11R register
X12R register, Y12R register
X13R register, Y13R register
X14R register, Y14R register
X15R register, Y15R register
b15
b0
Bits in XiR register
Bits in YjR register
i = 0 to 15
j = 0 to 15

Figure 21.4 Conversion Table when the XYC0 Bit is Set to 1

The XYC1 bit in the XYC register selects bit alignment written to the XiR register.
When the XYC1 bit is set to 0 (bit alignment not converted) and writing to the XiR register, bit alignment is written as is. When the XYC1 bit is set to 1 (bit alignment converted) and writing to the XiR register, inverted bit alignment is written.
Figure 21.5 shows a conversion when the XYC1 bit is set to 1.

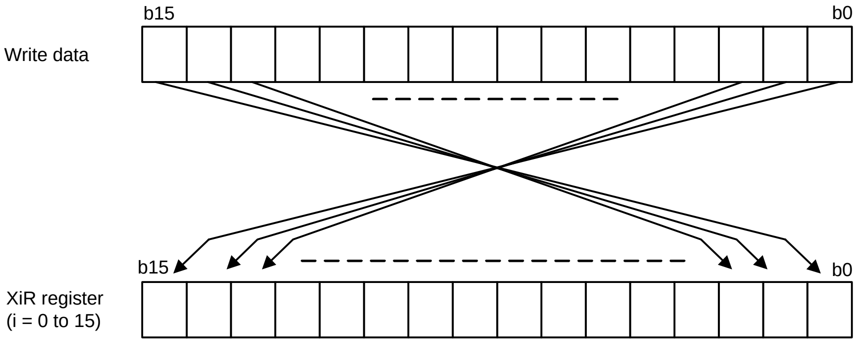


Figure 21.5 Conversion when the XYC1 Bit is Set to 1

22. Programmable I/O Ports

123 programmable I/O ports, P0 to P15 (excluding P8_5), are available in the 144-pin package. 87 programmable I/O ports, P0 to P10 (excluding P8_5), are available in the 100-pin package. The Port Pi Direction Registers determine individual port status, input or output. The pull-up control registers determine whether the ports, divided into groups of four, are pulled up or not. P8_5 is an input-only port and cannot be pulled up internally. The P8_5 bit in the P8 register indicates an $\overline{\text{NMI}}$ input level since P8_5 shares its pin with $\overline{\text{NMI}}$.
Figures 22.1 to 22.4 show programmable I/O port configurations.
Each pin functions as a programmable I/O port, I/O pin for internal peripheral function, or bus control pin.
To use as an I/O pin for peripheral function, refer to the description for individual peripheral functions. Refer to **8. Bus** when used as a bus control pin.
Registers associated with the programmable I/O ports are as follows.

22.1 Port Pi Direction Register (PDi Register, i = 0 to 15)

Figure 22.5 shows the PDi register.
The PDi register configures a programmable I/O port as either input or output. Each bit in the PDi register corresponds to one port.
In memory expansion mode and microprocessor mode, the PDi register corresponding to the following bus control pins cannot be written: A0 to A22, $\overline{\text{A23}}$, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WRL}}$ / $\overline{\text{WR}}$, $\overline{\text{WRH}}$ / $\overline{\text{BHE}}$, $\overline{\text{RD}}$, BCLK / ALE / CLKOUT, $\overline{\text{HLDA}}$ / ALE, $\overline{\text{HOLD}}$, ALE, and $\overline{\text{RDY}}$. No bit controlling P8_5 is provided in the PDi register.

22.2 Port Pi Register (Pi Register, i = 0 to 15)

Figure 22.6 shows the Pi register.
The MCU inputs/outputs data from/to external devices by reading and writing to the Pi register. The Pi register consists of a port latch to hold output data and a circuit to read the pin level. Each bit in the Pi register corresponds to one port.
In memory expansion mode and microprocessor mode, the Pi register corresponding to the following bus control pins cannot be written and the port level cannot be read from the Pi register: A0 to A22, $\overline{\text{A23}}$, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WRL}}$/ $\overline{\text{WR}}$, $\overline{\text{WRH}}$ / $\overline{\text{BHE}}$, $\overline{\text{RD}}$, BCLK / ALE / CLKOUT, $\overline{\text{HLDA}}$ / ALE, $\overline{\text{HOLD}}$, ALE, and $\overline{\text{RDY}}$.

22.3 Function Select Register A (PSj Register, j = 0 to 3)

Figures 22.7 to 22.8 show the PSj registers.
The PSj register selects either I/O port or peripheral function output if these functions share a single pin (excluding DA0 and DA1).
When multiple peripheral function outputs are assigned to a single pin, set registers PSL0 to PSL3, and PSC to select which function to use.
Tables 22.3 to 22.7 list peripheral function output control settings for each pin.

22.4 Function Select Register B (PSLk Register, k = 0 to 3)

Figures 22.9 to 22.10 show the PSLk register.
When multiple peripheral function outputs are assigned to a single pin, the PSLk register selects which peripheral function output to use.
Refer to **22.8 Analog Input and Other Peripheral Function Input** for information on bits PSL3_3 to PSL3_6 in the PSL3 register.

22.5 Function Select Register C (PSC Register)

Figure 22.11 shows the PSC register.
When multiple peripheral function outputs are assigned to a single pin, the PSC register selects which peripheral function output to use.
Refer to **22.8 Analog Input and Other Peripheral Function Input** for information on the PSC_7 bit in the PSC register.

22.6 Pull-up Control Register 0 to 4 (PUR0 to PUR4 Registers)

Figures 22.12 to 22.15 show registers PUR0 to PUR4.

Registers PUR0 to PUR4 select whether the ports, divided into groups of four, are pulled up or not. Set the bit in registers PUR0 to PUR4 to 1 (pull-up) and the bit in the PDi register to 0 (input mode) to pull-up the corresponding port.

In memory expansion mode and microprocessor mode, set bits, corresponding to the bus control pins (P0 to P5), in registers PUR0 and PUR1 to 0 (no pull-up). P0, P1, and P4_0 to P4_3 can be pulled up when they are used as input ports in memory expansion mode and microprocessor mode.

22.7 Port Control Register (PCR Register)

Figure 22.16 shows the PCR register.

The PCR register selects either CMOS output or N-channel open drain output as port P1 output format. When the PCR0 bit is set to 1, P channel in the CMOS port is turned off at all times and in result port P1 becomes N-channel open drain output. This is, however, pseudo open drain. Therefore, the absolute maximum rating of the input voltage is from -0.3 V to VCC2 + 0.3 V.

To use port P1 as data bus in memory expansion mode and microprocessor mode, set the PCR0 bit to 0 (CMOS output). When port P1 is used as a port in memory expansion mode and microprocessor mode, set the output format using the PCR0 bit.

22.8 Analog Input and Other Peripheral Function Input

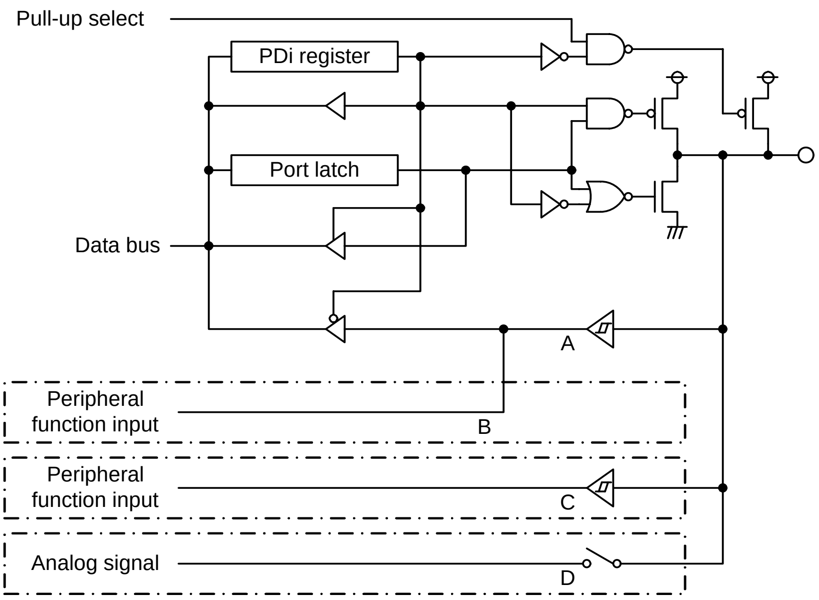
Bits PSL3_3 to PSL3_6 in the PSL3 register, and the PSC_7 bit in the PSC register are used to separate peripheral function inputs from analog input/output. If the analog I/O shares the pin with other peripheral function inputs, a through current may flow to the peripheral function inputs when an intermediate voltage is applied to the pin.

To use the analog I/O (DA0, DA1, ANEX0, ANEX1, or AN_4 to AN_7), set the corresponding bit to 1 (analog I/O), and disconnect the peripheral function inputs to prevent an intermediate voltage from being applied to the peripheral function inputs.

For P10_4 to P10_7 (AN_4 to AN_7/$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$), when the PSC_7 bit is set to 1, the input buffer for the peripheral functions including the port function is disconnected and ports P10_4 to P10_7 are read as undefined. Also, the IR bit in the KUPIC register remains unchanged as 0 (interrupt not requested) even if $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ pin input levels are changed.

Set the corresponding bit to 0 (except analog I/O) when analog I/O is not used.

Programmable I/O ports

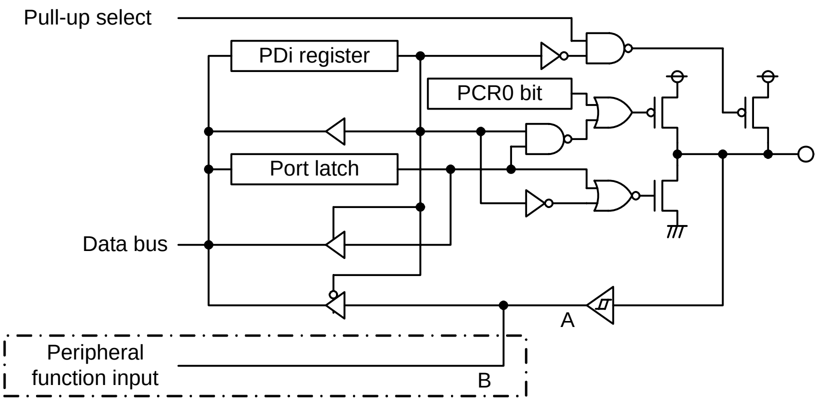


Option / Port	(A) Hysteresis	(B) Peripheral function input	(C) Peripheral function input	(D) Analog I/F
P0_0 to P0_7 P2_0 to P2_7	–	–	–	○
P3_0 to P3_7 P4_0 to P4_7 P5_0 to P5_2	–	–	–	–
P5_5, P5_7	–	–	○	–
P8_3, P8_4	○	○	–	–
P8_6, P8_7	–	–	–	–

○ : Available –: Not available

Figure 22.1 Programmable I/O Ports (1/4)

Programmable I/O ports with the port control register

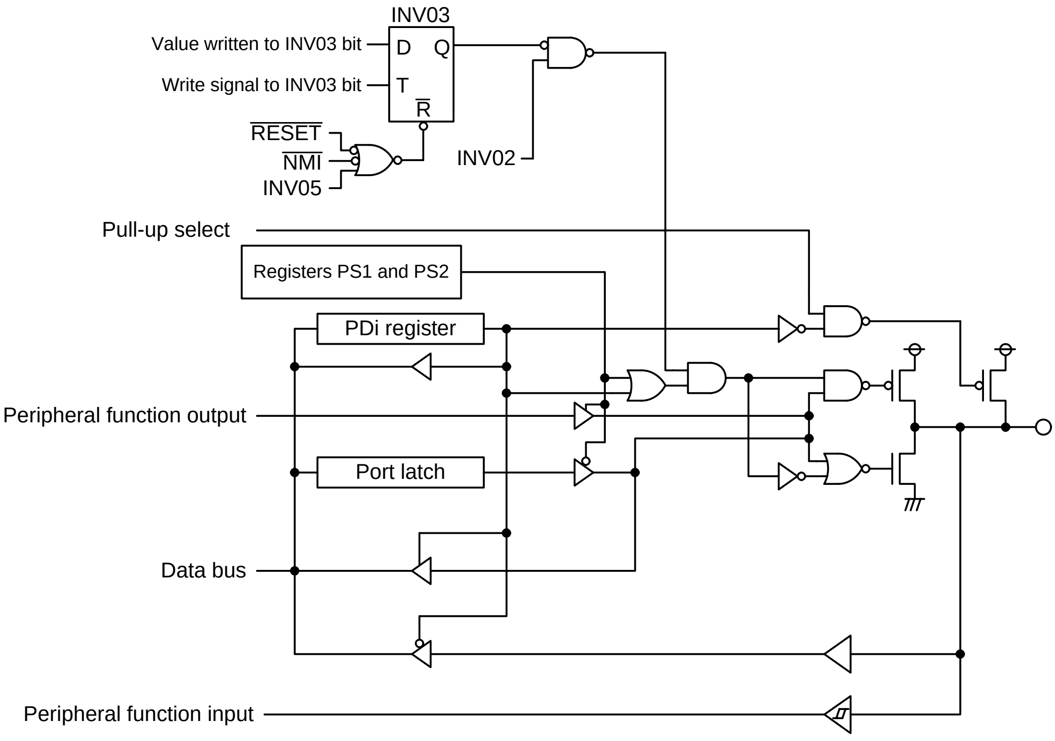


PCR0 bit: bit in the PCR register

Option Port	(A) Hysteresis	(B) Peripheral function input
P1_0 to P1_4	–	–
P1_5 to P1_7	○	○

○ : Available –: Not available

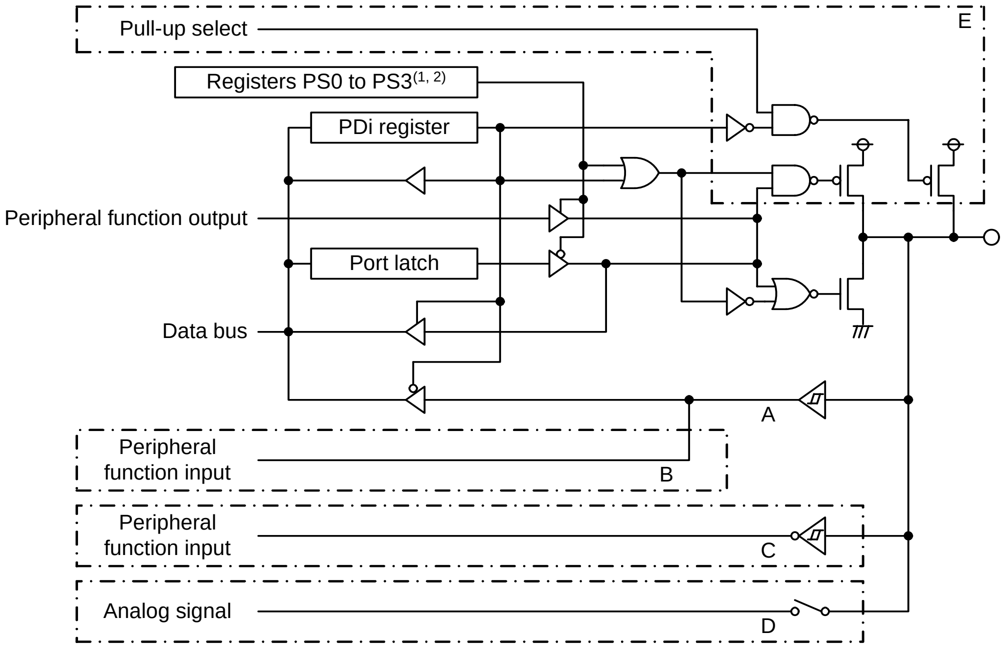
Programmable I/O ports with the function select register


INV03
Value written to INV03 bit
Write signal to INV03 bit
D
Q
T
$\overline{R}$
$\overline{\text{RESET}}$
$\overline{\text{NMI}}$
INV05
INV02
Pull-up select
Registers PS1 and PS2
PDi register
Peripheral function output
Port latch
Data bus
Peripheral function input

Port P7_2 to P7_5, P8_0, P8_1

Figure 22.2 Programmable I/O Ports (2/4)

Programmable I/O ports with the function select register



	Option Port	(A) Hysteresis	(B) Peripheral fucntion input	(C) Peripheral fucntion input	(D) Analog I/F	(E) Circuit
	P5_3(1)	–	–	–	–	○
	P5_4, P5_6(2)	–	–	–	–	○
	P6_0 to P6_7	–	–	○	–	○
	P7_0, P7_1(3)	–	–	○	–	–
	P7_6, P7_7	–	–	○	–	○
	P8_2	○	○	–	–	○
	P9_0 to P9_2	–	–	○	–	○
	P9_3 to P9_6	–	–	○	○	○
	P9_7	–	–	○	–	○
	P10_0 to P10_3	–	–	–	○	○
	P10_4 to P10_7	○	○	–	○	○
(note 4)	P11_0 to P11_3	–	–	–	–	○
	P11_4, P12_0	–	–	–	–	○
	P12_1 to P12_3	–	–	–	–	○
	P12_4 to P12_7 P13_0 to P13_4	–	–	–	–	○
	P13_5, P13_6	–	–	–	–	○
	P13_7	–	–	–	–	○
	P14_0 to P14_3	–	–	–	–	○
	P14_4 to P14_6	–	–	–	–	○
	P15_0	–	–	–	○	○
	P15_1 to P15_3	–	–	–	○	○
	P15_4	–	–	–	○	○
	P15_5 to P15_7	–	–	–	○	○

○ : Available –: Not available

NOTES:

1. For P5_3, use the PM07 bit in the PM0 register, bits PM15 and PM14 in the PM1 register, and bits CM01 and CM00 in the CM0 register to select CLKOUT or ALE output.
2. For P5_4 and P5_6, use bits PM15 and PM14 to select ALE output.
3. P7_0 and P7_1 are N-channel open drain output ports.
4. These ports are provided in the 144-pin package only.

Figure 22.3 Programmable I/O Ports (3/4)

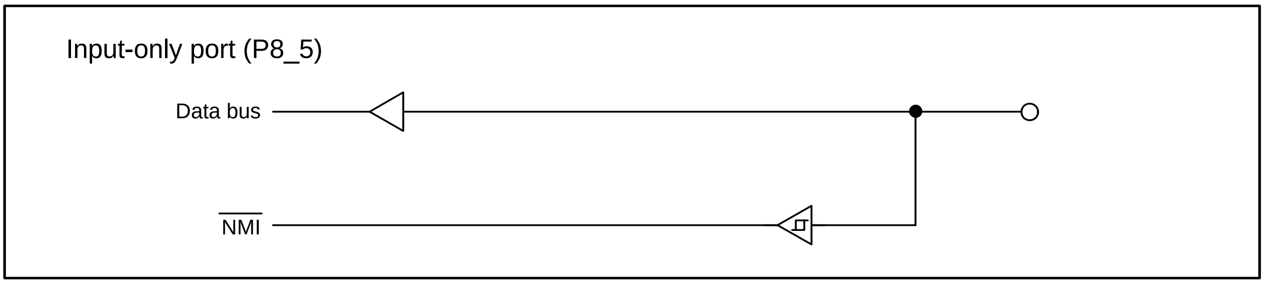
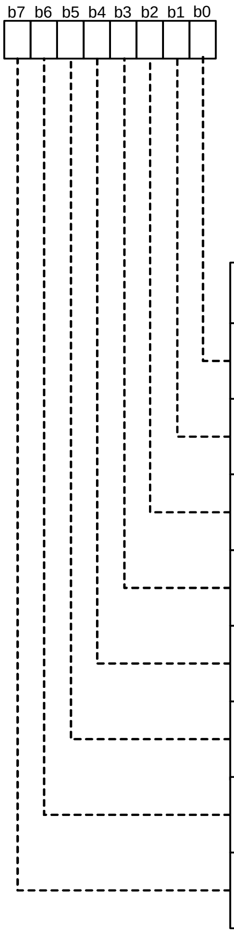


Figure 22.4 Programmable I/O Ports (4/4)

Port Pi Direction Register (i = 0 to 15)[2]



Symbol	Address	After Reset
PD0 to PD3	03E2h, 03E3h, 03E6h, 03E7h	00h
PD4 to PD7	03EAh, 03EBh, 03C2h, 03C3h	00h
PD8	03C6h[4]	00X0 0000b
PD9, PD10	03C7h[1], 03CAh	00h
PD11	03CBh[3, 4]	XXX0 0000b
PD12, PD13	03CEh, 03CFh[3]	00h
PD14	03D2h[3, 4]	X000 0000b
PD15	03D3h[3]	00h

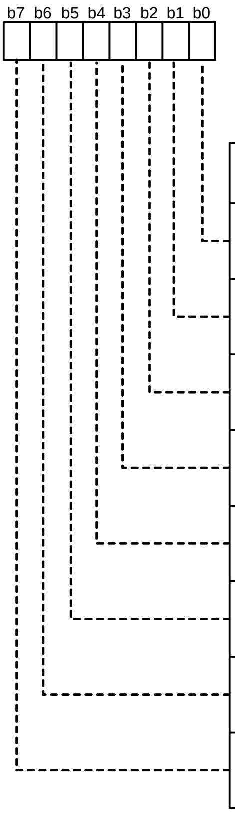
Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi_0 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW
PDi_1	Port Pi_1 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW
PDi_2	Port Pi_2 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW
PDi_3	Port Pi_3 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW
PDi_4	Port Pi_4 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW
PDi_5	Port Pi_5 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW
PDi_6	Port Pi_6 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW
PDi_7	Port Pi_7 direction bit	0: Input mode (functions as input port) 1: Output mode (functions as output port)	RW

NOTES:

1. Set the PD9 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
2. In memory expansion mode or microprocessor mode, the PDi register corresponding to the following bus control pins cannot be written: A0 to A22, $\overline{\text{A23}}$, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WRL}}$/ $\overline{\text{WR}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$, $\overline{\text{RD}}$, BCLK/ALE/CLKOUT, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE, $\overline{\text{RDY}}$.
3. Set registers PD11 to PD15 to FFh in the 100-pin package.
4. Nothing is implemented to the PD8_5 bit in the PD8 register, bits PD11_7 to PD11_5 in the PD11 register, and the P14_7 bit in the PD14 register. Write a 0. A read from these bits returns undefined value.

Figure 22.5 PD0 to PD15 Registers

Port Pi Register [1, 2] (i = 0 to 15)



Symbol	Address	After Reset
P0 to P5	03E0h, 03E1h, 03E4h, 03E5h, 03E8h, 03E9h	Undefined
P6 to P10	03C0h, 03C1h[3], 03C4h[4], 03C5h, 03C8h	Undefined
P11 to P15	03C9h[5], 03CCh, 03CDh, 03D0h[5], 03D1h	Undefined

Bit Symbol	Bit Name	Function	RW
Pi_0	Port Pi_0 bit	Input mode (The PDi_j bit (j = 0 to 7) in the PDi register = 0) Read: Return the pin level. Write: Write to the port latch. Output mode (The PDi_j bit in the PDi register = 1) Read: Return the port latch value. Write: Write to the port latch and the port latch value is output from the pin. 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. In memory expansion mode and microprocessor mode, the Pi register corresponding to the following bus control pins cannot be written: A0 to A22, $\overline{\text{A23}}$, D0 to D15, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WRL}}$/ $\overline{\text{WR}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$, $\overline{\text{RD}}$, BCLK/ALE/CLKOUT, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE, $\overline{\text{RDY}}$.
2. Ports P11 to P15 are provided in the 144-pin package only.
3. P7_0 and P7_1 are N-channel open drain output ports. The pins are placed into high-impedance states when the corresponding bits to P7_0 and P7_1 are set to 1.
4. The P8_5 bit is a read-only bit.
5. Nothing is implemented to bits P11_5 to P11_7 in the P11 register and the P14_7 bit in the P14 register. Write a 0. A read from these bits returns undefined value.

Figure 22.6 P0 to P15 Registers

Function Select Register A0

b7	b6	b5	b4	b3	b2	b1	b0

Symbol
PS0

Address
03B0h

After Reset
00h

Bit Symbol	Bit Name	Function	RW
PS0_0	Port P6_0 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_0 bit	RW
PS0_1	Port P6_1 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_1 bit	RW
PS0_2	Port P6_2 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_2 bit	RW
PS0_3	Port P6_3 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_3 bit	RW
PS0_4	Port P6_4 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_4 bit	RW
PS0_5	Port P6_5 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_5 bit	RW
PS0_6	Port P6_6 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_6 bit	RW
PS0_7	Port P6_7 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL0_7 bit	RW

Function Select Register A1

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol
PS1

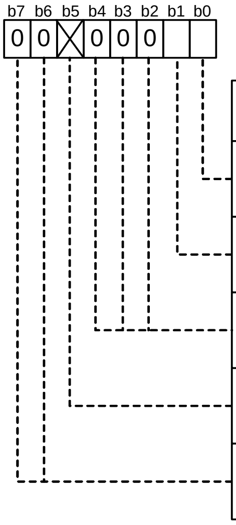
Address
03B1h

After Reset
00h

Bit Symbol	Bit Name	Function	RW
PS1_0	Port P7_0 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL1_0 bit	RW
PS1_1	Port P7_1 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL1_1 bit	RW
PS1_2	Port P7_2 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL1_2 bit	RW
PS1_3	Port P7_3 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL1_3 bit	RW
PS1_4	Port P7_4 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL1_4 bit	RW
PS1_5	Port P7_5 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL1_5 bit	RW
PS1_6	Port P7_6 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL1_6 bit	RW
PS1_7	Port P7_7 output function select bit	0: I/O port/peripheral function input 1: Do not set to this value	RW

Figure 22.7 PS0 Register, PS1 Register

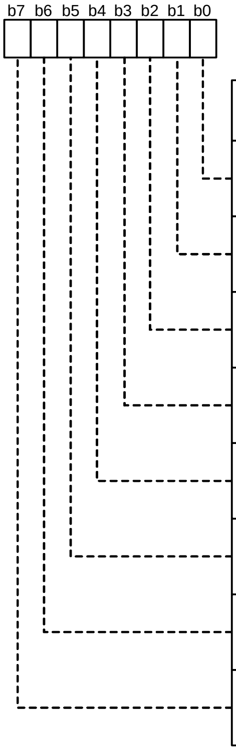
Function Select Register A2



Symbol	Address	After Reset
PS2	03B4h	00X0 0000b

Bit Symbol	Bit Name	Function	RW
PS2_0	Port P8_0 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL2_0 bit	RW
PS2_1	Port P8_1 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL2_1 bit	RW
– (b4-b2)	Reserved bits	Set to 0	RW
– (b5)	Unimplemented. Write 0. Read as undefined value.		–
– (b7-b6)	Reserved bits	Set to 0	RW

Function Select Register A3[1]



Symbol	Address	After Reset
PS3	03B5h	00h

Bit Symbol	Bit Name	Function	RW
PS3_0	Port P9_0 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL3_0 bit	RW
PS3_1	Port P9_1 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL3_1 bit	RW
PS3_2	Port P9_2 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL3_2 bit	RW
PS3_3	Port P9_3 output function select bit	0: I/O port/peripheral function input 1: $\overline{\text{RTS3}}$	RW
PS3_4	Port P9_4 output function select bit	0: I/O port/peripheral function input 1: $\overline{\text{RTS4}}$	RW
PS3_5	Port P9_5 output function select bit	0: I/O port/peripheral function input 1: CLK4 output	RW
PS3_6	Port P9_6 output function select bit	0: I/O port/peripheral function input 1: TXD4/SDA4 output	RW
PS3_7	Port P9_7 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL3_7 bit	RW

NOTE:

1. Set the PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.

Figure 22.8 PS2 Register, PS3 Register

Function Select Register B0

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

Symbol	Address	After Reset
PSL0	03B2h	00h

Bit Symbol	Bit Name	Function	RW
PSL0_0	Port P6_0 peripheral function output select bit	0: $\overline{RTS0}$ 1: Do not set to this value	RW
PSL0_1	Port P6_1 peripheral function output select bit	0: CLK0 output 1: Do not set to this value	RW
PSL0_2	Port P6_2 peripheral function output select bit	0: SCL0 output 1: STXD0	RW
PSL0_3	Port P6_3 peripheral function output select bit	0: TXD0/SDA0 output 1: Do not set to this value	RW
PSL0_4	Port P6_4 peripheral function output select bit	0: $\overline{RTS1}$ 1: Do not set to this value	RW
PSL0_5	Port P6_5 peripheral function output select bit	0: CLK1 output 1: Do not set to this value	RW
PSL0_6	Port P6_6 peripheral function output select bit	0: SCL1 output 1: STXD1	RW
PSL0_7	Port P6_7 peripheral function output select bit	0: TXD1/SDA1 output 1: Do not set to this value	RW

Function Select Register B1

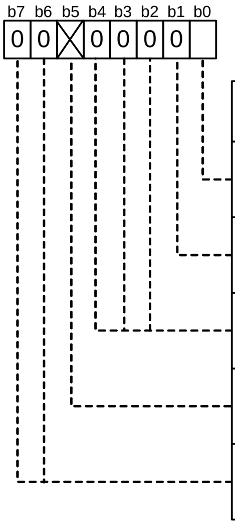
b7	b6	b5	b4	b3	b2	b1	b0
0	1	0					

Symbol	Address	After Reset
PSL1	03B3h	00h

Bit Symbol	Bit Name	Function	RW
PSL1_0	Port P7_0 peripheral function output select bit	0: Select by the PSC_0 bit 1: TA0OUT output	RW
PSL1_1	Port P7_1 peripheral function output select bit	0: Select by the PSC_1 bit 1: STXD2	RW
PSL1_2	Port P7_2 peripheral function output select bit	0: Select by the PSC_2 bit 1: TA1OUT output	RW
PSL1_3	Port P7_3 peripheral function output select bit	0: Select by the PSC_3 bit 1: $\overline{V}$	RW
PSL1_4	Port P7_4 peripheral function output select bit	0: Select by the PSC_4 bit 1: W	RW
PSL1_5	Port P7_5 peripheral function output select bit	0: $\overline{W}$ 1: Do not set to this value	RW
PSL1_6	Port P7_6 peripheral function output select bit	0: Do not set to this value 1: TA3OUT output	RW
– (b7)	Reserved bit	Set to 0	RW

Figure 22.9 PSL0 Register, PSL1 Register

Function Select Register B2



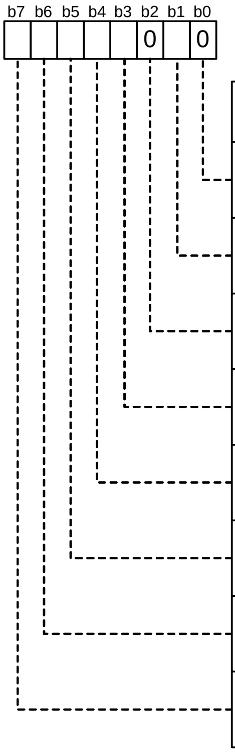
Symbol
PSL2

Address
03B6h

After Reset
00X0 0000b

Bit Symbol	Bit Name	Function	RW
PSL2_0	Port P8_0 peripheral function output select bit	0: TA4OUT output 1: U	RW
PSL2_1	Port P8_1 peripheral function output select bit	0: $\overline{U}$ 1: Do not set to this value	RW
– (b4-b2)	Reserved bits	Set to 0	RW
– (b5)	Unimplemented. Write 0. Read as undefined value.		–
– (b7-b6)	Reserved bits	Set to 0	RW

Function Select Register B3



Symbol
PSL3

Address
03B7h

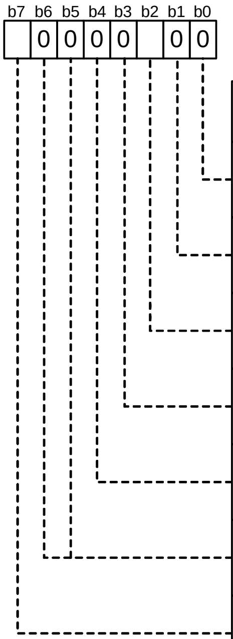
After Reset
00h

Bit Symbol	Bit Name	Function	RW
PSL3_0	Port P9_0 peripheral function output select bit	0: CLK3 output 1: Do not set to this value	RW
PSL3_1	Port P9_1 peripheral function output select bit	0: SCL3 output 1: STXD3	RW
PSL3_2	Port P9_2 peripheral function output select bit	0: TXD3/SDA3 output 1: Do not set to this value	RW
PSL3_3	Port P9_3 peripheral function output select bit(1)	0: Peripheral function input 1: DA0	RW
PSL3_4	Port P9_4 peripheral function output select bit(1)	0: Peripheral function input 1: DA1	RW
PSL3_5	Port P9_5 peripheral function output select bit(1)	0: Peripheral function input except ANEX0 1: ANEX0	RW
PSL3_6	Port P9_6 peripheral function output select bit(1)	0: Peripheral function input except ANEX1 1: ANEX1	RW
PSL3_7	Port P9_7 peripheral function output select bit	0: SCL4 output 1: STXD4	RW

NOTE:
1. If DA0, DA1, ANEX0, and ANEX1 are used with the PSL3_i bit (i = 3 to 6) setting to 0, current consumption may increase.

Figure 22.10 PSL2 Register, PSL3 Register

Function Select Register C



Symbol PSC

Address 03AFh

After Reset 00X0 0000b

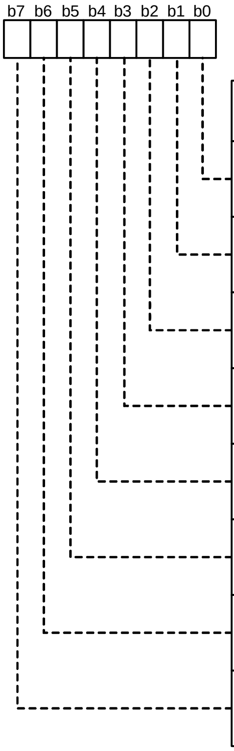
Bit Symbol	Bit Name	Function	RW
PSC_0	Port P7_0 peripheral function output select bit	0: TXD2/SDA2 output 1: Do not set to this value	RW
PSC_1	Port P7_1 peripheral function output select bit	0: SCL2 output 1: Do not set to this value	RW
PSC_2	Port P7_2 peripheral function output select bit	0: CLK2 output 1: V	RW
PSC_3	Port P7_3 peripheral function output select bit	0: $\overline{\text{RTS2}}$ 1: Do not set to this value	RW
PSC_4	Port P7_4 peripheral function output select bit	0: TA2OUT output 1: Do not set to this value	RW
– (b6-b5)	Reserved bits	Set to 0	RW
PSC_7	Port P10_4 to P10_7 peripheral function input select bit	0: P10_4 to P10_7 or $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ 1: AN_4 to AN_7[1]	RW

NOTE:

1. Set bits ILVL2 to ILVL0 in the KUPIC register to 000b (interrupt disabled) to change the PSC_7 bit. If AN_4 to AN_7 are used with the PSC_7 bit setting to 0, current consumption may increase.

Figure 22.11 PSC Register

Pull-Up Control Register 0[1]



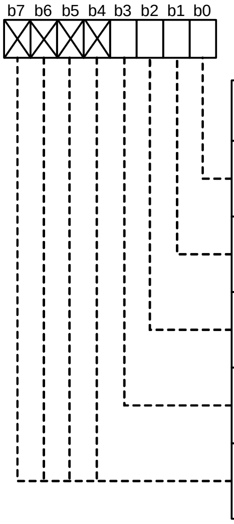
Symbol	Address	After Reset
PUR0	03F0h	00h

Bit Symbol	Bit Name	Function	RW
PU00	P0_0 to P0_3 pull-up	Pull-up setting for the corresponding ports 0: Not pulled up 1: Pulled up	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

NOTE:

1. In memory expansion mode and microprocessor mode, set each bit in the PUR0 register to 0 since port P0 to P5 are used as bus control pins. When using as I/O ports, it can be selected whether the ports are pulled up or not.

Pull-Up Control Register 1[1]



Symbol	Address	After Reset
PUR1	03F1h	XXXX 0000b

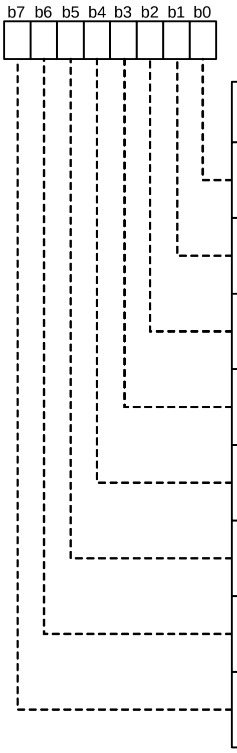
Bit Symbol	Bit Name	Function	RW
PU10	P4_0 to P4_3 pull-up	Pull-up setting for the corresponding ports 0: Not pulled up 1: Pulled up	RW
PU11	P4_4 to P4_7 pull-up		RW
PU12	P5_0 to P5_3 pull-up		RW
PU13	P5_4 to P5_7 pull-up		RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:

1. In memory expansion mode and microprocessor mode, set each bit in the PUR0 register to 0 since port P0 to P5 are used as bus control pins. When using as I/O ports, it can be selected whether the ports are pulled up or not.

Figure 22.12 PUR0 Register, PUR1 Register

Pull-Up Control Register 2



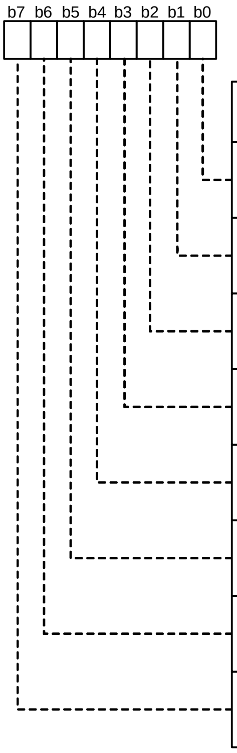
Symbol	Address	After Reset
PUR2	03DAh	00h

Bit Symbol	Bit Name	Function	RW
PU20	P6_0 to P6_3 pull-up	Pull-up setting for the corresponding ports 0: Not pulled up 1: Pulled up	RW
PU21	P6_4 to P6_7 pull-up		RW
PU22	P7_2 to P7_3 pull-up(1)		RW
PU23	P7_4 to P7_7 pull-up		RW
PU24	P8_0 to P8_3 pull-up		RW
PU25	P8_4 to P8_7 pull-up(2)		RW
PU26	P9_0 to P9_3 pull-up		RW
PU27	P9_4 to P9_7 pull-up		RW

NOTES:
1. P7_0 and P7_1 cannot be pulled up.
2. P8_5 cannot be pulled up internally.

Figure 22.13 PUR2 Register

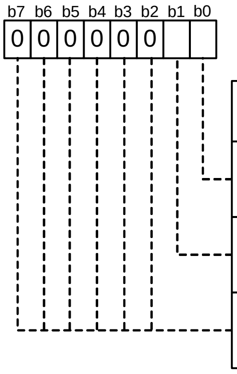
Pull-Up Control Register 3 <144-pin package>



Symbol	Address	After Reset
PUR3	03DBh	00h

Bit Symbol	Bit Name	Function	RW
PU30	P10_0 to P10_3 pull-up	Pull-up setting for the corresponding ports 0: Not pulled up 1: Pulled up	RW
PU31	P10_4 to P10_7 pull-up		RW
PU32	P11_0 to P11_3 pull-up		RW
PU33	P11_4 pull-up		RW
PU34	P12_0 to P12_3 pull-up		RW
PU35	P12_4 to P12_7 pull-up		RW
PU36	P13_0 to P13_3 pull-up		RW
PU37	P13_4 to P13_7 pull-up		RW

Pull-Up Control Register 3 <100-pin package>



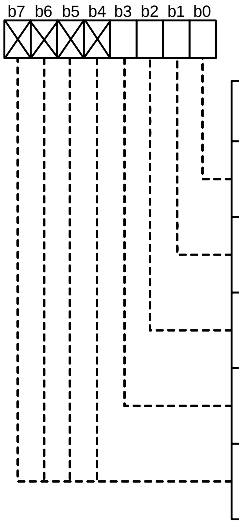
Symbol	Address	After Reset
PUR3	03DBh	00h

Bit Symbol	Bit Name	Function	RW
PU30	P10_0 to P10_3 pull-up	Pull-up setting for the corresponding ports 0: Not pulled up 1: Pulled up	RW
PU31	P10_4 to P10_7 pull-up		RW
– (b7-b2)	Reserved bits	Set to 0	RW

Figure 22.14 PUR3 Register

Pull-Up Control Register 4[1]

Symbol	Address	After Reset
PUR4	03DCh	XXXX 0000b



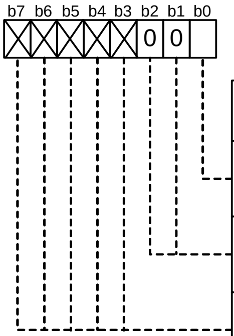
Bit Symbol	Bit Name	Function	RW
PU40	P14_0 to P14_3 pull-up	Pull-up setting for the corresponding ports 0: Not pulled up 1: Pulled up	RW
PU41	P14_4 to P14_6 pull-up		RW
PU42	P15_0 to P15_3 pull-up		RW
PU43	P15_4 to P15_7 pull-up		RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:

1. Set the PUR4 register to 00h in the 100-pin package.

Figure 22.15 PUR4 Register

Port Control Register



Symbol
PCR

Address
03FFh

After Reset
XXXX X000b

Bit Symbol	Bit Name	Function	RW
PCR0	Port P1 control bit[1]	0: CMOS output 1: N-channel open drain output[2]	RW
– (b2-b1)	Reserved bits	Set to 0	RW
– (b7-b3)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:

1. In memory expansion mode and microprocessor mode, set the PCR0 bit to 0 since port P1 is used as data bus. When using port P1 as an I/O port, CMOS or N-channel open drain output can be selected.
2. This function is designed to use port P1 as pseudo open drain by always turning off P channel of the CMOS port . Therefore, the absolute maximum rating of the input voltage is from -0.3 V to VCC2 + 0.3 V.

Figure 22.16 PCR Register

Table 22.1 Unassigned Pin Handling in Single-Chip Mode

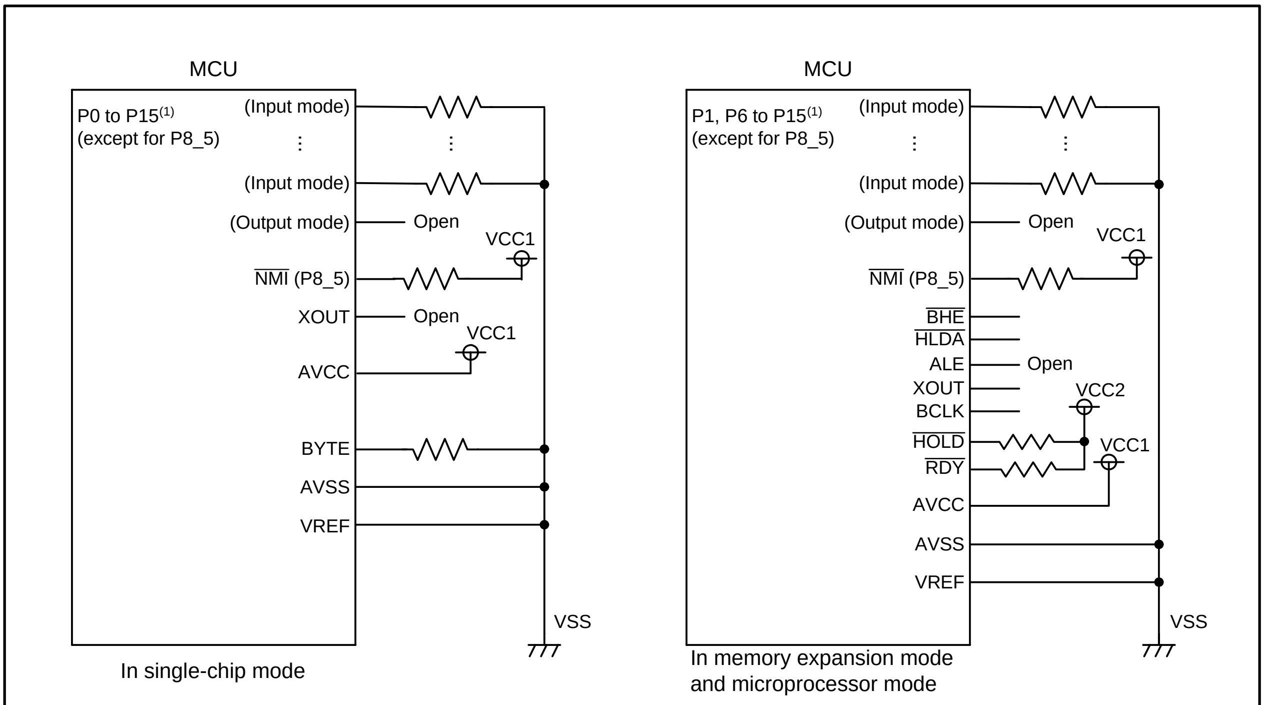
Pin Name	Handling
P0 to P15 (excluding P8_5)[1]	Set pins to input mode and connect each pin to VSS via a resistor (pull-down), or set pins to output mode and leave them open
XOUT[2]	Leave the pin open
NMI (P8_5)	Connect the pin to VCC1 via a resistor (pull-up)
VREF	Connect the pin to VSS

NOTES:
1. P11 to P15 are provided in the 144-pin package only.
2. It is when the external clock is input to the XIN pin.

Table 22.2 Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode

Pin Name	Handling
P1, P6 to P15 (excluding P8_5)[1]	Set pins to input mode and connect each pin to VSS via a resistor (pull-down), or set pins to output mode and leave them open
BHE, ALE, HLDA, XOUT[2], BCLK	Leave the pin open
HOLD, RDY	Connect the pin to VCC2 via a resistor (pull-up)
NMI(P8_5)	Connect the pin to VCC1 via a resistor (pull-up)
VREF	Connect the pin to VSS

NOTES:
1. P11 to P15 are provided in the 144-pin package only.
2. It is when the external clock is applied to the XIN pin.



NOTE:
1. P11 to P15 are provided in the 144-pin package only.

Figure 22.17 Unassigned Pin Handling

Table 22.3 Port P6 Peripheral Function Output Control

	PS0 Register	PSL0 Register
Bit 0	0: P6_0/$\overline{\text{CTS0}}$/$\overline{\text{SS0}}$ 1: Select by the PSL0_0 bit	0: $\overline{\text{RTS0}}$ 1: Do not set to this value
Bit 1	0: P6_1/CLK0 input 1: Select by the PSL0_1 bit	0: CLK0 output 1:Do not set to this value
Bit 2	0: P6_2/RXD0/SCL0 input 1: Select by the PSL0_2 bit	0: SCL0 output 1: STXD0
Bit 3	0: P6_3/SRXD0/SDA0 input 1: Select by the PSL0_3 bit	0: TXD0/SDA0 output 1: Do not set to this value
Bit 4	0: P6_4/$\overline{\text{CTS1}}$/$\overline{\text{SS1}}$ 1: Select by the PSL0_4 bit	0: $\overline{\text{RTS1}}$ 1: Do not set to this value
Bit 5	0: P6_5/CKL1 input 1: Select by the PSL0_5 bit	0: CLK1 output 1: Do not set to this value
Bit 6	0: P6_6/RXD1/SCL1 input 1: Select by the PSL0_6 bit	0: SCL1 output 1: STXD1
Bit 7	0: P6_7/SRXD1/SDA1 input 1: Select by the PSL0_7 bit	0: TXD1/SDA1 output 1: Do not set to this value

Table 22.4 Port P7 Peripheral Function Output Control

	PS1 Register	PSL1 Register	PSC Register
Bit 0	0: P7_0/TA0OUT input/SRXD2/SDA2 input 1: Select by the PSL1_0 bit	0: Select by the PSC_0 bit 1: TA0OUT output	0: TXD2/SDA2 output 1: Do not set to this value
Bit 1	0: P7_1/TA0IN/TB5IN/RXD2/SCL2 input 1: Select by the PSL1_1 bit	0: Select by the PSC_1 bit 1: STXD2	0: SCL2 output 1: Do not set to this value
Bit 2	0: P7_2/TA1OUT input/CLK2 input 1: Select by the PSL1_2 bit	0: Select by the PSC_2 bit 1: TA1OUT output	0: CLK2 output 1: V
Bit 3	0: P7_3/TA1IN/$\overline{\text{CTS2}}$/$\overline{\text{SS2}}$ 1: Select by the PSL1_3 bit	0: Select by the PSC_3 bit 1: $\overline{\text{V}}$	0: $\overline{\text{RTS2}}$ 1: Do not set to this value
Bit 4	0: P7_4/TA2OUT input 1: Select by the PSL1_4 bit	0: Select by the PSC_4 bit 1: W	0: TA2OUT output 1: Do not set to this value
Bit 5	0: P7_5/TA2IN 1: Select by the PSL1_5 bit	0: $\overline{\text{W}}$ 1: Do not set to this value	Set to 0
Bit 6	0: P7_6/TA3OUT input 1: Select by the PSL1_6 bit	0: Do not set to this value 1: TA3OUT output	Set to 0
Bit 7	0: P7_7/TA3IN 1: Do not set to this value	Set to 0	–

Table 22.5 Port P8 Peripheral Function Output Control

	PS2 Register	PSL2 Register
Bit 0	0: P8_0/TA4OUT input 1: Select by the PSL2_0 bit	0: TA4OUT output 1: U
Bit 1	0: P8_1/TA4IN 1: Select by the PSL2_1 bit	0: $\overline{\text{U}}$ 1: Do not set to this value
Bits 2 to 7	Set to 000000b	

Table 22.6 Port P9 Peripheral Function Output Control

	PS3 Register	PSL3 Register
Bit 0	0: P9_0/TB0IN/CLK3 input 1: Select by the PSL3_0 bit	0: CLK3 output 1: Do not set to this value
Bit 1	0: P9_1/TB1IN/RXD3/SCL3 input 1: Select by the PSL3_1 bit	0: SCL3 output 1: STXD3
Bit 2	0: P9_2/TB2IN/SRXD3/SDA3 input 1: Select by the PSL3_2 bit	0: TXD3/SDA3 output 1: Do not set to this value
Bit 3	0: P9_3/TB3IN/$\overline{\text{CTS3}}$/$\overline{\text{SS3}}$/DA0 1: $\overline{\text{RTS3}}$	0: Peripheral function input 1: DA0
Bit 4	0: P9_4/TB4IN/$\overline{\text{CTS4}}$/$\overline{\text{SS4}}$/DA1 1: $\overline{\text{RTS4}}$	0: Peripheral function input 1: DA1
Bit 5	0: P9_5/ANEX0/CLK4 input 1: CLK4 output	0: Peripheral function input except ANEX0 1: ANEX0
Bit 6	0: P9_6/SRXD4/ANEX1/SDA4 input 1: TXD4/SDA4 output	0: Peripheral function input except ANEX1 1: ANEX1
Bit 7	0: P9_7/RXD4/$\overline{\text{ADTRG}}$/SCL4 input 1: Select by the PSL3_7 bit	0: SCL4 output 1: STXD4

Table 22.7 Port P10 Peripheral Function Output Control

	PSC Register
Bit 7	0: P10_4 to P10_7 or $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$ 1: AN_4 to AN_7

23. Flash Memory

CPU rewrite mode, standard serial I/O mode, and parallel I/O mode can be used to erase and program the flash memory.

Table 23.1 lists specifications of the flash memory.

Table 23.1 Flash Memory Specifications

Item	Specification
Erase unit	On a block basis (See Figure 23.1)
Program unit	4 bytes
Erase and program endurance	100 times[1]
Erase and program control method	Software commands control erasing and programming on the flash memory
Number of commands	9 commands
Protect function	• Lock bit protect function (All modes) • ROM code protect function (Parallel I/O mode) • ID code check function (Standard serial I/O mode)
Flash memory stop function	Flash memory can be stopped and initialized
Flash memory rewrite mode	• CPU rewrite mode • Standard serial I/O mode • Parallel I/O mode

NOTE:

1. The erase and program endurance is the number of erase operations performed on individual blocks. For example, if the block A is erased without programming, the erased and program count stands at one for the block A.

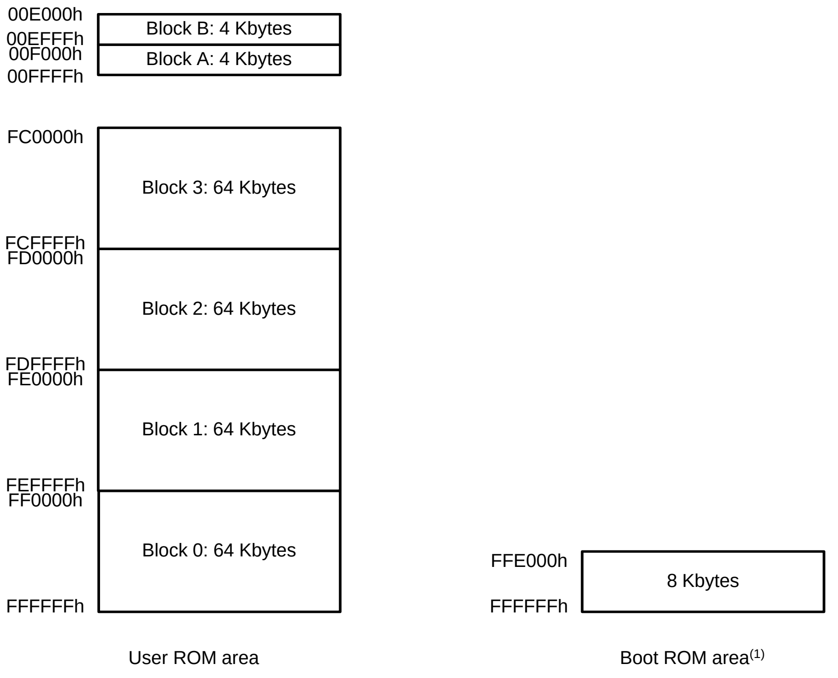
23.1 Memory Map

The user ROM area has an area to store programs, and another 4-Kbyte areas as the block A and the block B for data storage.

The user ROM area can be programmed in CPU rewrite mode, standard serial I/O mode, or parallel I/O mode.

The boot ROM area has one 8-Kbyte block and is allocated in addresses FFE000h to FFFFFFh, which overlap with part of the user ROM area . The rewrite control program for the standard serial I/O mode is stored in the boot ROM area. Do not rewrite the boot ROM area.

Figure 23.1 shows the flash memory map.



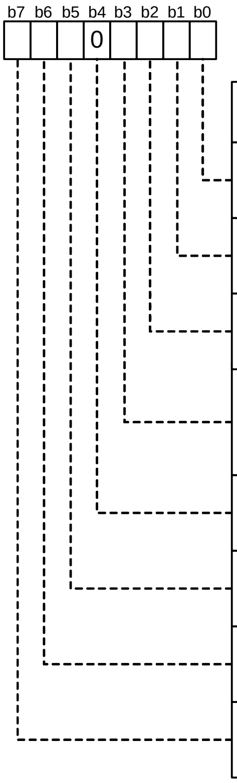
NOTES:
1. The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment.
2. When specifying a block, use the highest-order even address of the specified block.
3. This is a flash memory map in single-chip mode.

Figure 23.1 Flash Memory Map

23.2 Registers

Figures 23.2 to 23.4 show registers associated with the flash memory.

Flash Memory Control Register 0



Symbol	Address	After Reset
FMR0	0057h	0000 0001b

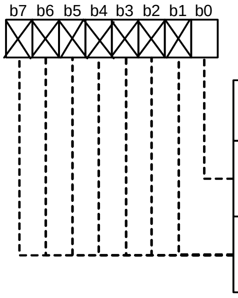
Bit Symbol	Bit Name	Function	RW
FMR00	RY/BY status flag	0: BUSY (programming or erasing in progress)[6] 1: READY	RO
FMR01	CPU rewrite mode select bit[1][7]	0: CPU rewrite mode disabled 1: CPU rewrite mode enabled	RW
FMR02	Lock bit disable select bit[2]	0: Lock bit enabled 1: Lock bit disabled	RW
FMSTP	Flash memory stop bit[3][5]	0: Flash memory started 1: Flash memory stopped (enters low-power consumption state and flash memory is initialized)	RW
– (b4)	Reserved bit	Set to 0	RW
FMR05	User ROM area select bit[3] (available in boot mode only)	0: Boot ROM area accessed 1: User ROM area accessed	RW
FMR06	Program status flag[4]	0: Successfully completed 1: Terminated by error	RO
FMR07	Erase status flag[4]	0: Successfully completed 1: Terminated by error	RO

NOTES:

1. Set bits FMR01 and FMR02 while the $\overline{\text{NMI}}$ pin level is held "H".
2. To set the FMR02 bit to 1, write a 1 to the FMR02 bit immediately after writing a 0 to the bit while the FMR01 bit is set to 1. Write the value in 8-bit units. Do not generate an interrupt or a DMA or DMACII transfer between these two settings.
3. Set bits FMSTP and FMR05 by the program placed in an area other than the flash memory.
4. Bits FMR07 and FMR06 are set to 0 by executing the clear status command.
5. The FMSTP bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode enabled). Bits FMSTP can be set to 1 even when the FMR01 bit is set to 0, but the flash memory does not enter low-power consumption state nor is initialized.
6. Program and read operations by lock bit program command, read lock bit status command, and protect bit program command are included.
7. **To change the FMR01 bit from 0 to 1**, write a 1 to the FMR01 bit immediately after writing a 0 to it. Write the value in 8-bit units. Do not generate an interrupt or a DMA or DMACII transfer between these two settings.
   To change the FMR01 bit from 1 to 0, enter read array mode first, and then write to the address 0057h in 16-bit units. Set the eight high-order bits to 00h.
   e.g., To change the FMR01 bit from 1 to 0;
   Assembly language: mov.w #0000h, 0057h

Figure 23.2 FMR0 Register

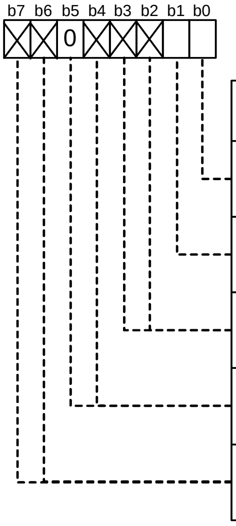
Flash Memory Control Register 2



Symbol	Address	After Reset
FMR2	0052h	XXXX XXX0b

Bit Symbol	Bit Name	Function	RW
FMR20	Protect bit	Writing to the FMR3 register is enabled. 0: Write disable 1: Write enable	RW
– (b7-b1)	Reserved bits	Read as undefined value.	–

Flash Memory Control Register 3[1]



Symbol	Address	After Reset
FMR3	0050h	XX0X XX00b

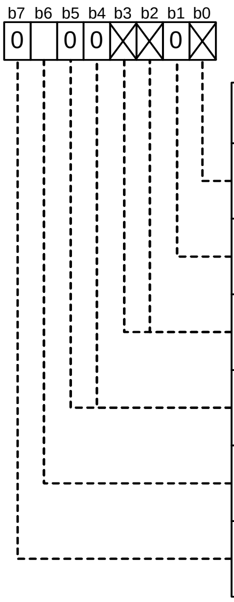
Bit Symbol	Bit Name	Function	RW
FMR30	EW1 mode select bit	0: EW0 mode 1: EW1 mode	RW
FMR31	Lock bit read setting bit	0: Read via data bus 1: Read by the FMR16 bit in the FMR1 register	RW
– (b4-b2)	Reserved bits	Read as undefined value.	RO
– (b5)	Reserved bit	Set to 0	RW
– (b7-b6)	Reserved bits	Read as undefined value.	RO

NOTE:

1. Write the FMR3 register after the FMR01 bit in the FMR0 register is set to 1 (CPU rewrite mode enabled) and the FMR20 bit in the FMR2 register is set to 1 (write enable). After exiting wait mode or stop mode, the FMR3 register becomes the value after reset.

Figure 23.3 FMR2 and FMR3 Registers

Flash Memory Control Register 1



Symbol	Address	After Reset
FMR1	0055h	0000 XX0Xb

Bit Symbol	Bit Name	Function	RW
– (b0)	Reserved bit	Read as undefined value.	–
– (b1)	Reserved bit	Set to 0	RW
– (b3-b2)	Reserved bits	Read as undefined value.	–
– (b5-b4)	Reserved bits	Set to 0	RW
FMR16	Lock bit status flag[1]	0: Locked 1: Unlocked	RO
– (b7)	Reserved bit	Set to 0	RW

NOTE:

1. The FMR16 bit is enabled when the FMR31 bit in the FMR3 register is set to 1 (Read by the FMR16 bit in the FMR1 register).

Figure 23.4 FMR1 Register

23.3 Protect Function

There are two types of protect function. One is to protect data from accidentally erase or program on the flash memory, which is provided by the lock bit protect function. The other is to prevent the program code from being leaked to the third party, which is provided by the ROM code protect function and the ID code check function.

23.3.1 Lock bit Protect Function

The lock bit protect function is used in any flash memory rewrite mode. This function provides protection against erase or program on a block basis. To use the lock bit protect function, set the FMR02 bit in the FMR0 register to 0 (lock bit enabled).

Each block in the flash memory has the lock bit. When the lock bit is set to 0 (locked), the block cannot be erased nor programmed. To set the lock bit to 0, execute the lock bit program command in the software command.

The FMR31 bit in the FMR3 register determines whether the lock bit status is read via the data bus or by the FMR16 bit in the FMR1 register. When the lock bit program command is executed, the lock bit status is read via the data bus if the FMR31 bit is set to 0, or is stored in the FMR16 bit if the FMR31 bit is set to 1.

To disable the lock bit protect function, set the FMR02 bit in the FMR0 register to 1 (lock bit disabled). The FMR02 bit diables the lock bit protect function without changing the lock bit status. When the FMR02 bit is set to 1, all the blocks can be erased and programmed regardless of the lock bit status. When the block erase command is executed while the FMR02 bit is set to 1, the lock bit data as well as data in the block are erased and the lock bit becomes 1 (unlocked).

23.3.2 ROM Code Protect Function

The ROM code protect function is used in parallel I/O mode. This function provides protection against read and program to all blocks. Each block has two protect bits. Table 23.2 lists addresses of the protect bits. If any of these protect bits is set to 0 (protected), all blocks becomes protected and the parallel programmer cannot read nor program to any areas in the flash memory. To set the protect bit to 0, execute the protect bit program command. To enhance security, set all the protect bits in the flash memory to 0 when using the ROM code protect function.

The protect bit status is read via the data bus by executing the read protect bit status command.

To disable the ROM code protect function, erase all the blocks whose protect bit is set to 0 by executing the block erase command. All the protect bits of the erased blocks become 1 (unprotected) and the ROM code protect function is disabled.

Table 23.2 Address of Protect Bit

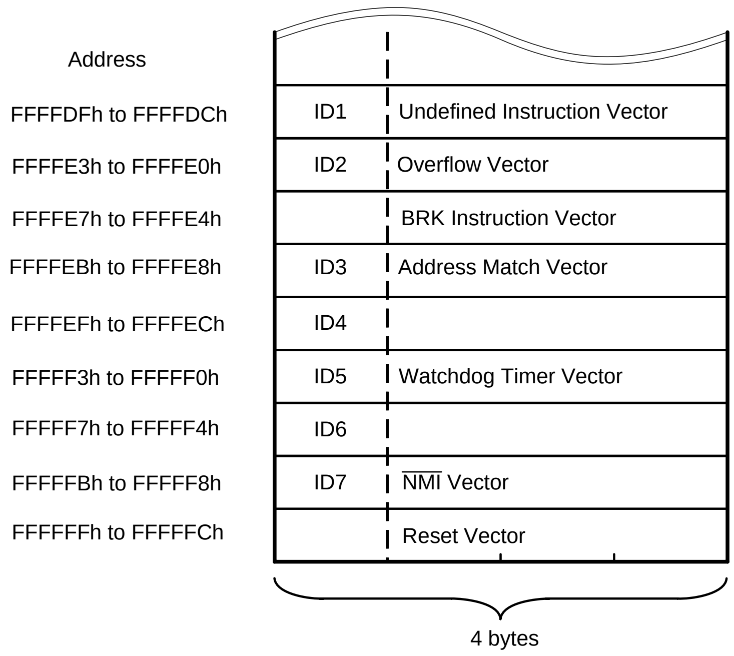
Block	Protect bit 1	Protect bit 0
Block B	00E300h	00E100h
Block A	00F300h	00F100h
Block 3	FC0300h	FC0100h
Block 2	FD0300h	FD0100h
Block 1	FE0300h	FE0100h
Block 0	FF0300h	FF0100h

23.3.3 ID Code Check Function

The ID code check function is used in standard serial I/O mode. The ID code sent from the serial programmer and the ID code written in the user ROM area of the flash memory are checked to see if they match. If these ID codes do not match, the commands sent from the serial programmer are not accepted. However, if the four bytes of the reset vector are set to FFFFFFFFh(1), the ID codes are not checked and all commands can be accepted.
The ID code is 7-byte data stored consecutively, beginning with the first byte, into addresses 0FFFFDFh, 0FFFFE3h, 0FFFFEBh, 0FFFFEFh, 0FFFFF3h, 0FFFFF7h, and 0FFFFFBh. To use the ID code check function, write the program which specifies the ID code to these addresses.

NOTE:

1. FFFFFFFFh is the factory default setting.



Address		
FFFFDFh to FFFFDCh	ID1	Undefined Instruction Vector
FFFFE3h to FFFFE0h	ID2	Overflow Vector
FFFFE7h to FFFFE4h		BRK Instruction Vector
FFFFEBh to FFFFE8h	ID3	Address Match Vector
FFFFEFh to FFFFECh	ID4	
FFFFF3h to FFFFF0h	ID5	Watchdog Timer Vector
FFFFF7h to FFFFF4h	ID6	
FFFFFBh to FFFFF8h	ID7	$\overline{\text{NMI}}$ Vector
FFFFFFh to FFFFFCh		Reset Vector

Figure 23.5 Addresses for Stored ID Codes

23.4 Flash Memory Stop Function

When the FMSTP bit in the FMR0 register is set to 1 (flash memory stopped), the flash memory control circuit stops and as a result power consumption in the flash memory can be reduced. When the FMSTP bit is set to 1 from 0 (flash memory started), the flash memory control circuit is initialized. Access to the flash memory is disabled when the FMSTP bit is set to 1. Set the FMSTP bit to 1 by the program placed in an area other than the flash memory.

Set the FMSTP bit to 1 in one of the following cases in accordance with the procedure shown in Figure 23.6.

- A flash memory access error occurs while erasing or programming in EW0 mode (the FMR00 bit does not switch back to 1 (ready)).
- To further reduce power consumption in low-power consumption mode or on-chip oscillator low-power consumption mode.

The flash memory is automatically turned off when entering wait mode or stop mode, and turned back on when exiting wait mode or stop mode. Set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) before entering wait mode or stop mode.

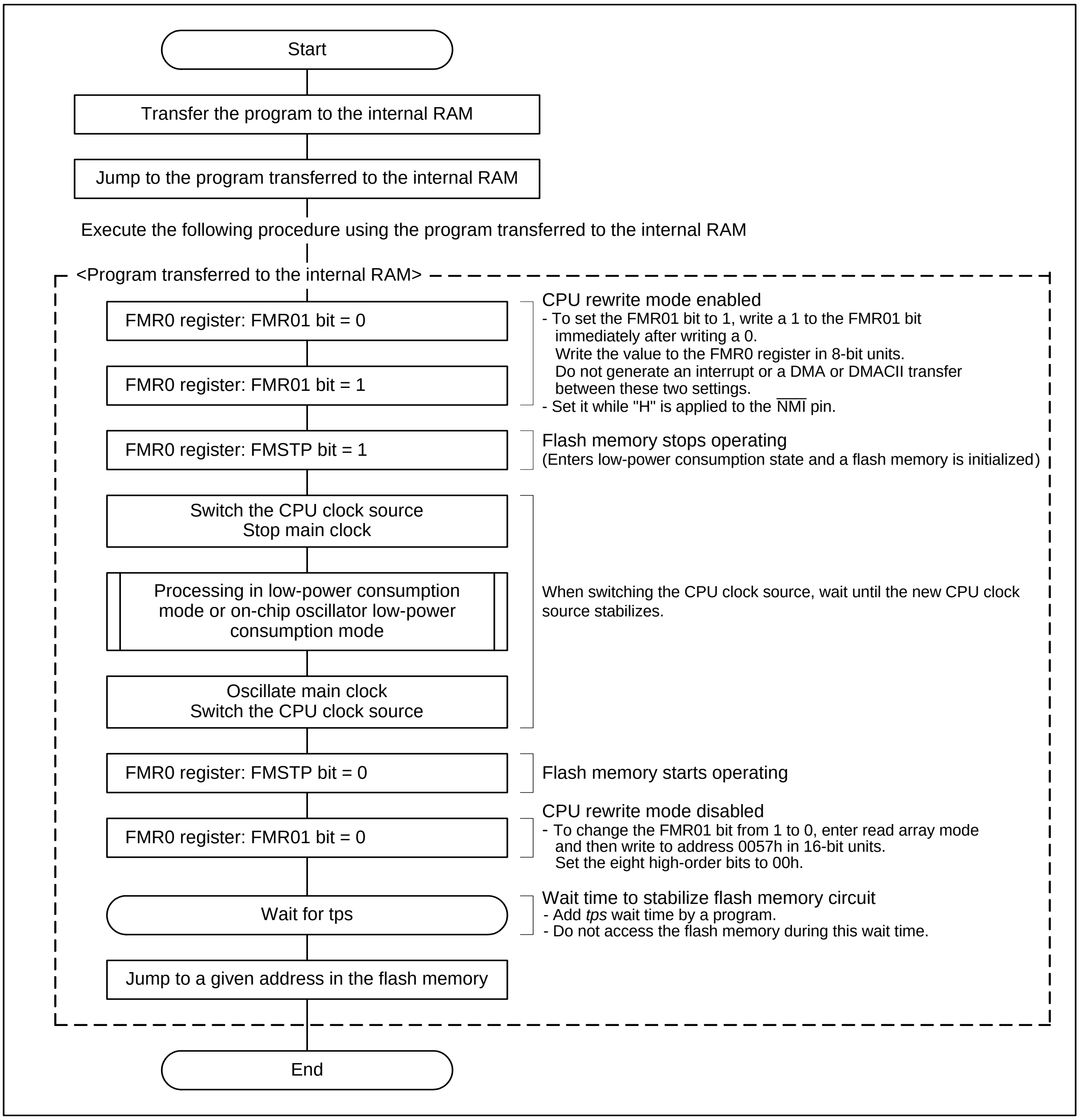

Start
Transfer the program to the internal RAM
Jump to the program transferred to the internal RAM
Execute the following procedure using the program transferred to the internal RAM
<Program transferred to the internal RAM>
FMR0 register: FMR01 bit = 0
FMR0 register: FMR01 bit = 1
CPU rewrite mode enabled
- To set the FMR01 bit to 1, write a 1 to the FMR01 bit immediately after writing a 0. Write the value to the FMR0 register in 8-bit units. Do not generate an interrupt or a DMA or DMACII transfer between these two settings.
- Set it while "H" is applied to the $\overline{NMI}$ pin.
FMR0 register: FMSTP bit = 1
Flash memory stops operating
(Enters low-power consumption state and a flash memory is initialized)
Switch the CPU clock source
Stop main clock
Processing in low-power consumption mode or on-chip oscillator low-power consumption mode
Oscillate main clock
Switch the CPU clock source
When switching the CPU clock source, wait until the new CPU clock source stabilizes.
FMR0 register: FMSTP bit = 0
Flash memory starts operating
FMR0 register: FMR01 bit = 0
CPU rewrite mode disabled
- To change the FMR01 bit from 1 to 0, enter read array mode and then write to address 0057h in 16-bit units. Set the eight high-order bits to 00h.
Wait for tps
Wait time to stabilize flash memory circuit
- Add *tps* wait time by a program.
- Do not access the flash memory during this wait time.
Jump to a given address in the flash memory
End

Figure 23.6 Procedure to Stop Flash Memory

23.5 Boot Mode

Use the following procedure to enter boot mode and a program in the boot ROM area is executed.

(1) Apply an “L” (pull-down) to the P6_5 pin or apply an “H” (pull-up) to the P6_7 pin
(2) Apply an “L” (pull-down) to the $\overline{\text{EPM}}$ (P5_5) pin and apply an “H” (pull-up) to the $\overline{\text{CE}}$ (P5_0) pin
(3) Apply an “H” to the CNVSS pin
(4) Perform a hardware reset

When switching from the boot ROM area to the user ROM area, set the FMR05 bit in the FMR0 register to 1 (access the user ROM area) by the program placed in the area other than the flash memory.

The rewrite control program for standard serial I/O mode is stored in the boot ROM area in the factory default configuration. Do not rewrite the boot ROM area.

23.6 Flash Memory Rewrite Mode

CPU rewrite mode, standard serial I/O mode, and parallel I/O mode can be used to erase and program the flash memory. Table 23.3 lists overview of flash memory rewrite mode.

Table 23.3 Flash Memory Rewrite Mode Overview

Flash Memory Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode[1]
Function	User ROM area is programmed by the CPU writing software commands. EW0 mode: Execute the rewrite control program placed in an area other than the flash memory. EW1 mode: Execute the rewrite control program placed in the flash memory.	User ROM area is programmed using a dedicated serial programmer. Standard serial I/O mode 1: Clock synchronous mode in UART1 Standard serial I/O mode 2: Clock asynchronous mode in UART1	User ROM area is programmed using a dedicated parallel programmer.
Rewritable area	User ROM area	User ROM area	User ROM area
Operating mode	Single-chip mode Memory expansion mode (EW0 mode)	Boot mode	Parallel I/O mode
ROM programmer	–	Serial programmer	Parallel programmer

NOTE:

1. In parallel I/O mode, the boot ROM area can be programmed. However, do not rewrite the boot ROM area since the rewrite control program for standard serial I/O mode is stored in the boot ROM area in the factory default configuration.

23.6.1 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be programmed by the CPU writing software commands with the MCU mounted on a board. In CPU rewrite mode, only the user ROM area shown in Figure 23.1 can be programmed. The boot ROM area cannot be rewritten. EW0 mode and EW1 mode are provided as CPU rewrite mode. Prior to accessing registers FMR0 to FMR3 or to entering CPU rewrite mode (EW0, EW1 mode), set the CPU clock frequency to 10 MHz or lower using bits MCD4 to MCD0 in the MCD register, and also set the PM12 bit in the PM1 register to 1 (1 wait state).

Table 23.4 lists specifications of EW0 mode and EW1 mode. Figure 23.7 shows a setting procedure for EW0 mode. Figure 23.8 shows a setting procedure for EW1 mode.

Table 23.4 Specifications of EW0 Mode and EW1 Mode

Item	EW0 Mode	EW1 Mode
Operation	• Program the user ROM area by executing the rewrite control program placed in an area other than the flash memory.	• Erase and program a block where the rewrite control program is not placed, by executing the rewrite control program placed in the user ROM area.
Processor mode	• Single-chip mode • Memory expansion mode	• Single-chip mode
Areas where a rewrite program can be stored	• User ROM area	• User ROM area
Software command	All commands are available	• Do not execute the following commands to the block storing the rewrite control program: program, block erase, lock bit program, protect bit program • Do not execute the following commands: read status register, read lock bit status(4), read protect bit status
Flash memory mode after erasing or programming	Read status register mode	Read array mode
Flash memory status detection	• Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program • Execute the read status register command to read bits SR7, SR5, and SR4 in the SRD register	• Read bits FMR00, FMR06, and FMR07 in the FMR0 register by a program
CPU status during erase or program operation(1)	Operating	In a hold state (CPU stops) (I/O port maintains the status which is before executing a command)
Peripheral interrupt request, DMA request, and DMACII request during erase or program operation	Acknowledged(2)	Not acknowledged(3)

NOTES:

1. In both the EW0 mode and EW1 mode, when an $\overline{\text{NMI}}$ interrupt or watchdog timer interrupt is generated, the erase or program operation in progress is aborted and the interrupt is acknowledged.
2. To use peripheral function interrupts, place interrupt routine programs and the relocatable vector table in an area other than flash memory.
3. Do not generate an interrupt (except $\overline{\text{NMI}}$ interrupt and watchdog timer interrupt) or a DMA or DMACII transfer during erase or program operation.
4. When the FMR31 bit in the FMR3 register is 0 (read through the data bus).

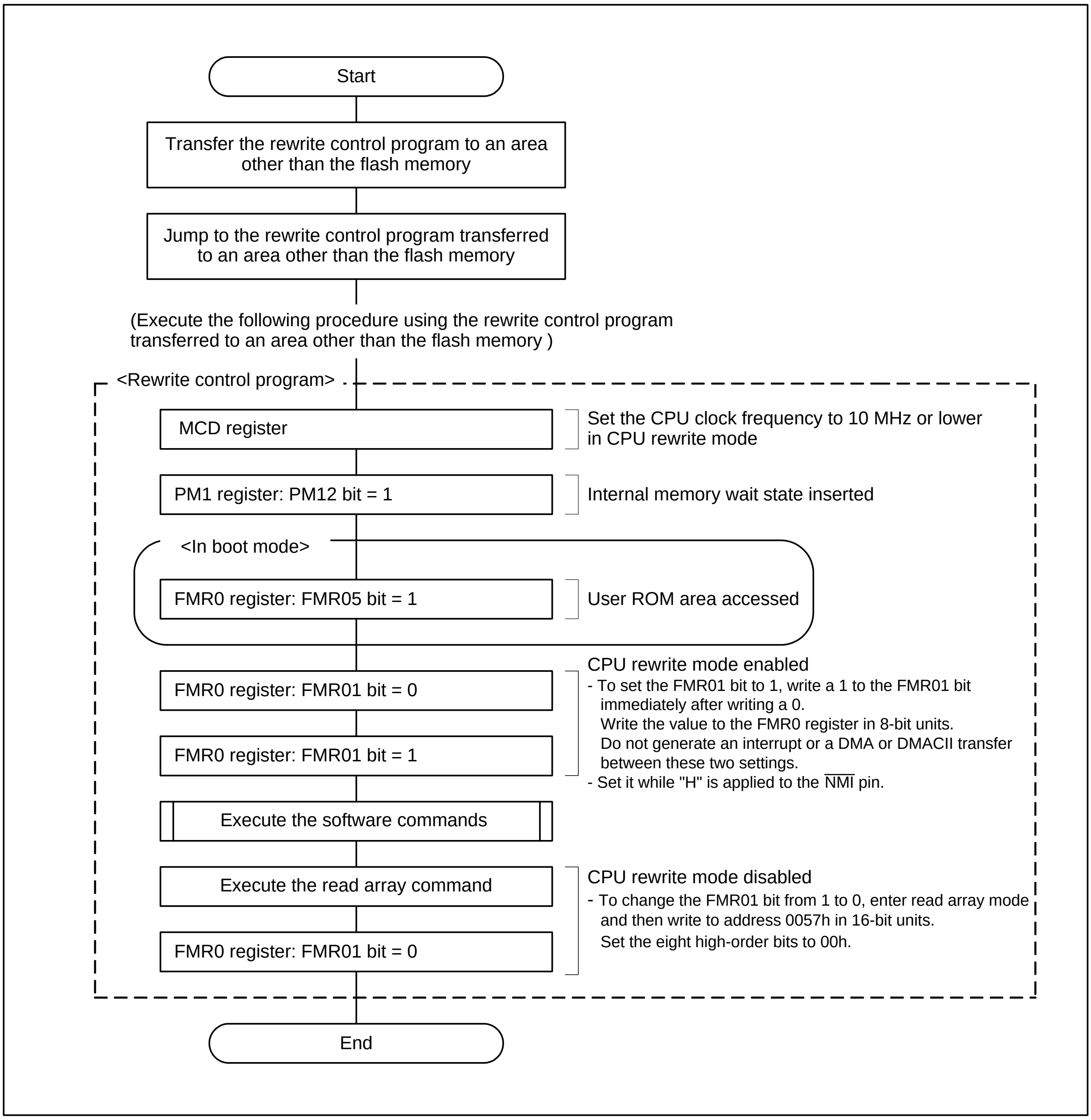
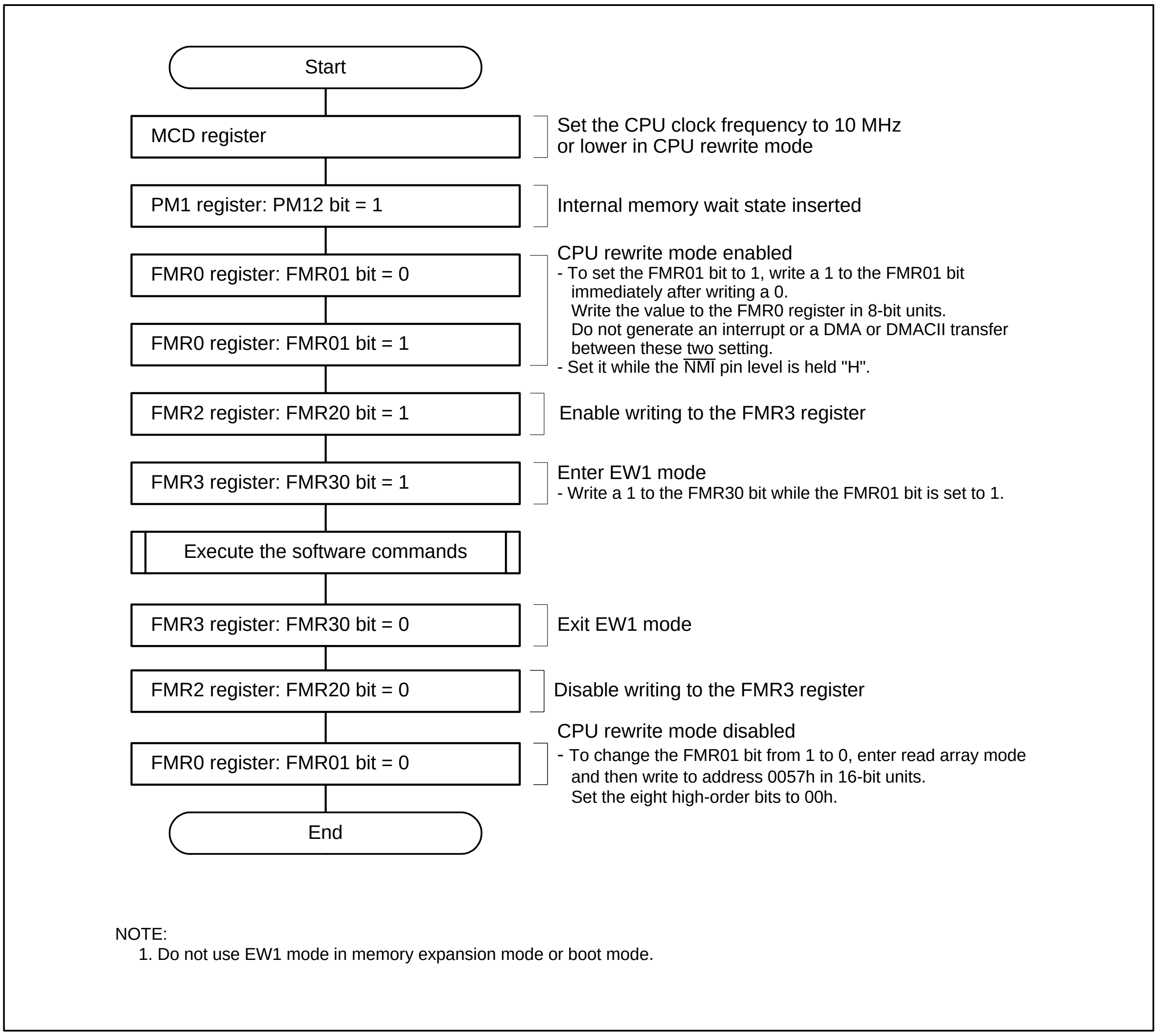

Start
Transfer the rewrite control program to an area other than the flash memory
Jump to the rewrite control program transferred to an area other than the flash memory
(Execute the following procedure using the rewrite control program transferred to an area other than the flash memory)
<Rewrite control program>
MCD register
Set the CPU clock frequency to 10 MHz or lower in CPU rewrite mode
PM1 register: PM12 bit = 1
Internal memory wait state inserted
<In boot mode>
FMR0 register: FMR05 bit = 1
User ROM area accessed
FMR0 register: FMR01 bit = 0
FMR0 register: FMR01 bit = 1
CPU rewrite mode enabled
- To set the FMR01 bit to 1, write a 1 to the FMR01 bit immediately after writing a 0.
Write the value to the FMR0 register in 8-bit units.
Do not generate an interrupt or a DMA or DMACII transfer between these two settings.
- Set it while "H" is applied to the $\overline{\text{NMI}}$ pin.
Execute the software commands
Execute the read array command
FMR0 register: FMR01 bit = 0
CPU rewrite mode disabled
- To change the FMR01 bit from 1 to 0, enter read array mode and then write to address 0057h in 16-bit units.
Set the eight high-order bits to 00h.
End

Figure 23.7 Setting Procedure for EW0 Mode


Start
MCD register
PM1 register: PM12 bit = 1
FMR0 register: FMR01 bit = 0
FMR0 register: FMR01 bit = 1
FMR2 register: FMR20 bit = 1
FMR3 register: FMR30 bit = 1
Execute the software commands
FMR3 register: FMR30 bit = 0
FMR2 register: FMR20 bit = 0
FMR0 register: FMR01 bit = 0
End

Set the CPU clock frequency to 10 MHz or lower in CPU rewrite mode

Internal memory wait state inserted

CPU rewrite mode enabled
- To set the FMR01 bit to 1, write a 1 to the FMR01 bit immediately after writing a 0. Write the value to the FMR0 register in 8-bit units. Do not generate an interrupt or a DMA or DMACII transfer between these two setting.
- Set it while the $\overline{\text{NMI}}$ pin level is held "H".

Enable writing to the FMR3 register

Enter EW1 mode
- Write a 1 to the FMR30 bit while the FMR01 bit is set to 1.

Exit EW1 mode

Disable writing to the FMR3 register

CPU rewrite mode disabled
- To change the FMR01 bit from 1 to 0, enter read array mode and then write to address 0057h in 16-bit units. Set the eight high-order bits to 00h.

NOTE:
1. Do not use EW1 mode in memory expansion mode or boot mode.

Figure 23.8 Setting Procedure for EW1 Mode

23.6.1.1 Software Commands

Write commands or read and write data to the specified even addresses in the user ROM area in 16-bit units. When writing a command code, 8 high-order bits (D15 to D8) are ignored.

Table 23.5 Software Commands

Software Command	First Bus Cycle			Second Bus Cycle			Third Bus Cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read array	Write	FA	xxFFh	–	–	–	–	–	–
Read status register	Write	FA	xx70h	Read	FA	SRD	–	–	–
Clear status register	Write	FA	xx50h	–	–	–	–	–	–
Program	Write	WA0	xx41h	Write	WA0	WD0	Write	WA1	WD1
Block erase	Write	FA	xx20h	Write	BA0	xxD0h	–	–	–
Lock bit program	Write	BA0	xx77h	Write	BA0	xxD0h	–	–	–
Read lock bit status(1)	Write	FA	xx71h	Write	BA0	xxD0h	–	–	–
Read lock bit status(2)	Write	FA	xx71h	Read	BA1	RD0	–	–	–
Protect bit program	Write	PBA	xx67h	Write	PBA	xxD0h	–	–	–
Read protect bit status	Write	FA	xx61h	Read	PBA	RD1	–	–	–

FA: Any even address in the user ROM area
WA0: 16 low-order bits of write address

- Set the lowest 2-bit of the address to 00b.
- The address specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

WA1: 16 high-order bits of write address

- Set the lowest 2-bit of the address to 10b.
- Specify WA0 and WA1 in the consecutive even addresses.

BA0: Highest-order even address of a block
BA1: Any even address of a block
PBA: The protect bit address (See table 23.2)
WD0: 16 low-order bit of write data
WD1: 16 high-order bit of write data
RD0: Read data (bit 6 is the lock bit data)
RD1: Read data (bit 6 is the protect bit data)
SRD: Data in the Status Register (b7 to b0)
xx: 8 high-order bits of command code (ignored)

NOTES:

1. When the FMR31 bit in the FMR3 register is set to 1 (read by the FMR16 bit in the FMR1 register).
2. When the FMR31 bit in the FMR3 register is set to 0 (read via the data bus).

(1) Read Array Command

The read array command is used to read the flash memory.

The flash memory enters read array mode when the command code xxFFh is written in the first bus cycle. The content of the specified address can be read in 16-bit units when a read address is specified after the next bus cycle. The flash memory remains in read array mode until the other command is written. Therefore, the contents of multiple addresses can be read in succession.

(2) Read Status Register Command

The read status register command is used to read the Status Register. When the command code xx70h is written in the first bus cycle, the Status Register can be read after the second bus cycle (refer to **23.6.1.2 Status Register** for details). To read the Status Register, read an even address in the user ROM area. Do not execute the read status register command in EW1 mode.

(3) Clear Status Register Command

The clear status register command is used to clear the Status Register. When the command code xx50h is written in the first bus cycle, bits FMR07 and FMR06 in the FMR0 register become 00b and bits SR5 and SR4 in the Status Register become 00b.

(4) Program Command

The program command is used to write data to the flash memory in 4-byte units.

A program operation (program and verify data) starts by writing the command code xx41h in the first bus cycle, and data to the 16 low-order bits of write address in the second bus cycle and to the 16 high-order bits of write address in the third bus cycle. The address value specified in the first bus cycle must be the same even address as the 16 low-order bits of write address specified in the second bus cycle. Specify the 16 low-order bits of write address and the 16 high-order bits of write address in the consecutive even addresses.

The FMR00 bit in the FMR0 register can be used to determine whether a program operation has been completed or not. The FMR00 bit becomes 0 (busy) during the program operation and becomes 1 (ready) when the program operation is completed.

After a program operation is completed, the FMR06 bit in the FMR0 register is used to determine whether a program operation is completed successfully or not. (Refer to **23.6.1.3 Error Check** for details.)

Do not execute the program command to the same address more than once without executing the block erase command. Figure 23.9 shows a flow chart of the program command.

The lock bit can protect each block from being programmed inadvertently. (Refer to **23.3.1 Lock bit Protect Function** for details.)

In EW1 mode, do not execute the program command to the block where the rewrite control program is stored.

In EW0 mode, the flash memory enters read status register mode when a program operation starts.

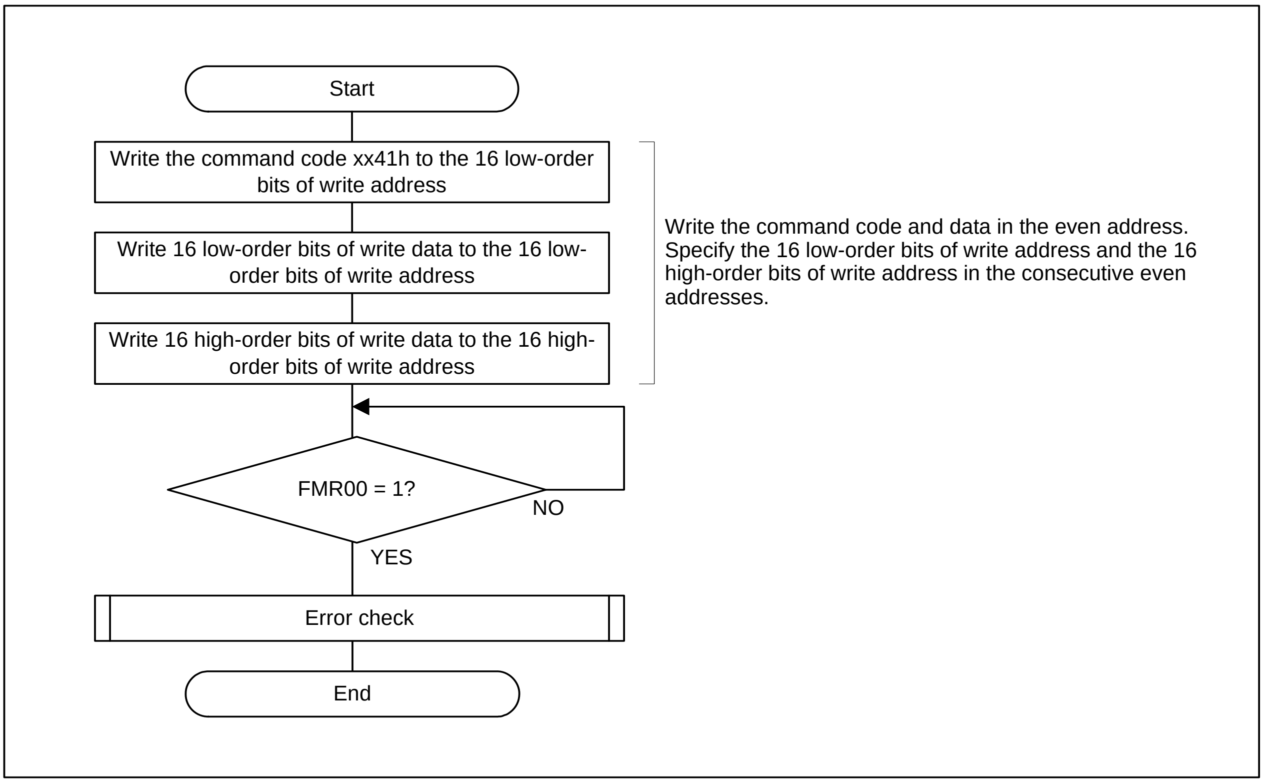

Start
Write the command code xx41h to the 16 low-order bits of write address
Write 16 low-order bits of write data to the 16 low-order bits of write address
Write 16 high-order bits of write data to the 16 high-order bits of write address
Write the command code and data in the even address. Specify the 16 low-order bits of write address and the 16 high-order bits of write address in the consecutive even addresses.
FMR00 = 1?
NO
YES
Error check
End

Figure 23.9 Program Command

(5) Block Erase Command

By writing the command code xx20h in the first bus cycle and xxD0h to the highest-order even address of a block to be erased in the second bus cycle, an erase operation (erase and verify) starts on the specified block.
The FMR00 bit in the FMR0 register can be used to determine whether an erase operation has been completed or not. The FMR00 bit becomes 0 (busy) during the erase operation, and becomes 1 (ready) when the erase operation is completed.
After the erase operation is completed, the FMR07 bit in the FMR0 register is used to determine whether the erase operation is completed successfully or not. (Refer to **23.6.1.3 Error Check** for details.)
Figure 23.10 shows a flow chart of block erase command.
The lock bit can protect each block from being erased inadvertently. (Refer to **23.3.1 Lock bit Protect Function** for details.)
In EW1 mode, do not execute the block erase command to the block where the rewrite control program is stored.
In EW0 mode, the flash memory enters read status register mode when an erase operation starts.

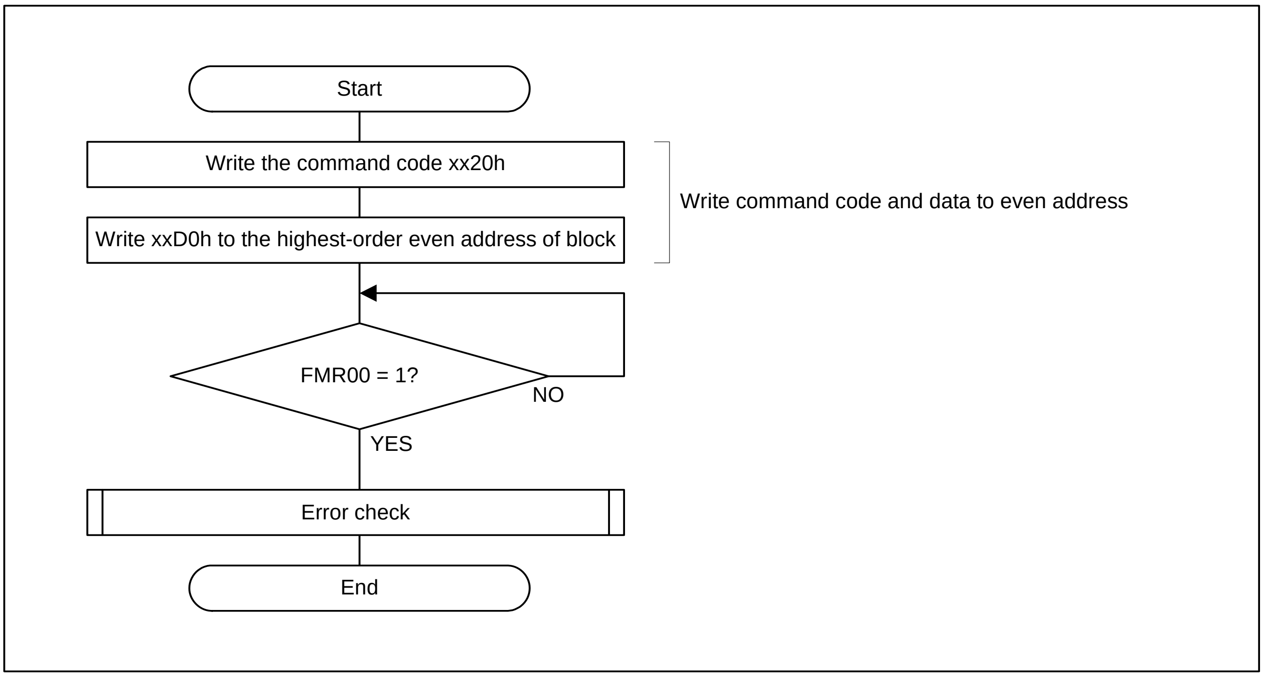

Start
Write the command code xx20h
Write xxD0h to the highest-order even address of block
Write command code and data to even address
FMR00 = 1?
NO
YES
Error check
End

Figure 23.10 Block Erase Command

(6) Lock Bit Program Command

The lock bit program command is used to set the lock bit of a given block to 0 (locked).

By writing the command code xx77h in the first bus cycle and xxD0h to the highest-order even address of a block to be locked in the second bus cycle, the lock bit of the specified block becomes 0. The address specified in the first bus cycle must be the same highest-order even address of the block specified in the second bus cycle.

Figure 23.11 shows a flow chart of lock bit program command. Execute the read lock bit status command to read lock bit status (lock bit data).

The FMR00 bit in the FMR0 register can be used to determine whether a lock bit program operation has been completed or not.

Refer to **23.3.1 Lock bit Protect Function** for information on lock bit functions and how to set it to 1 (unlocked).

In EW1 mode, do not execute the lock bit program command to the block where the rewrite control program is stored.

In EW0 mode, the flash memory enters read status register mode when a program operation starts.

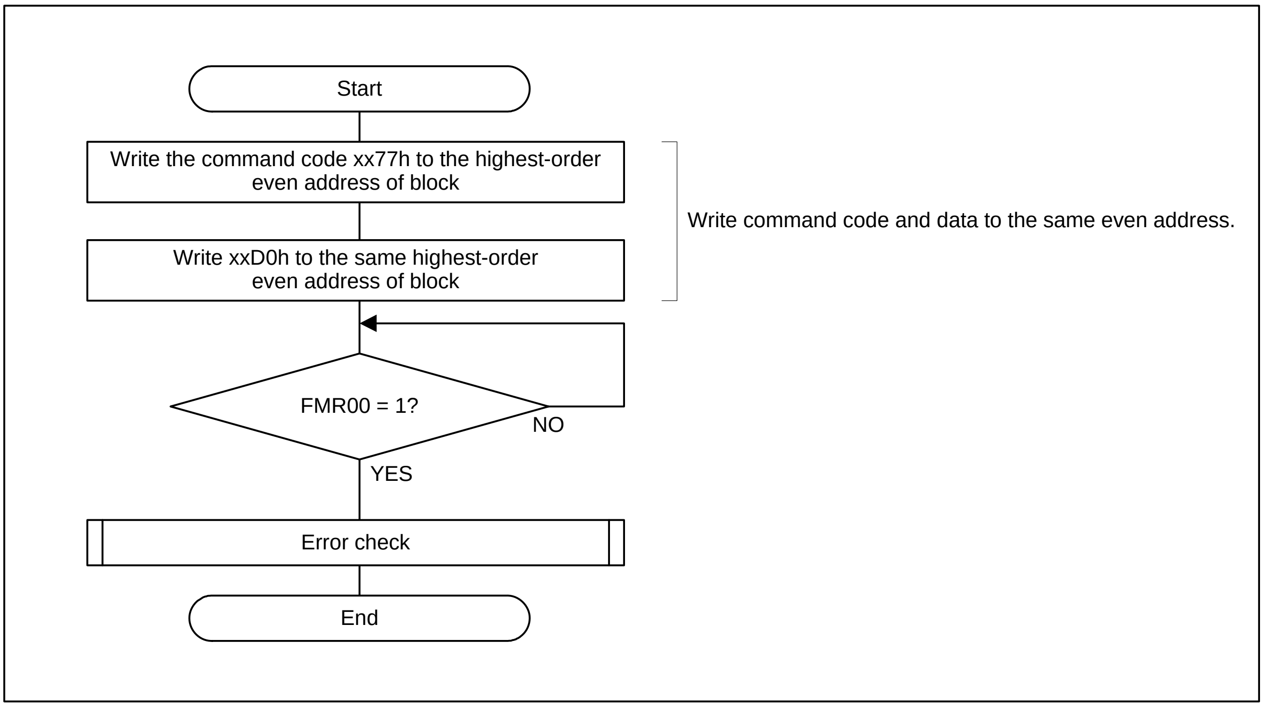

Start
Write the command code xx77h to the highest-order even address of block
Write xxD0h to the same highest-order even address of block
Write command code and data to the same even address.
FMR00 = 1?
NO
YES
Error check
End

Figure 23.11 Lock Bit Program Command

(7) Read Lock Bit Status Command

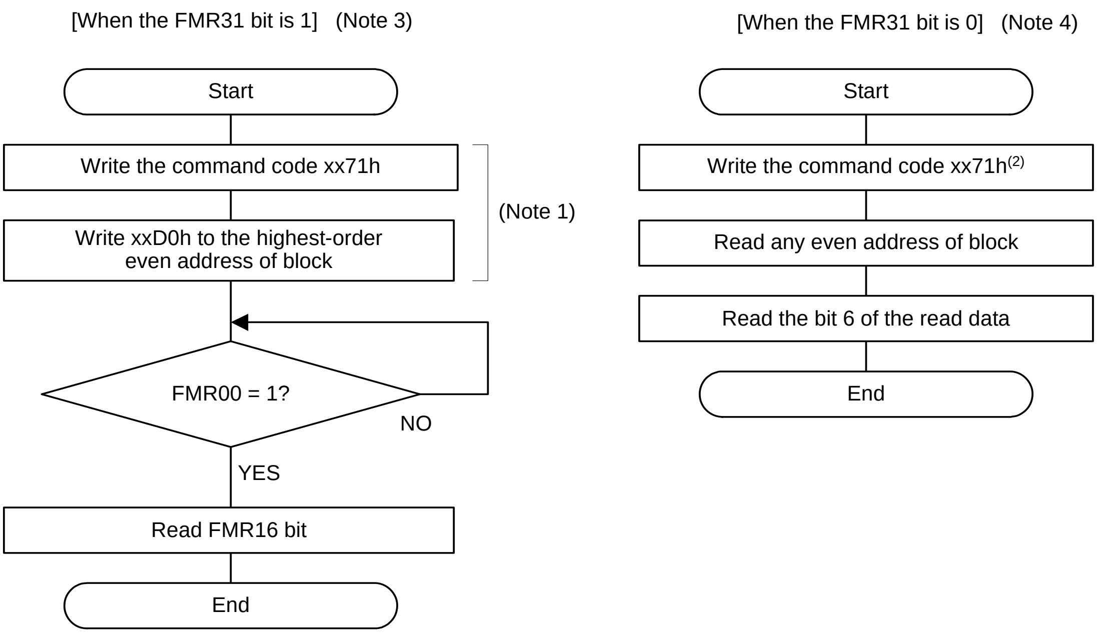
The read lock bit status command reads a lock bit status of a given block.

If the FMR31 bit in the FMR3 register is set to 1 (read by the FMR16 bit in the FMR1 register), the lock bit status of the specified block is stored into the FMR16 bit by writing the command code xx71h in the first bus cycle and xxD0h to the highest-order even address of the block in the second bus cycle. Read the FMR16 bit after the FMR00 bit in the FMR0 register becomes 1 (ready).

If the FMR31 bit in the FMR3 register is set to 0 (read via the data bus), execute the read lock bit status command by the program placed in the RAM. By writing xx71h in the first bus cycle and reading any even address of the specified block in the second bus cycle, the lock bit status of the block can be read by the bit 6 of the read data. When the bit 6 is 0, the block is locked; when the bit 6 is 1, the block is unlocked.

Figure 23.12 shows a flow chart of read lock bit status command.

When the FMR31 bit is set to “0”, do not execute the read lock bit status command in EW1 mode.


[When the FMR31 bit is 1] (Note 3)
Start
Write the command code xx71h
Write xxD0h to the highest-order even address of block
(Note 1)
FMR00 = 1?
NO
YES
Read FMR16 bit
End
[When the FMR31 bit is 0] (Note 4)
Start
Write the command code xx71h(2)
Read any even address of block
Read the bit 6 of the read data
End

NOTES:
1. Write command code and data to even address.
2. Write command code and data by the program placed in the RAM.
3. The flash memory enters read array mode after this command is executed.
4. The flash memory enters read lock bit status mode after this command is executed. A lock bit status can be read in succession by reading any even address of the specified block.

Figure 23.12 Read Lock Bit Status Command

(8) Protect Bit Program Command

The protect bit program command is used to set the protect bit of a given block to 0 (protected).

By writing the command code xx67h in the first bus cycle and xxD0h to the address of the protect bit in the second bus cycle, the protect bit of the specified block becomes 0. The address specified in the first bus cycle must be the same address of the protect bit specified in the second bus cycle.

Figure 23.13 shows a flow chart of protect bit program command. Execute the read protect bit status command to read protect bit status (protect bit data).

The FMR00 bit in the FMR0 register can be used to determine whether a protect bit program operation has been completed or not.

Refer to **23.3.2 ROM Code Protect Function** for addresses of the protect bits, information on protect bit functions, and how to set it to 1 (unprotected).

In EW1 mode, do not execute the protect bit program command to the block where the rewrite control program is stored.

In EW0 mode, the flash memory enters read status register mode when a program operation starts.

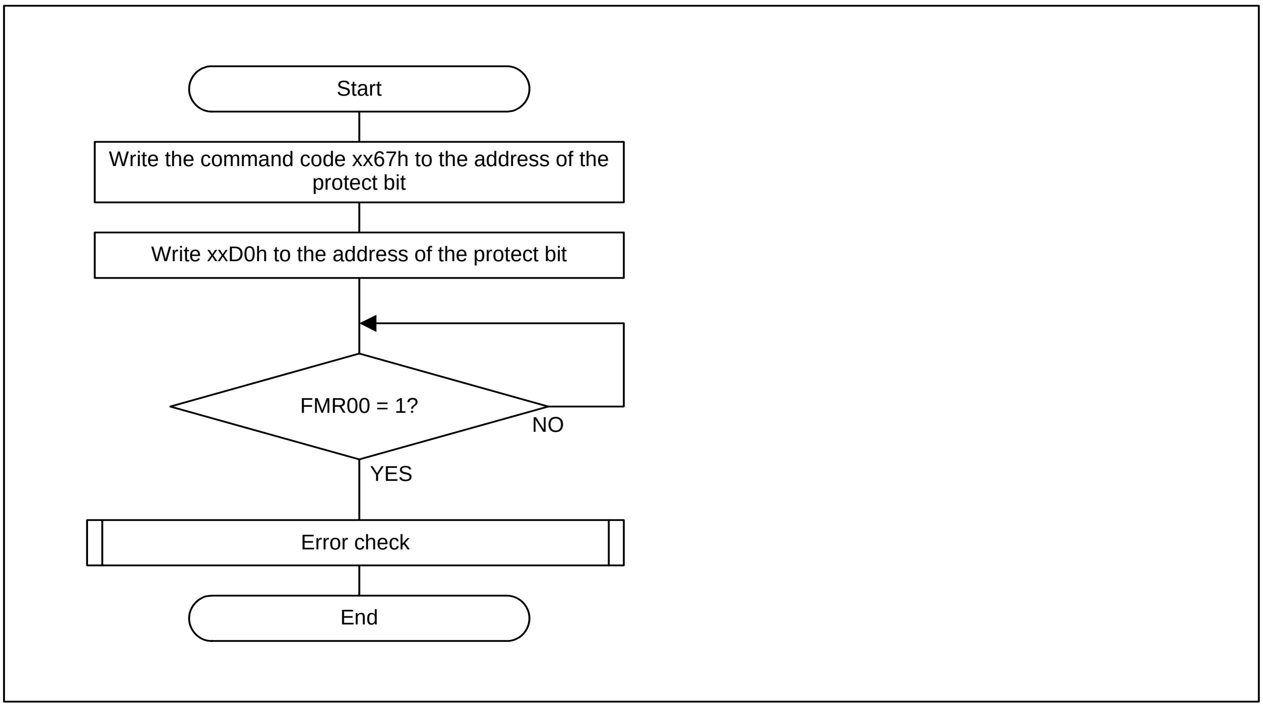

Start
Write the command code xx67h to the address of the protect bit
Write xxD0h to the address of the protect bit
FMR00 = 1?
NO
YES
Error check
End

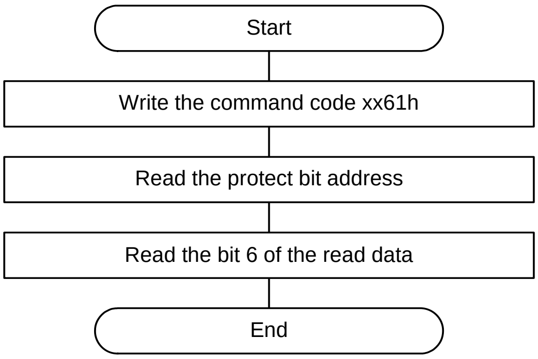
Figure 23.13 Protect Bit Program Command

(9) Read Protect Bit Status Command

The read protect bit status command reads a protect bit status of a given block.

Execute the read protect bit status command by the program placed in the RAM. By writing xx61h in the first bus cycle and reading the protect bit address of the specified block in the second bus cycle, the protect bit status of the block can be read by the bit 6 of the read data. When the bit 6 is 0 (protected), the flash memory is protected by the specified protect bit; when the bit 6 is 1 (unprotected), the flash memory is not protected by the specified protect bit. Figure 23.14 shows a flow chart of read protect bit status command. The flash memory enters read protect bit status mode after this command is executed. A protect bit status can be read in succession by reading a protect bit address.

In EW1 mode, do not execute the read protect bit status command.


Start
Write the command code xx61h
Read the protect bit address
Read the bit 6 of the read data
End

Write command code to an even address.
Write by a program placed in the RAM.

Figure 23.14 Read Protect Bit Status Command

23.6.1.2 Status Register

The Status Register can be read in EW0 mode. It indicates the operating status of the flash memory and whether an erase or program operation has completed successfully or not. The Status Register value is reflected on bits FMR00, FMR06, and FMR07 in the FMR0 register. After executing the read status register command, program command, block erase command, lock bit program command, or protect bit program command, the flash memory enters read status register mode and the Status Register returns its value by reading any even address in the user ROM area. Table 23.6 shows the Status Register.

Table 23.6 Status Register

Bit in Status Register	Bit in FMR0 Register	Status Name	Description		Value after Reset
			0	1	
SR0 (b0)	–	Reserved bit	–	–	–
SR1 (b1)	–	Reserved bit	–	–	–
SR2 (b2)	–	Reserved bit	–	–	–
SR3 (b3)	–	Reserved bit	–	–	–
SR4 (b4)	FMR06(1)	Program status	Successfully completed	Error	0
SR5 (b5)	FMR07(1)	Erase status	Successfully completed	Error	0
SR6 (b6)	–	Reserved bit	–	–	–
SR7 (b7)	FMR00	Sequencer status	BUSY	READY	1

b7 to b0: These bits return the value of 8 low-order bits by reading an even address of the flash memory in 16-bit units.

NOTE:

1. Bits FMR07 (SR5) and FMR06 (SR4) become 0 by executing the clear status register command. When the FMR07 (SR5) or FMR06 (SR4) bit is 1, the program command, block erase command, lock bit program command, read lock bit status command, and protect bit program command cannot be accepted by the flash memory.

23.6.1.3 Error Check

To confirm whether the program command, block erase command, lock bit program command, or protect bit program command is executed successfully, read bits FMR07 and FMR06 in the FMR0 register after each operation is completed.

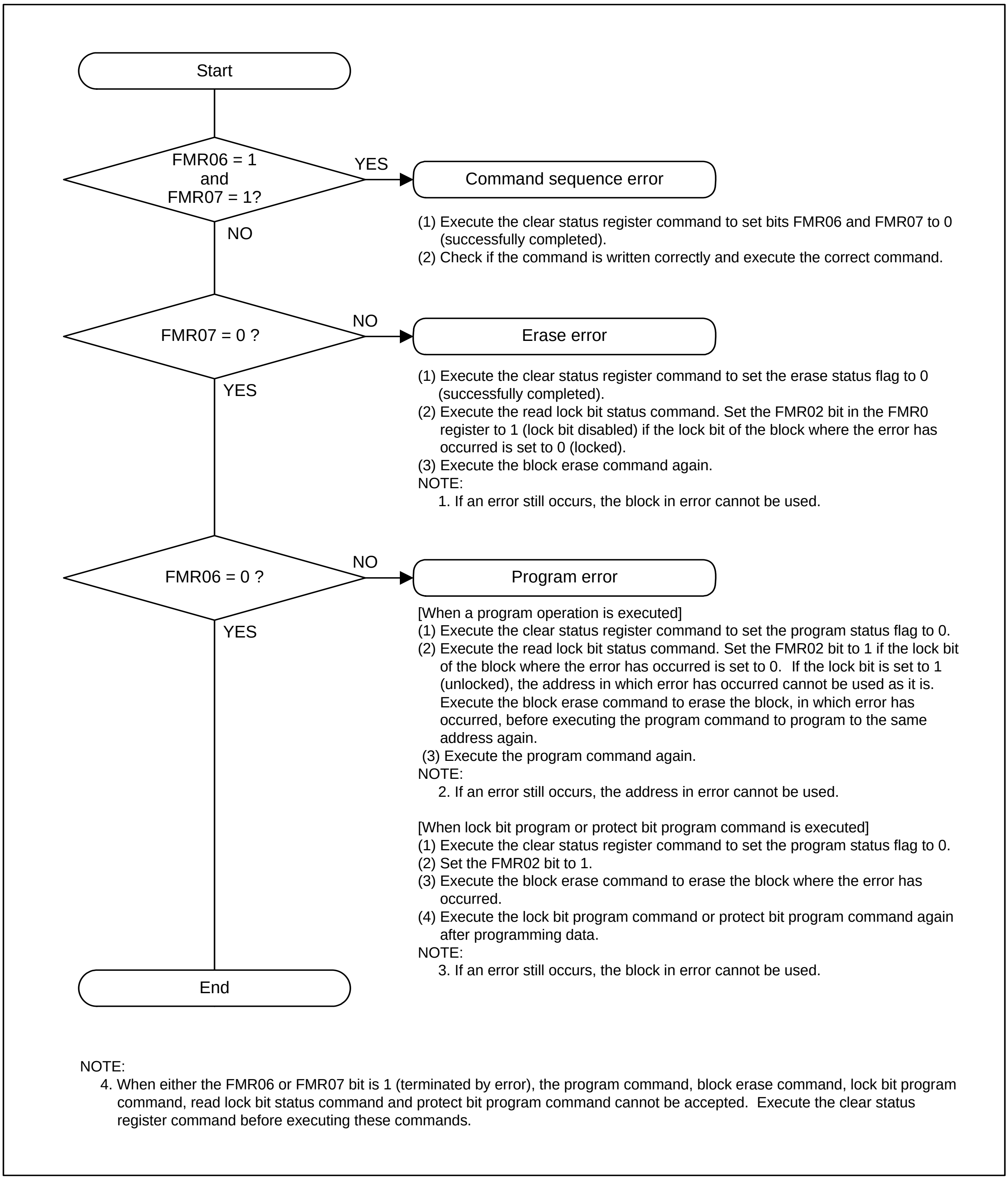
Table 23.7 lists error types and occurrence conditions. Figure 23.15 shows a flow chart of the error check and handling procedure for each error.

Table 23.7 Error Types and Occurrence Conditions

FMR0 Register (Status Register) values		Error	Error Occurrence Condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	• When a command is written incorrectly • When invalid data (data other than xxD0h or xxFFh) is written in the second bus cycle of the lock bit program command, block erase command, read lock bit status command, or protect bit program command(1)
1	0	Erase error	• When the block erase command is executed to a locked block(2) • When the block erase command is executed to an unlocked block, but the erase operation is not completed successfully
0	1	Program error	• When the program command is executed to a locked block(2) • When the program command is executed to an unlocked block, but the program operation is not completed successfully • The lock bit program command is executed, but the program operation is not completed successfully • The protect bit program command is executed, but the program operation is not completed successfully

NOTES:

1. The flash memory enters read array mode when the command code xxFFh is written in the second bus cycle of these commands. At the same time, the command code written in the first bus cycle is ignored.
2. When the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled), no error occurs under these conditions.



(1) Execute the clear status register command to set bits FMR06 and FMR07 to 0 (successfully completed).
(2) Check if the command is written correctly and execute the correct command.

(1) Execute the clear status register command to set the erase status flag to 0 (successfully completed).
(2) Execute the read lock bit status command. Set the FMR02 bit in the FMR0 register to 1 (lock bit disabled) if the lock bit of the block where the error has occurred is set to 0 (locked).
(3) Execute the block erase command again.
NOTE:
1. If an error still occurs, the block in error cannot be used.

[When a program operation is executed]
(1) Execute the clear status register command to set the program status flag to 0.
(2) Execute the read lock bit status command. Set the FMR02 bit to 1 if the lock bit of the block where the error has occurred is set to 0. If the lock bit is set to 1 (unlocked), the address in which error has occurred cannot be used as it is. Execute the block erase command to erase the block, in which error has occurred, before executing the program command to program to the same address again.
(3) Execute the program command again.
NOTE:
2. If an error still occurs, the address in error cannot be used.

[When lock bit program or protect bit program command is executed]
(1) Execute the clear status register command to set the program status flag to 0.
(2) Set the FMR02 bit to 1.
(3) Execute the block erase command to erase the block where the error has occurred.
(4) Execute the lock bit program command or protect bit program command again after programming data.
NOTE:
3. If an error still occurs, the block in error cannot be used.

NOTE:
4. When either the FMR06 or FMR07 bit is 1 (terminated by error), the program command, block erase command, lock bit program command, read lock bit status command and protect bit program command cannot be accepted. Execute the clear status register command before executing these commands.

Figure 23.15 Error Check and Handling Procedure for Each Error

23.6.2 Standard Serial I/O Mode

In standard serial I/O mode, the user ROM area can be programmed with the MCU mounted on a board by using a serial programmer supporting the M32C/8B Group. For additional information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual of your serial programmer for details on operating instructions. Table 23.8 lists pin functions for flash memory standard serial I/O mode. Figures 23.16 to 23.17 show pin connections for standard serial I/O mode.

Table 23.8 Pin Functions for Flash Memory Standard Serial I/O Mode

Pin Name	Function	Input/ Output	Supply Voltage	Description
VCC VSS	Power supply input	I	–	Apply the guaranteed erase/program supply voltage to the VCC1 pin. Apply 0 V to the VSS pin
CNVSS	CNVSS	I	VCC1	Apply an "H" signal to the pin
$\overline{\text{RESET}}$	Reset input	I	VCC1	Reset input pin
XIN	Clock input	I	VCC1	Connect a ceramic resonator or a crystal oscillator between pins XIN and XOUT
XOUT	Clock output	O	VCC1	To use the external clock, input the clock to the XIN pin and leave the XOUT pin open
BYTE	BYTE input	I	VCC1	Apply an "H" or "L" signal to the pin
AVCC, AVSS	Analog power supply input	I	–	Connect AVCC to VCC1 Connect AVSS to VSS
VREF	Reference voltage input	I	–	Reference voltage input pin for the A/D converter
P0_0 to P0_7	Input port P0	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open
P1_0 to P1_7	Input port P1	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open
P2_0 to P2_7	Input port P2	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open
P3_0 to P3_7	Input port P3	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open
P4_0 to P4_7	Input port P4	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open
P5_0	$\overline{\text{CE}}$ input	I	VCC2	Apply an "H" signal to the pin
P5_5	$\overline{\text{EPM}}$ input	I	VCC2	Apply an "L" signal to the pin
P5_1 to P5_4 P5_6, P5_7	Input port P5	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open
P6_0 to P6_3	Input port P6	I	VCC1	Apply an "H" or "L" signal to the pin, or leave it open
P6_4	BUSY output	O	VCC1	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Program operation verify monitor
P6_5	SCLK input	I	VCC1	Standard serial I/O mode 1: Serial clock input pin. This pin needs to be pulled up. Standard serial I/O mode 2: Apply an "L" signal to the pin
P6_6	Data input RXD	I	VCC1	Serial data input pin
P6_7	Data output TXD	O	VCC1	Serial data output pin. This pin needs to be pulled up when used in standard serial I/O mode1.
P7_0 to P7_7	Input port P7	I	VCC1	Apply an "H" or "L" signal to the pin, or leave it open
P8_0 to P8_4 P8_6, P8_7	Input port P8	I	VCC1	Apply an "H" or "L" signal to the pin, or leave it open
P8_5	$\overline{\text{NMI}}$ input	I	VCC1	Apply an "H" signal
P9_0 to P9_7	Input port P9	I	VCC1	Apply an "H" or "L" signal to the pin, or leave it open
P10_0 to P10_7	Input port P10	I	VCC1	Apply an "H" or "L" signal to the pin, or leave it open
P11_0 to P11_7	Input port P11	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open(1)
P12_0 to P12_7	Input port P12	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open(1)
P13_0 to P13_7	Input port P13	I	VCC2	Apply an "H" or "L" signal to the pin, or leave it open(1)
P14_0 to P14_7	Input port P14	I	VCC1	Apply an "H" or "L" signal to the pin, or leave it open(1)
P15_0 to P15_7	Input port P15	I	VCC1	Apply an "H" or "L" signal to the pin, or leave it open(1)

NOTE:

1. These pins are provided in the 144-pin package only.

Mode setting

Signal	Value
CNVSS	VCC1
$\overline{\text{EPM}}$	VSS
$\overline{\text{RESET}}$	VSS→VCC1
$\overline{\text{CE}}$	VCC2

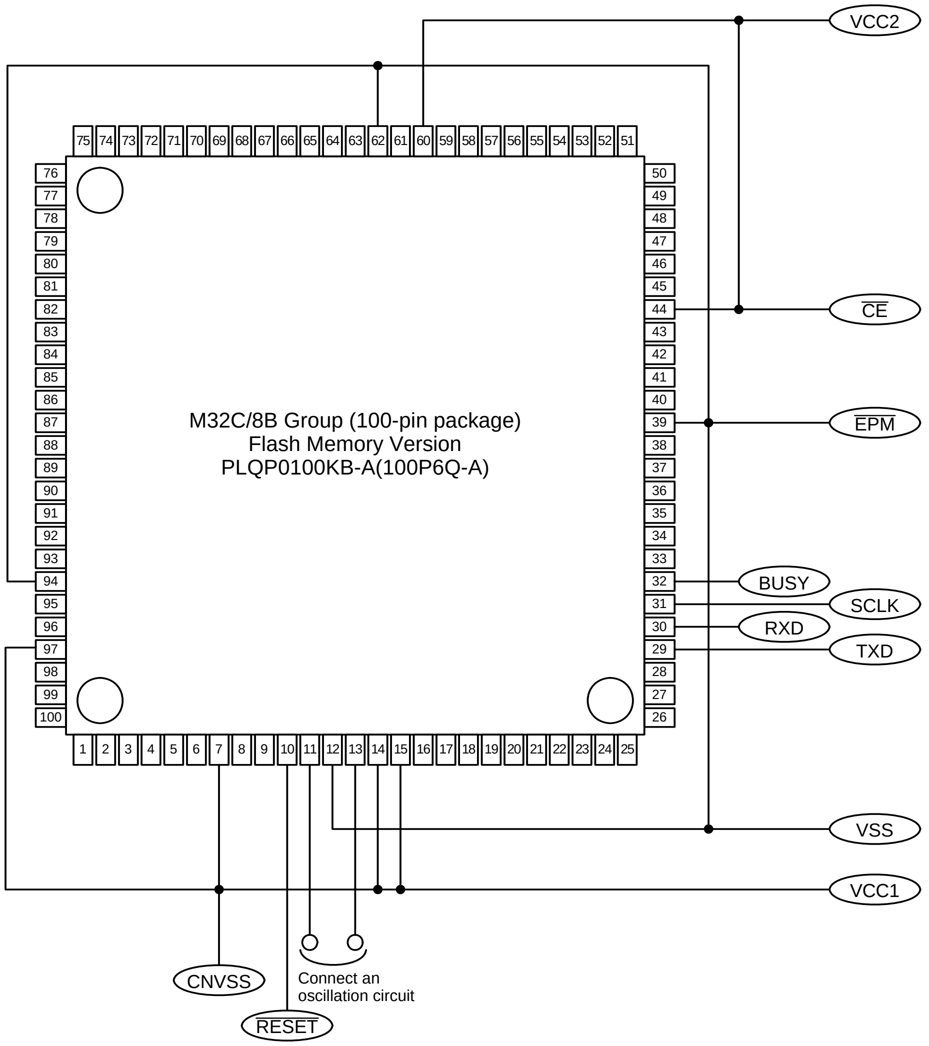


Figure 23.16 Pin Connections in Standard Serial I/O Mode (1)

Mode setting

Signal	Value
CNVSS	VCC1
$\overline{\text{EPM}}$	VSS
$\overline{\text{RESET}}$	VSS→VCC1
$\overline{\text{CE}}$	VCC2

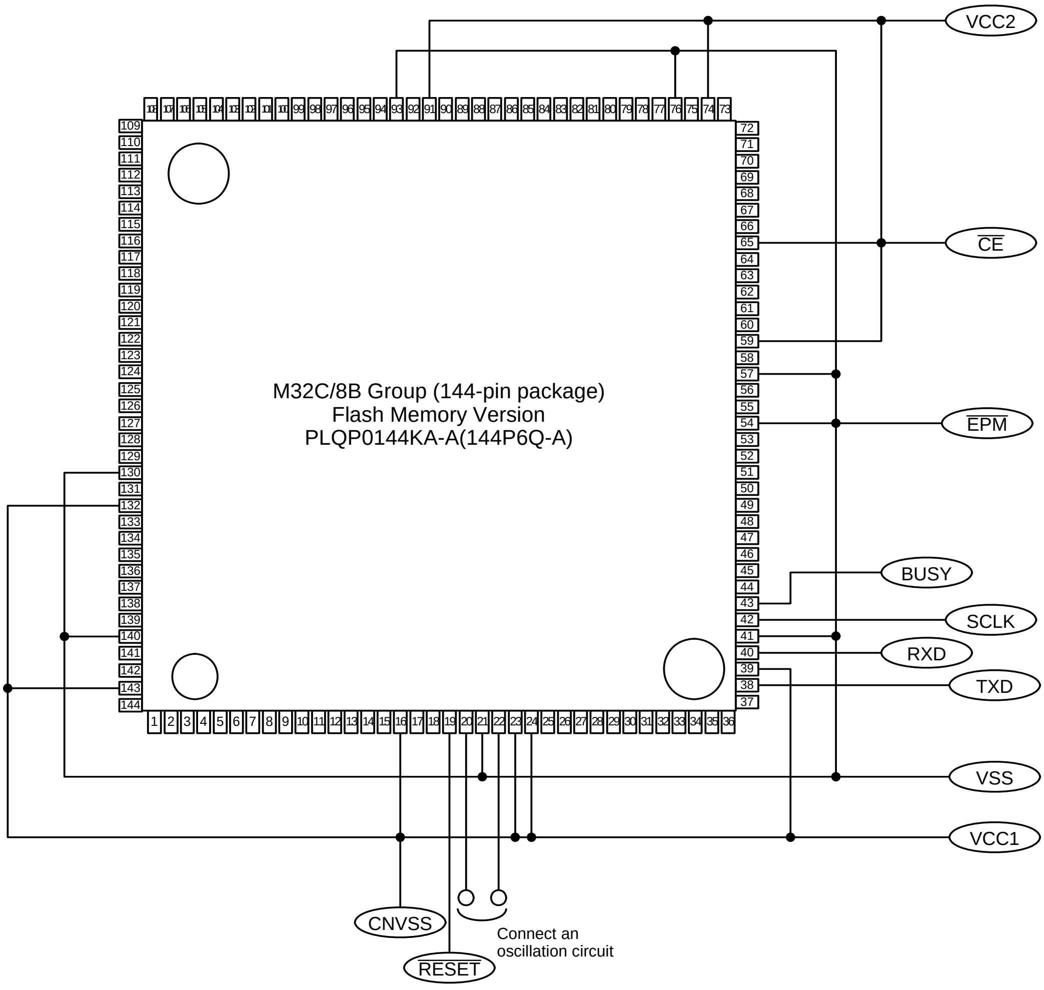
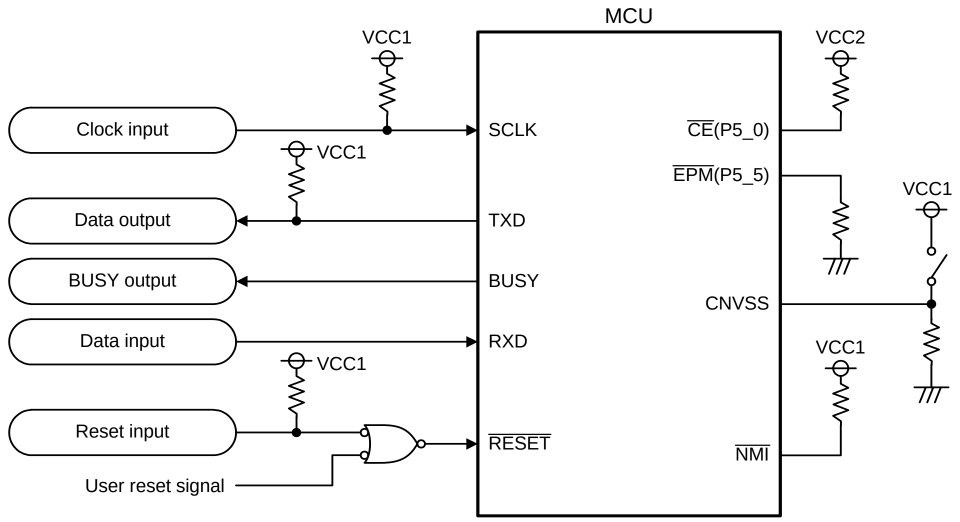


Figure 23.17 Pin Connections in Standard Serial I/O Mode (2)

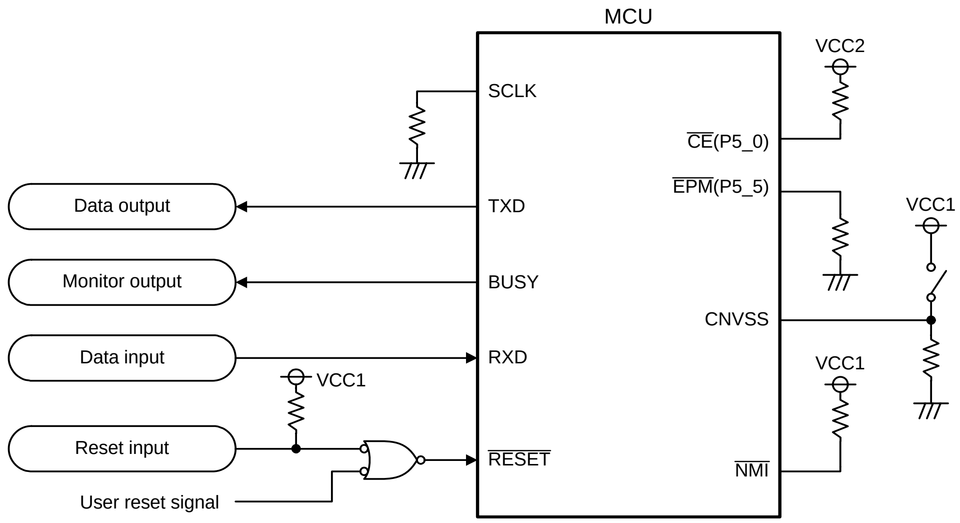
23.6.2.1 Pin Handling in Standard Serial I/O Mode

Figure 23.18 shows an example of a pin handling in standard serial I/O mode 1. Figure 23.19 shows an example of a pin handling in standard serial I/O mode 2. Refer to the user’s manual of your serial programmer to handle pins controlled by the serial programmer since controlled pins vary depending on the serial programmer.



NOTES:
1. Control pins and external circuit vary depending on the programmer. Refer to the user's manual of the programmer for information.
2. In this example, a selector controls the input voltage applied to CNVSS to switch between single-chip mode and standard serial I/O mode.
3. If there is a possibility the user reset signal becomes "L" in standard serial I/O mode 1, break the connection between the user reset signal and the RESET pin by using such as a jumper selector.

Figure 23.18 Pin Handling in Standard Serial I/O Mode 1



NOTE:
1. In this example, a selector controls the input voltage applied to CNVSS to switch between single-chip mode and standard serial I/O mode.

Figure 23.19 Pin Handling in Standard Serial I/O Mode 2

23.6.3 Parallel I/O Mode

In parallel I/O mode, the user ROM area can be programmed by using a parallel programmer supporting the M32C/8B Group. The boot ROM area can also be programmed. However, do not rewrite the boot ROM area since the rewrite control program for standard serial I/O mode is stored in the boot ROM area in the factory default configuration.
For additional information about the parallel programmer, contact your parallel programmer manufacturer. Refer to the user's manual of your parallel programmer for details on operating instructions.

24. Electrical Characteristics

Table 24.1 Absolute Maximum Ratings

Symbol	Parameter		Condition	Value	Unit
VCC1, VCC2	Supply voltage		VCC1 = AVCC	-0.3 to 6.0	V
VCC2	Supply voltage		–	-0.3 to VCC1 + 0.1	V
AVCC	Analog supply voltage		VCC1 = AVCC	-0.3 to 6.0	V
VI	Input voltage	$\overline{\text{RESET}}$, CNVSS, BYTE, P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P14_0 to P14_6, P15_0 to P15_7[1], VREF, XIN		-0.3 to VCC1 + 0.3	V
		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, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7[1]		-0.3 to VCC2 + 0.3	
		P7_0, P7_1		-0.3 to 6.0	
VO	Output voltage	P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P14_0 to 14_6, P15_0 to P15_7[1], XOUT		-0.3 to VCC1 + 0.3	V
		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, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7[1]		-0.3 to VCC2 + 0.3	
		P7_0, P7_1		-0.3 to 6.0	
Pd	Power consumption		-40°C≤Topr≤85°C	500	mW
Topr	Operating ambient temperature	during CPU operation		-20 to 85/ -40 to 85[2]	°C
		during programming or erasing Flash memory		0 to 60	°C
Tstg	Storage temperature			-65 to 150	°C

NOTES:
1. P11 to P15 are provided in the 144-pin package only.
2. Contact a Renesas sales office if temperature range of -40 to 85°C is required.

Table 24.2 Recommended Operating Conditions (1/3) (VCC1 = VCC2 = 3.0 to 5.5 V, Topr = -20 to 85°C unless otherwise specified)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
VCC1, VCC2	Supply voltage (VCC1 ≥ VCC2)		3.0	5.0	5.5	V
AVCC	Analog supply voltage			VCC1		V
VSS	Supply voltage			0		V
AVSS	Analog supply voltage			0		V
VIH	Input high “H” voltage	P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7(2)	0.8VCC2		VCC2	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_7(1), P9_0 to P9_7, P10_0 to P10_7, P14_0 to P14_6, P15_0 to P15_7(2), XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	0.8VCC1		VCC1	
		P7_0, P7_1	0.8VCC1		6.0	
		P0_0 to P0_7, P1_0 to P1_7 (in single-chip mode)	0.8VCC2		VCC2	
		P0_0 to P0_7, P1_0 to P1_7 (in memory expansion mode and microprocessor mode)	0.5VCC2		VCC2	
VIL	Input low “L” voltage	P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7(2)	0		0.2VCC2	V
		P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7(1), P9_0 to P9_7, P10_0 to P10_7, P14_0 to P14_6, P15_0 to P15_7(2), XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	0		0.2VCC1	
		P0_0 to P0_7, P1_0 to P1_7 (in single-chip mode)	0		0.2VCC2	
		P0_0 to P0_7, P1_0 to P1_7 (in memory expansion mode and microprocessor mode)	0		0.16VCC2	

NOTES:

1. VIH and VIL reference for P8_7 apply when P8_7 is used as a programmable input port. It does not apply when P8_7 is used as XCIN.
2. P11 to P15 are provided in the 144-pin package only.

Table 24.3 Recommended Operating Conditions (2/3)
(VCC1 = VCC2 = 3.0 to 5.5 V, Topr = -20 to 85°C unless otherwise specified)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
IOH(peak)	Peak output high "H" current(2)	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_2 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(3)			-10.0	mA
IOH(avg)	Average output high "H" current(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_2 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(3)			-5.0	mA
IOL(peak)	Peak output low "L" current(2)	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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(3)			10.0	mA
IOL(avg)	Average output low "L" current(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, P9_0 to P9_7, P10_0 to P10_7, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(3)			5.0	mA

NOTES:

1. Average output current is the average value within 100 ms.
2. A total IOL(peak) of P0, P1, P2, P8_6, P8_7, P9, P10, P11, P14, and P15 must be 80 mA or less.
   A total IOL(peak) of P3, P4, P5, P6, P7,P8_0 to P8_4, P12, and P13 must be 80 mA or less.
   A total IOH(peak) of P0, P1, P2, and P11 must be -40 mA or less.
   A total IOH(peak) of P8_6 to P8_7, P9, P10, P14, and P15 must be -40 mA or less.
   A total IOH(peak) of P3, P4, P5, P12, and P13 must be -40 mA or less.
   A total IOH(peak) of P6, P7, and P8_0 to P8_4 must be -40 mA or less.
3. P11 to P15 are provided in the 144-pin package only.

Table 24.4 Recommended Operating Conditions (3/3)
(VCC1 = VCC2 = 3.0 to 5.5 V, Topr = -20 to 85°C unless otherwise specified)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
f(CPU)	CPU clock frequency (same frequency as f(BCLK))	VCC1 = 3.0 to 5.5V	0		32	MHz
f(XIN)	Main clock input frequency	VCC1 = 3.0 to 5.5V	0		16	MHz
f(XCIN)	Sub clock frequency			32.768	50	kHz
f(Ring)	On-chip oscillator frequency		0.5	1	2	MHz
f(PLL)	PLL clock frequency	VCC1 = 3.0 to 5.5V	10		32	MHz
tsu(PLL)	Wait time to stabilize PLL frequency synthesizer	VCC1 = 5.0V			20	ms
		VCC1 = 3.3V			50	ms

Table 24.5 Flash Memory Electrical Characteristics (VCC1 = VCC2 = 3.0 V to 5.5 V, Topr = 0 to 60°C unless otherwise specified)

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
–	CPU clock frequency (in CPU rewrite mode)(2)				10	MHz
–	Erase and program endurance(1)		100			times
–	Program time (4 bytes) (Topr = 25°C)	Other than Data flash		150	900	µs
		Data flash		300	1700	
–	Lock bit program time	Other than Data flash		70	500	µs
		Data flash		140	1000	
–	Block erase time (Topr = 25°C)	4-Kbyte block		0.2	3	s
		8-Kbyte block		0.2	3	s
		64-Kbyte block		0.2	3	s
tps	Wait time to stabilize flash memory circuit				50	µs
–	Data hold time (Topr = -40 to 85°C)		10			years

NOTES:

1. If erase and program endurance is n times (n = 100), each block can be erased n times. For example, if a 4-Kbyte block A is erased after programming four-byte data 1,024 times, each to a different address, this counts as one erase and program time. Data cannot be programmed to the same address more than once without erasing the block (rewrite prohibited).
2. Prior to accessing registers FMR0 to FMR3 or to entering CPU rewrite mode (EW0, EW1 mode), set the CPU clock frequency to 10 MHz or lower using bits MCD4 to MCD0 in the MCD register, and also set the PM12 bit in the PM1 register to 1 (1 wait state).

VCC1 = VCC2 = 5V

Table 24.6 Electrical Characteristics (1/3)
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C, f(CPU) = 32 MHz unless otherwise specified)

Symbol	Parameter			Condition	Standard			Unit
					Min.	Typ.	Max.	
VOH	Output high “H” 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, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7(1)		IOH = -5 mA	VCC2 - 2.0		VCC2	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7,P10_0 to P10_7, P14_0 to P14_6, P15_0 to P15_7(1)		IOH = -5 mA	VCC1 - 2.0		VCC1	
		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, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7(1)		IOH = -200 μA	VCC2 - 0.3		VCC2	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7,P10_0 to P10_7, P14_0 to P14_6, P15_0 to P15_7(1)		IOH = -200 μA	VCC1 - 0.3		VCC1	
		XOUT		IOH = -1 mA	3.0		VCC1	V
		XCOUT	Drive capability = high	No load applied		2.5		V
			Drive capability = low	No load applied		1.7		V
VOL	Output low “L” 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(1)		IOL = 5 mA			2.0	V
		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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(1)		IOL = 200 μA			0.45	V
		XOUT		IOL = 1 mA			2.0	V
		XCOUT	Drive capability = high	No load applied		0		V
			Drive capability = low	No load applied		0		V
VT+ - VT-	Hysteresis	$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, TA0IN to TA4IN, TB0IN to TB5IN, $\overline{\text{INT0}}$ to $\overline{\text{INT5}}$, $\overline{\text{ADTRG}}$, $\overline{\text{CTS0}}$ to $\overline{\text{CTS4}}$, CLK0 to CLK4, TA0OUT to TA4OUT, $\overline{\text{NMI}}$, $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$, RXD0 to RXD4, SCL0 to SCL4, SDA0 to SDA4			0.2		1.0	V
		$\overline{\text{RESET}}$			0.2		1.8	V

NOTE:
1. P11 to P15 are provided in the 144-pin package only.

VCC1 = VCC2 = 5V

Table 24.7 Electrical Characteristics (2/3)
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C, f(CPU) = 32 MHz unless otherwise specified)

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
IIH	Input high "H" 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7[1], XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	VI = 5 V			5.0	μA
IIL	Input low "L" 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7[1], XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	VI = 0V			-5.0	μA
RPULLUP	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_2 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7[1]	VI = 0V	30	50	170	kΩ
RfXIN	Feedback resistance	XIN			1.5		MΩ
RfXCIN	Feedback resistance	XCIN			15		MΩ
VRAM	RAM data retention voltage	In stop mode		2.0			V

NOTE:

1. P11 to P15 are provided in the 144-pin package only.

VCC1 = VCC2 = 5V

Table 24.8 Electrical Characteristics (3/3) (VCC1 = VCC2 = 5.5 V, VSS = 0 V, Topr = 25°C)

Symbol	Parameter	Condition[1]	Standard			Unit
			Min.	Typ.	Max.	
ICC	Power supply current	f(CPU) = 32 MHz		26	42	mA
		f(CPU) = 16 MHz		16		mA
		f(CPU) = 8 MHz		10		mA
		f(CPU) = f(Ring)[3] In on-chip oscillator low-power consumption mode		1.5		mA
		In on-chip oscillator low-power consumption mode, flash memory is stopped[2]		400		µA
		f(CPU) = 32 kHz[4] In low-power consumption mode, flash memory is operating		430		µA
		f(CPU) = 32 kHz[5] In low-power consumption mode, flash memory is stopped[2]		50		µA
		Wait mode: f(CPU) = f(Ring) After entering wait mode from on-chip oscillator low-power consumption mode		110		µA
		Wait mode: f(CPU) = 32kHz[6] After entering wait mode from low-power consumption mode		10		µA
		Stop mode (clock is stopped)		4		µA
		Stop mode (clock is stopped) Topr = 85°C			200	µA

NOTES:

1. In single-chip mode, leave the output pins open and connect the input pins to VSS.
2. When setting the FMSTP bit in the FMR0 register to 1 (flash memory stopped) and running the program on RAM.
3. When the FMR40 bit in the FMR4 register is set to 1 (low-speed access).
4. When the FMR40 bit is set to 1 and the MRS bit in the VRCR register is set to 1 (main voltage regulator stops).
5. When the MRS bit is set to 1.
6. When the MRS bit is set to 1 and the CM0 bit in the CM03 register is set to 0 (XCIN-XCOUT drive capability Low).

VCC1 = VCC2 = 5V

Table 24.9 A/D Conversion Characteristics (VCC1 = VCC2 = AVCC = VREF = 4.2 to 5.5 V, VSS = AVSS = 0 V, Topr = -20 to 85°C, f(CPU) = 32MHz unless otherwise specified)

Symbol	Parameter	Measurement Condition		Standard			Unit
				Min.	Typ.	Max.	
–	Resolution	VREF = VCC1				10	Bits
INL	Integral nonlinearity error	VREF = VCC1 = VCC2 = 5 V	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_7, ANEX0, ANEX1			±3	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential nonlinearity error					±1	LSB
–	Offset error					±3	LSB
–	Gain error					±3	LSB
RLADDER	Resistor ladder	VREF = VCC1		4		20	kΩ
tCONV	10-bit conversion time(1)(2)			2.06			μs
tCONV	8-bit conversion time(1)(2)			1.75			μs
tSAMP	Sampling time(1)			0.188			μs
VREF	Reference voltage			3		VCC1	V
VIA	Analog input voltage			0		VREF	V

NOTES:

1. The value is obtained when φAD frequency is at 16 MHz. Keep φAD frequency at 16 MHz or lower.
2. With using the sample and hold function

Table 24.10 D/A Conversion Characteristics (VCC1 = VCC2 = VREF = 4.2 to 5.5 V, VSS = AVSS = 0 V, Topr = -20 to 85°C, f(CPU) = 32MHz unless otherwise specified)

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
tsu	Setup time				3	μs
RO	Output resistance		4	10	20	kΩ
IVREF	Reference power supply input current	(note 1)			1.5	mA

NOTE:

1. Measured when one D/A converter is used, and the DAi register (i = 0, 1) of the unused D/A converter is set to 00h. The current flown into the resistor ladder in the A/D converter is excluded. IVREF flows even if the VCUT bit in the AD0CON1 register is set to 0 (VREF not connected).

VCC1 = VCC2 = 5V

Table 24.11 Voltage Detection Circuit Electrical Characteristics (VCC1 = VCC2 = 3.0 to 5.5 V, VSS = 0 V, Topr = 25°C unless otherwise specified)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
ΔVdet	Detection voltage level accuracy	VCC1 = 3.0 V to 5.5 V			±0.30	V

Table 24.12 Power Supply Timing Characteristics

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait time to stabilize internal supply voltage when power-on	VCC1 = 3.0 to 5.5 V			2	ms
td(E-A)	Start-up time for Vdet detection circuit	VCC1 = 3.0 to 5.5 V			150	µs

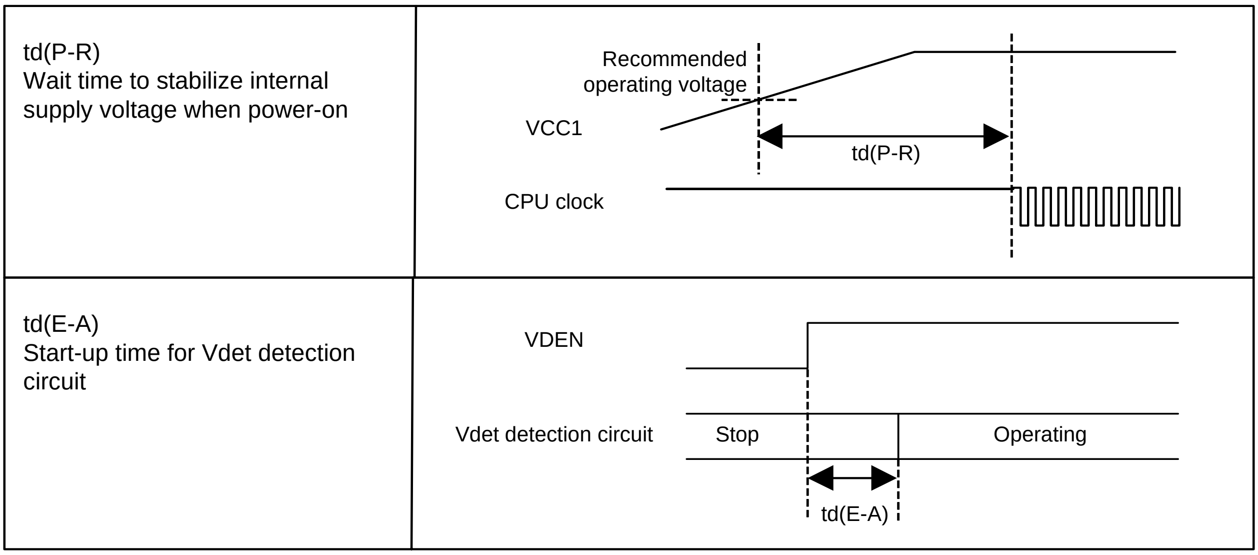

td(P-R)
Wait time to stabilize internal supply voltage when power-on
Recommended operating voltage
VCC1
td(P-R)
CPU clock
td(E-A)
Start-up time for Vdet detection circuit
VDEN
Vdet detection circuit
Stop
Operating
td(E-A)

Figure 24.1 Power Supply Timing Diagram

VCC1 = VCC2 = 5V

Timing Requirements
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.13 External Clock Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External clock input cycle time	62.5		ns
tw(H)	External clock input high (“H”) pulse width	27.5		ns
tw(L)	External clock input low (“L”) pulse width	27.5		ns
tr	External clock rise time		5	ns
tf	External clock fall time		5	ns

Table 24.14 Timer A Input (Count Source Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	100		ns
tw(TAH)	TAiIN input high (“H”) pulse width	40		ns
tw(TAL)	TAiIN input low (“L”) pulse width	40		ns

i = 0 to 4

Table 24.15 Timer A Input (Gate Signal Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	400		ns
tw(TAH)	TAiIN input high (“H”) pulse width	200		ns
tw(TAL)	TAiIN input low (“L”) pulse width	200		ns

i = 0 to 4

Table 24.16 Timer A Input (External Trigger Input in One-Shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	200		ns
tw(TAH)	TAiIN input high (“H”) pulse width	100		ns
tw(TAL)	TAiIN input low (“L”) pulse width	100		ns

i = 0 to 4

Table 24.17 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(TAH)	TAiIN input high (“H”) pulse width	100		ns
tw(TAL)	TAiIN input low (“L”) pulse width	100		ns

i = 0 to 4

VCC1 = VCC2 = 5V

Timing Requirements
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.18 Timer A Input (Counter Increment/Decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(UP)	TAiOUT input cycle time	2000		ns
tw(UPH)	TAiOUT input high (“H”) pulse width	1000		ns
tw(UPL)	TAiOUT input low (“L”) pulse width	1000		ns
tsu(UP-TIN)	TAiOUT input setup time	400		ns
th(TIN-UP)	TAiOUT input hold time	400		ns

i = 0 to 4

Table 24.19 Timer A Input (Two-Phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	800		ns
tsu(TAIN-TAOUT)	TAiOUT input setup time	200		ns
tsu(TAOUT-TAIN)	TAiIN input setup time	200		ns

i = 0 to 4

Table 24.20 Timer B Input (Count Source Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time (counted on one edge)	100		ns
tw(TBH)	TBiIN input high (“H”) pulse width (counted on one edge)	40		ns
tw(TBL)	TBiIN input low (“L”) pulse width (counted on one edge)	40		ns
tc(TB)	TBiIN input cycle time (counted on both edges)	200		ns
tw(TBH)	TBiIN input high (“H”) pulse width (counted on both edges)	80		ns
tw(TBL)	TBiIN input low (“L”) pulse width (counted on both edges)	80		ns

i = 0 to 5

Table 24.21 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time	400		ns
tw(TBH)	TBiIN input high (“H”) pulse width	200		ns
tw(TBL)	TBiIN input low (“L”) pulse width	200		ns

i = 0 to 5

Table 24.22 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time	400		ns
tw(TBH)	TBiIN input high (“H”) pulse width	200		ns
tw(TBL)	TBiIN input low (“L”) pulse width	200		ns

i = 0 to 5

VCC1 = VCC2 = 5V

Timing Requirements
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.23 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(AD)	$\overline{\text{ADTRG}}$ input cycle time (required for trigger)	1000		ns
tw(ADL)	$\overline{\text{ADTRG}}$ input low (“L”) pulse width	125		ns
tw(ADH)	$\overline{\text{ADTRG}}$ input high (“H”) pulse width	3		ϕAD

Table 24.24 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	CLKi input cycle time	200		ns
tw(CKH)	CLKi input high (“H”) pulse width	100		ns
tw(CKL)	CLKi input low (“L”) pulse width	100		ns
td(C-Q)	TXDi output delay time		80	ns
th(C-Q)	TXDi output hold time	0		ns
tsu(D-C)	RXDi input setup time	80		ns
th(C-D)	RXDi input hold time	90		ns

i=0 to 4

Table 24.25 External Interrupt $\overline{\text{INTi}}$ Input (Edge Sensitive)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(INH)	$\overline{\text{INTi}}$ input high (“H”) pulse width	250		ns
tw(INL)	$\overline{\text{INTi}}$ input low (“L”) pulse width	250		ns

i=0 to 5

VCC1 = VCC2 = 5V

Timing Requirements
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.26 Memory Expansion mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tac1(RD-DB)	Data input access time (RD standard)		(note 1)	ns
tac1(AD-DB)	Data input access time (AD standard, CS standard)		(note 1)	ns
tac2(RD-DB)	Data input access time (RD standard, when accessing a space with the multiplexed bus)		(note 1)	ns
tac2(AD-DB)	Data input access time (AD standard, when accessing a space with the multiplexed bus)		(note 1)	ns
tsu(DB-BCLK)	Data input setup time	26		ns
tsu(RDY-BCLK)	$\overline{\text{RDY}}$ input setup time	26		ns
tsu(HOLD-BCLK)	$\overline{\text{HOLD}}$ input setup time	30		ns
th(RD-DB)	Data input hold time	0		ns
th(BCLK-RDY)	$\overline{\text{RDY}}$ input hold time	0		ns
th(BCLK-HOLD)	$\overline{\text{HOLD}}$ input hold time	0		ns
td(BCLK-HLDA)	$\overline{\text{HLDA}}$ output delay time		25	ns

NOTE:

1. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equations. Insert wait states or lower the operation frequency, f(BCLK), if the calculated value is negative.

$$\mathrm{tac1(RD\text{-}DB)} = \frac{10^9 \times m}{f(\mathrm{BCLK}) \times 2} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) + 1)$$

$$\mathrm{tac1(AD\text{-}DB)} = \frac{10^9 \times n}{f(\mathrm{BCLK})} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ n = a + b)$$

$$\mathrm{tac2(RD\text{-}DB)} = \frac{10^9 \times m}{f(\mathrm{BCLK}) \times 2} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) - 1)$$

$$\mathrm{tac2(AD\text{-}DB)} = \frac{10^9 \times p}{f(\mathrm{BCLK}) \times 2} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ p = \{(a + b - 1) \times 2\} + 1)$$

VCC1 = VCC2 = 5V

Switching Characteristics
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.27 Memory Expansion Mode and Microprocessor Mode (when accessing external memory space)

Symbol	Parameter	Measurement Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	See Figure 24.2		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		-3		ns
th(RD-AD)	Address output hold time (RD standard)(3)		0		ns
th(WR-AD)	Address output hold time (WR standard)(3)		(note 1)		ns
td(BCLK-CS)	Chip-select signal output delay time			18	ns
th(BCLK-CS)	Chip-select signal output hold time (BCLK standard)		-3		ns
th(RD-CS)	Chip-select signal output hold time (RD standard)(3)		0		ns
th(WR-CS)	Chip-select signal output hold time (WR standard)(3)		(note 1)		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		-5		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		-5		ns
td(DB-WR)	Data output delay time (WR standard)		(note 2)		ns
th(WR-DB)	Data output hold time (WR standard)(3)		(note 1)		ns
tw(WR)	WR output width		(note 2)		ns

NOTES:

1. Values, which depend on BCLK frequency, can be obtained from the following equations.

$$th(WR\text{-}DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

2. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equations.

$$td(DB\text{-}WR) = \frac{10^9 \times m}{f(BCLK)} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } m = b\text{)}$$

$$tw(WR) = \frac{10^9 \times n}{f(BCLK) \times 2} - 15 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } n = (b \times 2) - 1\text{)}$$

3. tc [ns] is added when recovery cycle is inserted.

VCC1 = VCC2 = 5V

Switching Characteristics
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.28 Memory Expansion Mode and Microprocessor Mode
(when accessing external memory space with multiplexed bus)

Symbol	Parameter	Measurement Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	See Figure 24.2		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		-3		ns
th(RD-AD)	Address output hold time (RD standard)(5)		(note 1)		ns
th(WR-AD)	Address output hold time (WR standard)(5)		(note 1)		ns
td(BCLK-CS)	Chip-select signal output delay time			18	ns
th(BCLK-CS)	Chip-select signal output hold time (BCLK standard)		-3		ns
th(RD-CS)	Chip-select signal output hold time (RD standard)(5)		(note 1)		ns
th(WR-CS)	Chip-select signal output hold time (WR standard)(5)		(note 1)		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		-5		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		-5		ns
td(DB-WR)	Data output delay time (WR standard)		(note 2)		ns
th(WR-DB)	Data output hold time (WR standard)(5)		(note 1)		ns
td(BCLK-ALE)	ALE signal output delay time (BCLK standard)			18	ns
th(BCLK-ALE)	ALE signal output hold time (BCLK standard)		-2		ns
td(AD-ALE)	ALE signal output delay time (address standard)		(note 3)		ns
th(ALE-AD)	ALE signal output hold time (address standard)		(note 4)		ns
tdz(RD-AD)	Address output float start time			8	ns

NOTES:

1. Values, which depend on BCLK frequency, can be obtained from the following equations.

$$th(RD\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(RD\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

2. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$td(DB\text{-}WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } m = (b \times 2) - 1)$$

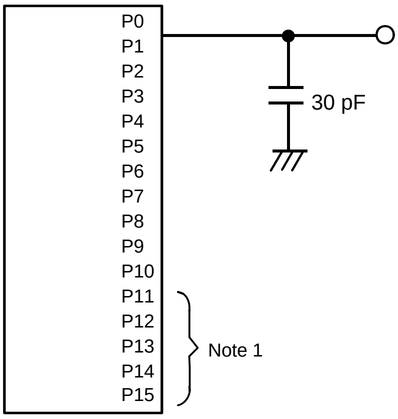
3. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$td(AD\text{-}ALE) = \frac{10^9 \times n}{f(BCLK) \times 2} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } n = a)$$

4. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$th(ALE\text{-}AD) = \frac{10^9 \times n}{f(BCLK) \times 2} - 10 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } n = a)$$

5. tc [ns] is added when recovery cycle is inserted.



NOTE:
1. P11 to P15 are provided in the 144-pin package only.

Figure 24.2 P0 to P15 Measurement Circuit

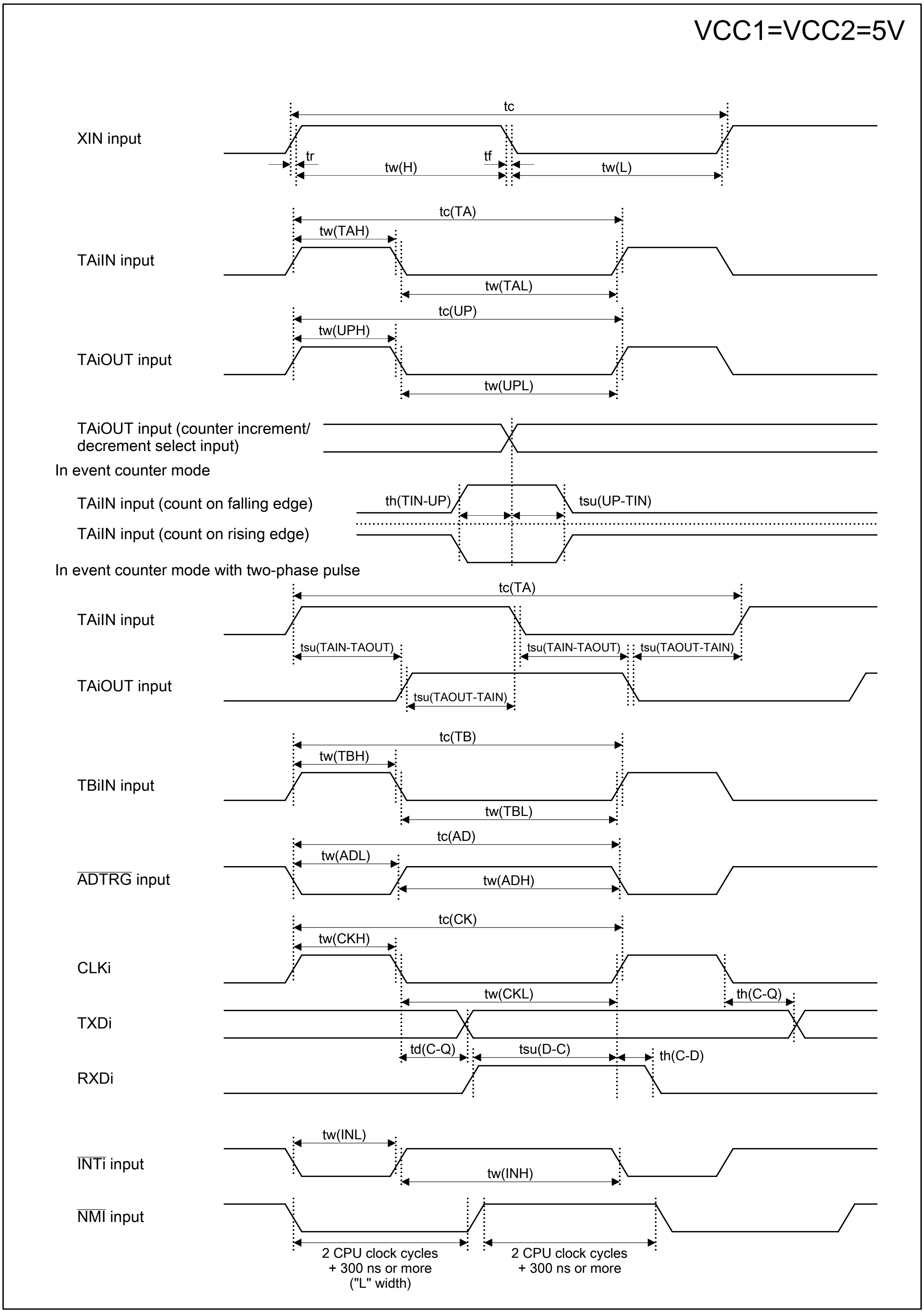
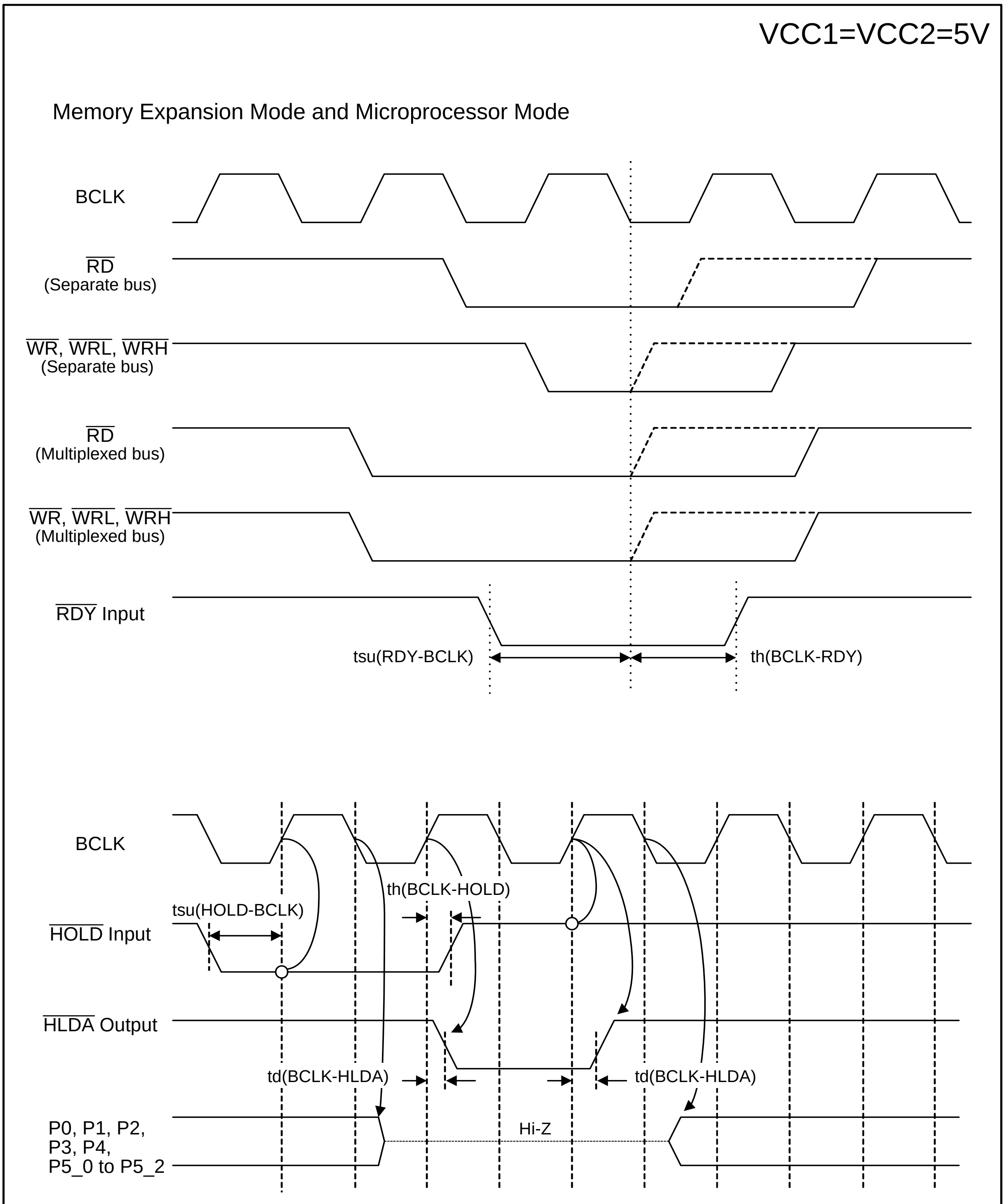
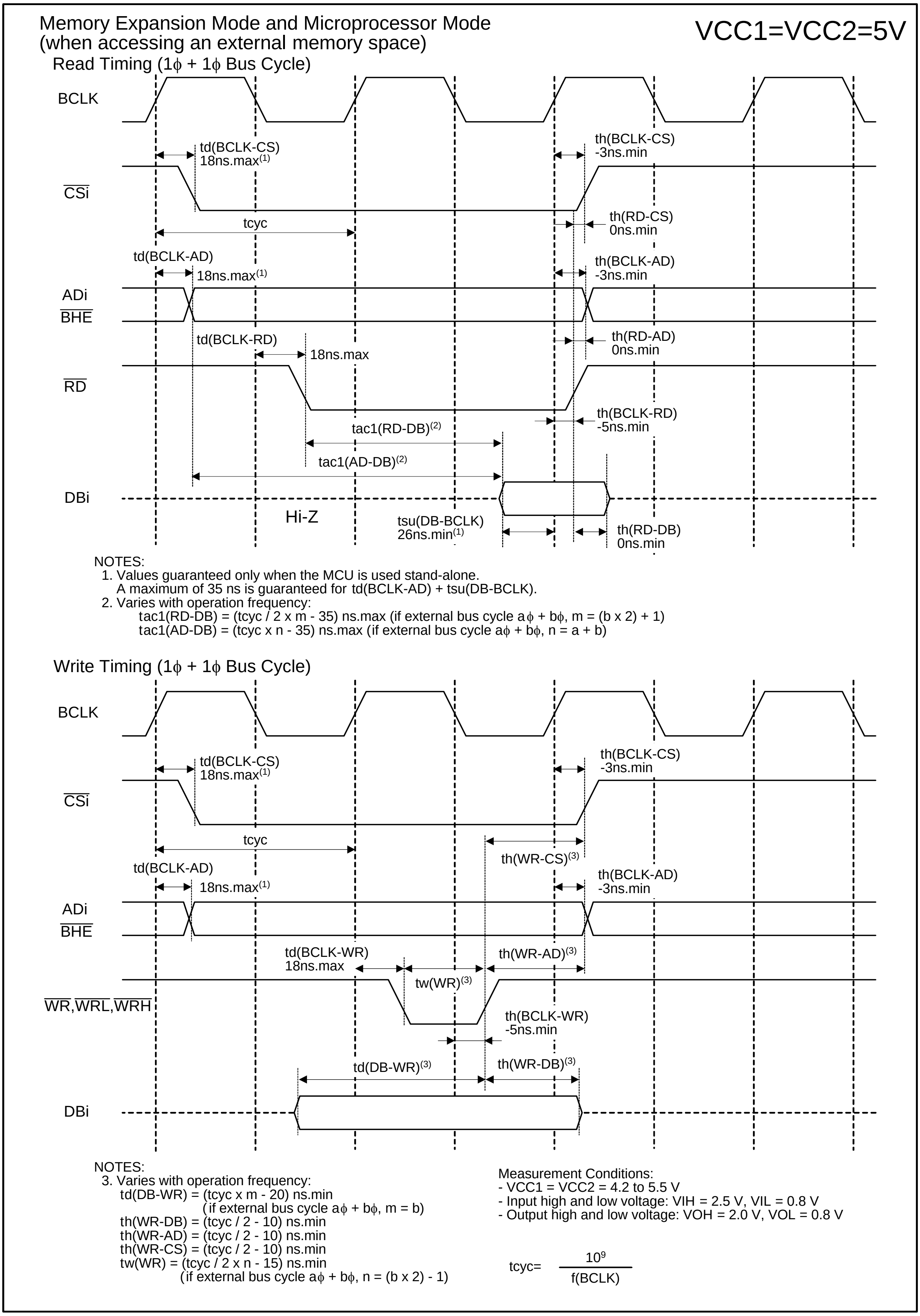


Figure 24.3 VCC1 = VCC2 = 5 V Timing Diagram (1)



Measurement Conditions
- VCC1 = VCC2 = 4.2 to 5.5 V
- Input high and low voltage: VIH = 4.0 V, VIL = 1.0 V
- Output high and low voltage: VOH = 2.5 V, VOL = 2.5 V

Figure 24.4 VCC1 = VCC2 = 5 V Timing Diagram (2)



NOTES:
1. Values guaranteed only when the MCU is used stand-alone.
A maximum of 35 ns is guaranteed for td(BCLK-AD) + tsu(DB-BCLK).
2. Varies with operation frequency:
tac1(RD-DB) = (tcyc / 2 x m - 35) ns.max (if external bus cycle aϕ + bϕ, m = (b x 2) + 1)
tac1(AD-DB) = (tcyc x n - 35) ns.max (if external bus cycle aϕ + bϕ, n = a + b)

NOTES:
3. Varies with operation frequency:
td(DB-WR) = (tcyc x m - 20) ns.min
(if external bus cycle aϕ + bϕ, m = b)
th(WR-DB) = (tcyc / 2 - 10) ns.min
th(WR-AD) = (tcyc / 2 - 10) ns.min
th(WR-CS) = (tcyc / 2 - 10) ns.min
tw(WR) = (tcyc / 2 x n - 15) ns.min
(if external bus cycle aϕ + bϕ, n = (b x 2) - 1)

Measurement Conditions:
- VCC1 = VCC2 = 4.2 to 5.5 V
- Input high and low voltage: VIH = 2.5 V, VIL = 0.8 V
- Output high and low voltage: VOH = 2.0 V, VOL = 0.8 V

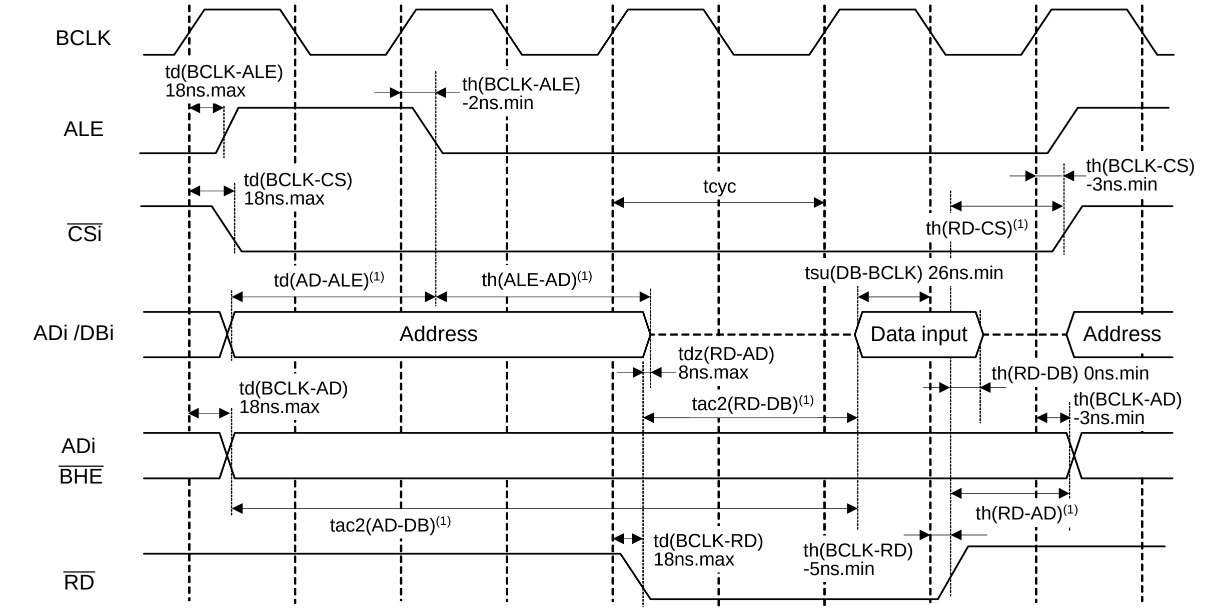
$$tcyc = \frac{10^9}{f(BCLK)}$$

Figure 24.5 VCC1 = VCC2 = 5 V Timing Diagram (3)

Memory Expansion Mode and Microprocessor Mode
(when accessing an external memory space with the multiplexed bus)

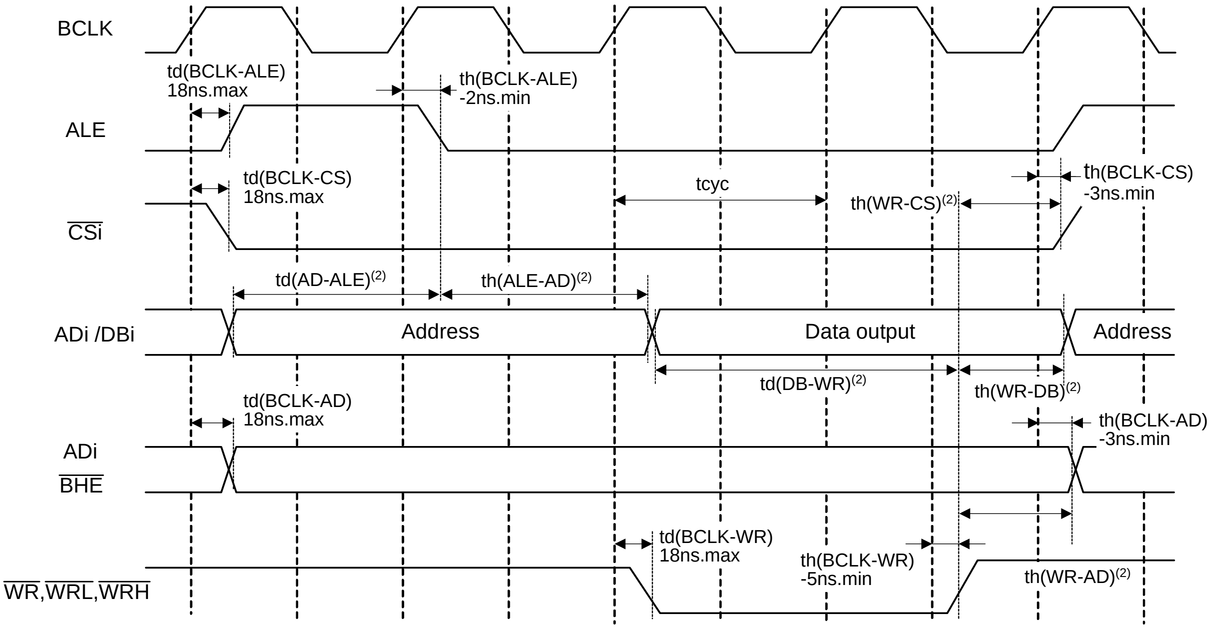
VCC1=VCC2=5V

Read Timing (2ϕ + 2ϕ Bus Cycle)



NOTES:
1. Varies with operation frequency:
td(AD-ALE) = (tcyc / 2 x n - 20) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(ALE-AD) = (tcyc / 2 x n - 10) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(RD-AD) = (tcyc / 2 - 10) ns.min, th(RD-CS) = (tcyc / 2 - 10) ns.min
tac2(RD-DB) = (tcyc / 2 x m - 35) ns.max (if external bus cycle aϕ + bϕ, m = (b x 2) - 1)
tac2(AD-DB) = (tcyc / 2 x p - 35) ns.max (if external bus cycle aϕ + bϕ, p = {(a + b - 1) x 2} + 1)

Write Timing (2ϕ + 2ϕ Bus Cycle)



NOTES:
1. Varies with operation frequency:
td(AD-ALE) = (tcyc / 2 x n - 20) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(ALE-AD) = (tcyc / 2 x n - 10) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(WR-AD) = (tcyc / 2 - 10) ns.min, th(WR-CS) = (tcyc / 2 - 10) ns.min
th(WR-DB) = (tcyc / 2 - 10) ns.min
td(DB-WR) = (tcyc / 2 x m - 25) ns.min (if external bus cycle aϕ + bϕ, m = (b x 2) - 1)

$$tcyc= \frac{10^9}{f(BCLK)}$$

Measurement Conditions:
- VCC1 = VCC2 = 4.2 to 5.5 V
- Input high and low voltage VIH = 2.5 V, VIL = 0.8 V
- Output high and low voltage VOH = 2.0 V, VOL = 0.8 V

Figure 24.6 VCC1 = VCC2 = 5 V Timing Diagram (4)

VCC1 = VCC2 = 3.3 V

Table 24.29 Electrical Characteristics (1/3)
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C, f(CPU) = 24 MHz unless otherwise specified)

Symbol	Parameter			Condition	Standard			Unit
					Min.	Typ.	Max.	
VOH	Output high "H" 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, P11_0 to P11_4, P12_0 to P12_7, P13_0 to P13_7[1]		IOH = -1 mA	VCC2 - 0.6		VCC2	V
		P6_0 to P6_7, P7_2 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7,P10_0 to P10_7, P14_0 to P14_6, P15_0 to P15_7[1]			VCC1 - 0.6		VCC1	
		XOUT		IOH = -0.1 mA	2.7		VCC1	V
		XCOUT	Drive capability = high	No load applied		2.5		V
			Drive capability = low	No load applied		1.7		V
VOL	Output low "L" 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7[1]		IOL = 1 mA			0.5	V
		XOUT		IOL = 0.1 mA			0.5	V
		XCOUT	Drive capability = high	No load applied		0		V
			Drive capability = low	No load applied		0		V
VT+ - VT-	Hysteresis	$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, TA0IN to TA4IN, TB0IN to TB5IN, $\overline{\text{INT0}}$ to $\overline{\text{INT5}}$, $\overline{\text{ADTRG}}$, $\overline{\text{CTS0}}$ to $\overline{\text{CTS4}}$, CLK0 to CLK4, TA0OUT to TA4OUT, $\overline{\text{NMI}}$, $\overline{\text{KI0}}$ to $\overline{\text{KI3}}$, RXD0 to RXD4, SCL0 to SCL4, SDA0 to SDA4			0.2		1.0	V
		$\overline{\text{RESET}}$			0.2		1.8	V

NOTE:
1. P11 to P15 are provided in the 144-pin package only.

VCC1 = VCC2 = 3.3 V

Table 24.30 Electrical Characteristics (2/3)
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C, f(CPU) = 24 MHz unless otherwise specified)

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
IIH	Input high “H” 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(1), XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	VI = 3 V			4.0	μA
IIL	Input low “L” 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(1), XIN, $\overline{\text{RESET}}$, CNVSS, BYTE	VI = 0V			-4.0	μA
RPULLUP	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_2 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_4, P12_0 to P12_7, P13_0 to P13_7, P14_0 to P14_6, P15_0 to P15_7(1)	VI=0V	50	100	500	kΩ
RfXIN	Feedback resistance	XIN			3.0		MΩ
RfXCIN	Feedback resistance	XCIN			25		MΩ
VRAM	RAM data retention voltage	In stop mode		2.0			V

NOTE:

1. P11 to P15 are provided in the 144-pin package only.

VCC1 = VCC2 = 3.3 V

Table 24.31 Electrical Characteristics (3/3) (VCC1 = VCC2 = 3.3 V, VSS = 0 V, Topr = 25°C)

Symbol	Parameter	Condition(1)	Standard			Unit
			Min.	Typ.	Max.	
ICC	Power supply current	f(CPU) = 32 MHz		23	37	mA
		f(CPU) = 16 MHz		15		mA
		f(CPU) = 8 MHz		9		mA
		f(CPU) = f(Ring)(3) In on-chip oscillator low-power consumption mode		1.5		mA
		In on-chip oscillator low-power consumption mode, flash memory is stopped(2)		400		µA
		f(CPU) = 32 kHz(4) In low-power consumption mode, flash memory is operating		430		µA
		f(CPU) = 32 kHz(5) In low-power consumption mode, flash memory is stopped(2)		50		µA
		Wait mode: f(CPU) = f(Ring) After entering wait mode from on-chip oscillator low-power consumption mode		110		µA
		Wait mode: f(CPU) = 32kHz(6) After entering wait mode from low-power consumption mode		8		µA
		Stop mode (clock is stopped)		4		µA
		Stop mode (clock is stopped) Topr = 85°C			200	µA

NOTES:

1. In single-chip mode, leave the output pins open and connect the input pins to VSS.
2. When setting the FMSTP bit in the FMR0 register to 1 (flash memory stopped) and running the program on RAM.
3. When the FMR40 bit in the FMR4 register is set to 1 (low-speed access).
4. When the FMR40 bit is set to 1 and the MRS bit in the VRCR register is set to 1 (main voltage regulator stops).
5. When the MRS bit is set to 1.
6. When the MRS bit is set to 1 and the CM0 bit in the CM03 register is set to 0 (XCIN-XCOUT drive capability Low).

VCC1 = VCC2 = 3.3 V

Table 24.32 A/D Conversion Characteristics (VCC1 = VCC2 = AVCC = VREF = 3.0 to 3.6 V, VSS = AVSS = 0 V, Topr = -20 to 85°C, f(CPU) = 24MHz unless otherwise specified)

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution	VREF = VCC1			10	Bits
INL	Integral nonlinearity error (8-bit)	VREF = VCC1 = VCC2 = 3.3 V			±2	LSB
DNL	Differential nonlinearity error (8-bit)				±1	LSB
–	Offset error (8-bit)				±2	LSB
–	Gain error (8-bit)				±2	LSB
RLADDER	Resistor ladder	VREF = VCC1	4		20	kΩ
tCONV	8-bit conversion time(1)(2)		4.9			µs
VREF	Reference voltage		3		VCC1	V
VIA	Analog input voltage		0		VREF	V

NOTES:

1. The value when ϕAD frequency is at 10 MHz. Keep ϕAD frequency at 10 MHz or lower.
   If f(CPU) (=fAD) is 24 MHz, divide f(CPU) by 3 to make it 8 MHz. The conversion time in this case is 6.1 µs.
2. S&H not available.

Table 24.33 D/A Conversion Characteristics (VCC1 = VCC2 = VREF = 3.0 to 3.6 V, VSS = AVSS = 0 V at Topr = -20 to 85°C, f(CPU) = 24MHz unless otherwise specified)

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
tsu	Setup time				3	µs
RO	Output resistance		4	10	20	kΩ
IVREF	Reference power supply input current	(note 1)			1.0	mA

NOTE:

1. Measurement when one D/A converter is used, and the DAi register (i = 0, 1) of the unused D/A converter is set to 00h. The current flown into the resistor ladder in the A/D converter is excluded. IVREF flows even if VCUT bit in the AD0CON1 register is set to 0 (VREF not connected).

VCC1 = VCC2 = 3.3 V

Timing Requirements
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.34 External Clock Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External clock input cycle time	62.5		ns
tw(H)	External clock input high (“H”) pulse width	27.5		ns
tw(L)	External clock input low (“L”) pulse width	27.5		ns
tr	External clock rise time		5	ns
tf	External clock fall time		5	ns

Table 24.35 Timer A Input (Count Source Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	100		ns
tw(TAH)	TAiIN input high (“H”) pulse width	40		ns
tw(TAL)	TAiIN input low (“L”) pulse width	40		ns

i = 0 to 4

Table 24.36 Timer A Input (Gate Signal Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	400		ns
tw(TAH)	TAiIN input high (“H”) pulse width	200		ns
tw(TAL)	TAiIN input low (“L”) pulse width	200		ns

i = 0 to 4

Table 24.37 Timer A Input (External Trigger Input in One-Shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	200		ns
tw(TAH)	TAiIN input high (“H”) pulse width	100		ns
tw(TAL)	TAiIN input low (“L”) pulse width	100		ns

i = 0 to 4

Table 24.38 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(TAH)	TAiIN input high (“H”) pulse width	100		ns
tw(TAL)	TAiIN input low (“L”) pulse width	100		ns

i = 0 to 4

VCC1 = VCC2 = 3.3 V

Timing Requirements
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.39 Timer A Input (Counter Increment/Decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(UP)	TAiOUT input cycle time	2000		ns
tw(UPH)	TAiOUT input high (“H”) pulse width	1000		ns
tw(UPL)	TAiOUT input low (“L”) pulse width	1000		ns
tsu(UP-TIN)	TAiOUT input setup time	400		ns
th(TIN-UP)	TAiOUT input hold time	400		ns

i = 0 to 4

Table 24.40 Timer A Input (Two-Phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	2		µs
tsu(TAIN-TAOUT)	TAiOUT input setup time	500		ns
tsu(TAOUT-TAIN)	TAiIN input setup time	500		ns

i = 0 to 4

Table 24.41 Timer B Input (Count Source Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time (counted on one edge)	100		ns
tw(TBH)	TBiIN input high (“H”) pulse width (counted on one edge)	40		ns
tw(TBL)	TBiIN input low (“L”) pulse width (counted on one edge)	40		ns
tc(TB)	TBiIN input cycle time (counted on both edges)	200		ns
tw(TBH)	TBiIN input high (“H”) pulse width (counted on both edges)	80		ns
tw(TBL)	TBiIN input low (“L”) pulse width (counted on both edges)	80		ns

i = 0 to 5

Table 24.42 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time	400		ns
tw(TBH)	TBiIN input high (“H”) pulse width	200		ns
tw(TBL)	TBiIN input low (“L”) pulse width	200		ns

i = 0 to 5

Table 24.43 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TB)	TBiIN input cycle time	400		ns
tw(TBH)	TBiIN input high (“H”) pulse width	200		ns
tw(TBL)	TBiIN input low (“L”) pulse width	200		ns

i = 0 to 5

VCC1 = VCC2 = 3.3 V

Timing Requirements
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.44 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(AD)	$\overline{\text{ADTRG}}$ input cycle time (required for trigger)	1000		ns
tw(ADL)	$\overline{\text{ADTRG}}$ input low (“L”) pulse width	125		ns
tw(ADH)	$\overline{\text{ADTRG}}$ input high (“H”) pulse width	3		ϕAD

Table 24.45 Serial Interface

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	CLKi input cycle time	200		ns
tw(CKH)	CLKi input high (“H”) pulse width	100		ns
tw(CKL)	CLKi input low (“L”) pulse width	100		ns
td(C-Q)	TXDi output delay time		80	ns
th(C-Q)	TXDi output hold time	0		ns
tsu(D-C)	RXDi input setup time	80		ns
th(C-D)	RXDi input hold time	90		ns

i=0 to 4

Table 24.46 External Interrupt $\overline{\text{INTi}}$ Input (Edge Sensitive)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(INH)	$\overline{\text{INTi}}$ input high (“H”) pulse width	250		ns
tw(INL)	$\overline{\text{INTi}}$ input low (“L”) pulse width	250		ns

i=0 to 5

VCC1 = VCC2 = 3.3 V

Timing Requirements
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.47 Memory Expansion Mode and Microprocessor Mode

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tac1(RD-DB)	Data input access time (RD standard)		(note 1)	ns
tac1(AD-DB)	Data input access time (AD standard, CS standard)		(note 1)	ns
tac2(RD-DB)	Data input access time (RD standard, when accessing a space with the multiplexed bus)		(note 1)	ns
tac2(AD-DB)	Data input access time (AD standard, when accessing a space with the multiplexed bus)		(note 1)	ns
tsu(DB-BCLK)	Data input setup time	27		ns
tsu(RDY-BCLK)	$\overline{\text{RDY}}$ input setup time	30		ns
tsu(HOLD-BCLK)	$\overline{\text{HOLD}}$ input setup time	40		ns
th(RD-DB)	Data input hold time	0		ns
th(BCLK-RDY)	$\overline{\text{RDY}}$ input hold time	0		ns
th(BCLK-HOLD)	$\overline{\text{HOLD}}$ input hold time	0		ns
td(BCLK-HLDA)	$\overline{\text{HLDA}}$ output delay time		25	ns

NOTE:

1. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equations. Insert wait states or lower the operation frequency, f(BCLK), if the calculated value is negative.

$$\text{tac1(RD-DB)} = \frac{10^9 \times m}{\text{f(BCLK)} \times 2} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) + 1)$$

$$\text{tac1(AD-DB)} = \frac{10^9 \times n}{\text{f(BCLK)}} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ n = a + b)$$

$$\text{tac2(RD-DB)} = \frac{10^9 \times m}{\text{f(BCLK)} \times 2} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) - 1)$$

$$\text{tac2(AD-DB)} = \frac{10^9 \times p}{\text{f(BCLK)} \times 2} - 35\ \text{[ns] (if external bus cycle is } a\phi + b\phi,\ p = \{(a + b - 1) \times 2\} + 1)$$

VCC1 = VCC2 = 3.3 V

Switching Characteristics
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.48 Memory Expansion Mode and Microprocessor Mode (when accessing external memory space)

Symbol	Parameter	Measurement Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	See Figure 24.2		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		0		ns
th(RD-AD)	Address output hold time (RD standard)(3)		0		ns
th(WR-AD)	Address output hold time (WR standard)(3)		(note 1)		ns
td(BCLK-CS)	Chip-select signal output delay time			18	ns
th(BCLK-CS)	Chip-select signal output hold time (BCLK standard)		0		ns
th(RD-CS)	Chip-select signal output hold time (RD standard)(3)		0		ns
th(WR-CS)	Chip-select signal output hold time (WR standard)(3)		(note 1)		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		-3		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(DB-WR)	Data output delay time (WR standard)		(note 2)		ns
th(WR-DB)	Data output hold time (WR standard)(3)		(note 1)		ns
tw(WR)	WR output width		(note 2)		ns

NOTES:

1. Values, which depend on BCLK frequency, can be obtained from the following equations.

$$th(WR\text{-}DB) = \frac{10^9}{f(BCLK) \times 2} - 15 \text{ [ns]}$$

$$th(WR\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

2. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equations.

$$td(DB\text{-}WR) = \frac{10^9 \times m}{f(BCLK)} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi, m = b)$$

$$tw(WR) = \frac{10^9 \times n}{f(BCLK) \times 2} - 15 \text{ [ns] (if external bus cycle is } a\phi + b\phi, n = (b \times 2) - 1)$$

3. tc [ns] is added when recovery cycle is inserted.

VCC1 = VCC2 = 3.3 V

Switching Characteristics
(VCC1 = VCC2 = 3.0 to 3.6 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 24.49 Memory Expansion Mode and Microprocessor Mode (when accessing external memory space with multiplexed bus)

Symbol	Parameter	Measurement Condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	See Figure 24.2		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		0		ns
th(RD-AD)	Address output hold time (RD standard)(5)		(note 1)		ns
th(WR-AD)	Address output hold time (WR standard)(5)		(note 1)		ns
td(BCLK-CS)	Chip-select signal output delay time			18	ns
th(BCLK-CS)	Chip-select signal output hold time (BCLK standard)		0		ns
th(RD-CS)	Chip-select signal output hold time (RD standard)(5)		(note 1)		ns
th(WR-CS)	Chip-select signal output hold time (WR standard)(5)		(note 1)		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		-3		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(DB-WR)	Data output delay time (WR standard)		(note 2)		ns
th(WR-DB)	Data output hold time (WR standard)(5)		(note 1)		ns
td(BCLK-ALE)	ALE signal output delay time (BCLK standard)			18	ns
th(BCLK-ALE)	ALE signal output hold time (BCLK standard)		-2		ns
td(AD-ALE)	ALE signal output delay time (address standard)		(note 3)		ns
th(ALE-AD)	ALE signal output hold time (address standard)		(note 4)		ns
tdz(RD-AD)	Address output float start time			8	ns

NOTES:

1. Values, which depend on BCLK frequency, can be obtained from the following equations.

$$\mathrm{th(RD\text{-}AD)} = \frac{10^9}{\mathrm{f(BCLK)} \times 2} - 10 \text{ [ns]}$$

$$\mathrm{th(WR\text{-}AD)} = \frac{10^9}{\mathrm{f(BCLK)} \times 2} - 10 \text{ [ns]}$$

$$\mathrm{th(RD\text{-}CS)} = \frac{10^9}{\mathrm{f(BCLK)} \times 2} - 10 \text{ [ns]}$$

$$\mathrm{th(WR\text{-}CS)} = \frac{10^9}{\mathrm{f(BCLK)} \times 2} - 10 \text{ [ns]}$$

$$\mathrm{th(WR\text{-}DB)} = \frac{10^9}{\mathrm{f(BCLK)} \times 2} - 15 \text{ [ns]}$$

2. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$\mathrm{td(DB\text{-}WR)} = \frac{10^9 \times m}{\mathrm{f(BCLK)} \times 2} - 25 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) - 1)$$

3. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$\mathrm{td(AD\text{-}ALE)} = \frac{10^9 \times n}{\mathrm{f(BCLK)} \times 2} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ n = a)$$

4. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$\mathrm{th(ALE\text{-}AD)} = \frac{10^9 \times n}{\mathrm{f(BCLK)} \times 2} - 10 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ n = a)$$

5. tc [ns] is added when recovery cycle is inserted.

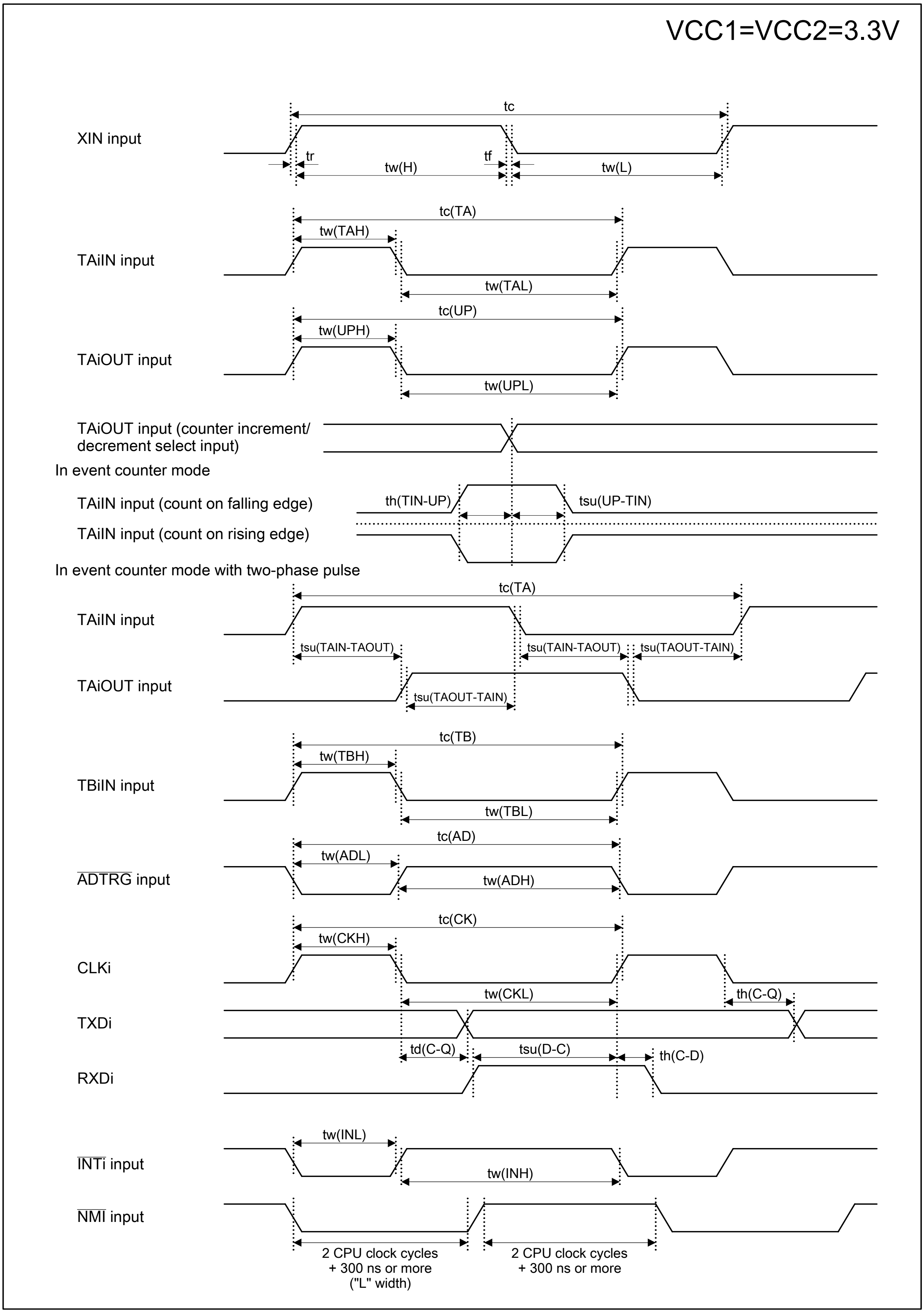
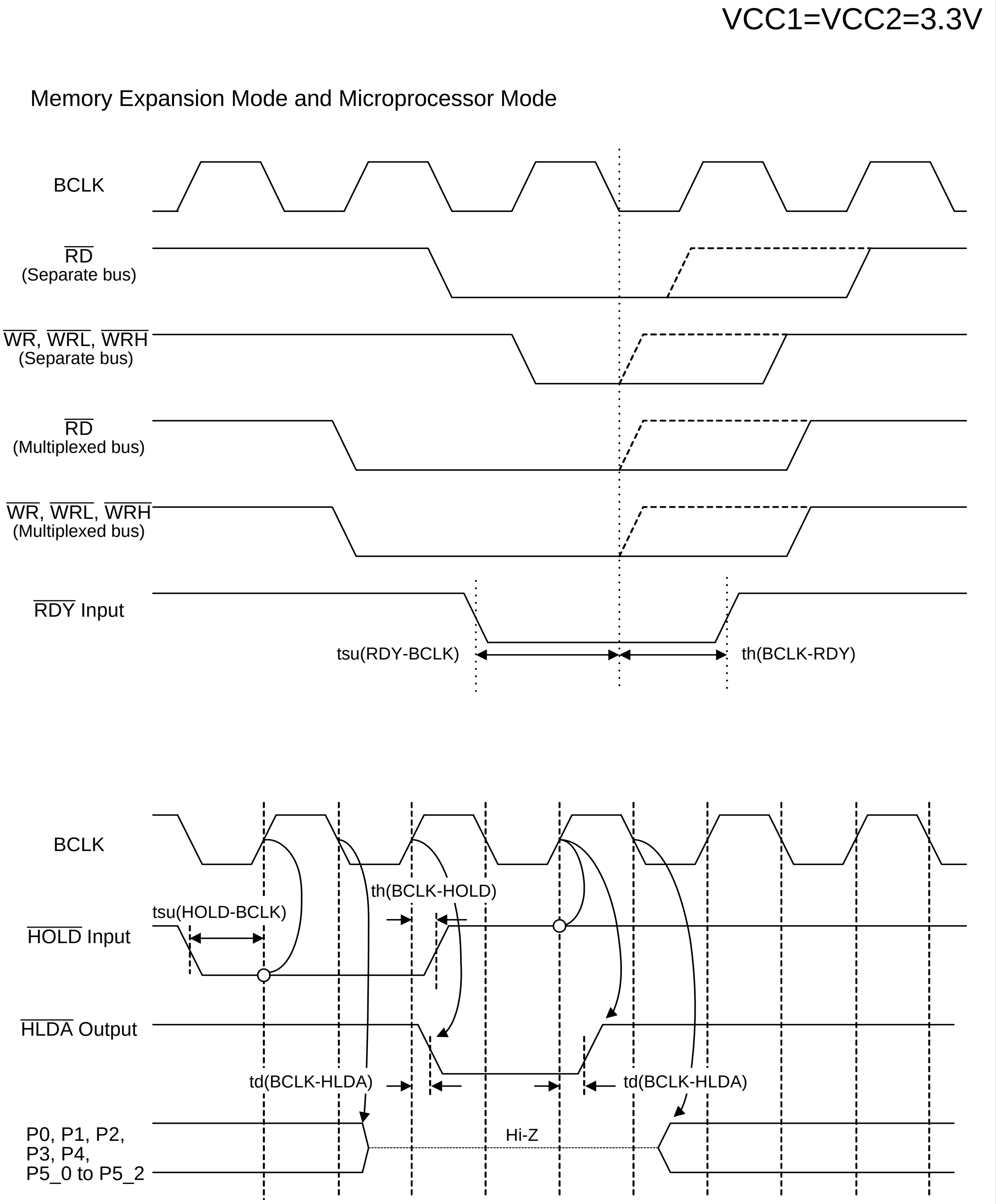


Figure 24.7 VCC1 = VCC2 = 3.3 V Timing Diagram (1)



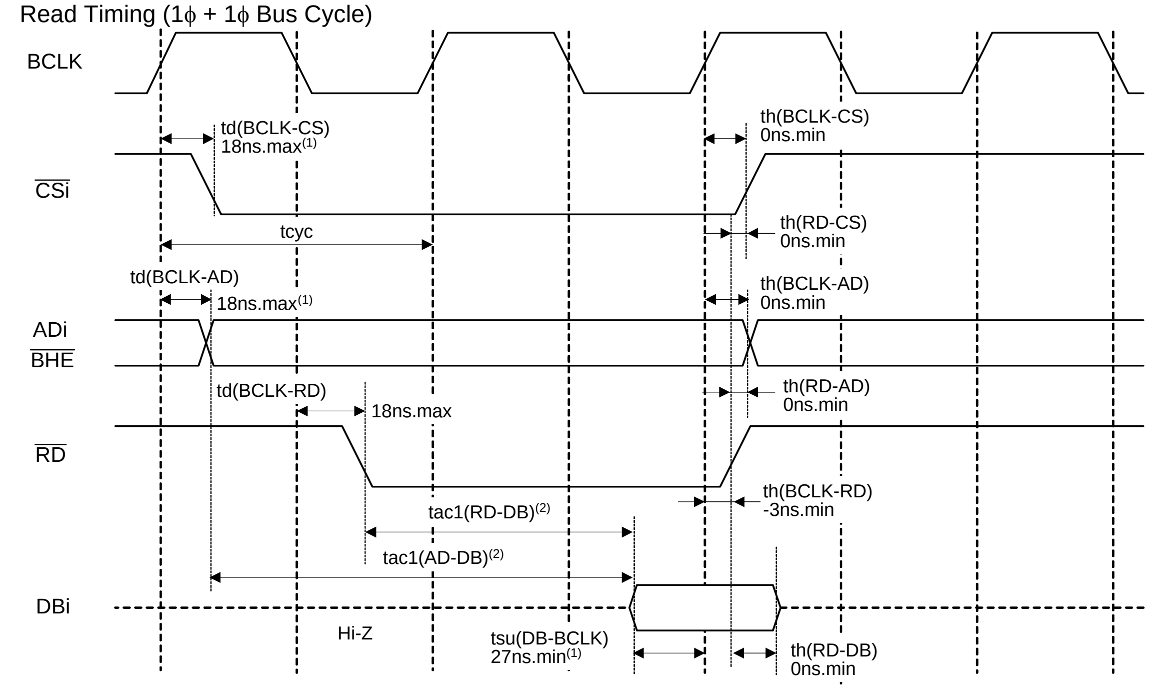
Measurement Conditions
-VCC1 = VCC2 = 3.0 to 3.6 V
-Input high and low voltage: VIH = 2.4 V, VIL = 0.6 V
-Output high and low voltage: VOH = 1.5 V, VOL = 1.5 V

Figure 24.8 VCC1 = VCC2 = 3.3 V Timing Diagram (2)

Memory Expansion Mode and Microprocessor Mode
(when accessing an external memory space)

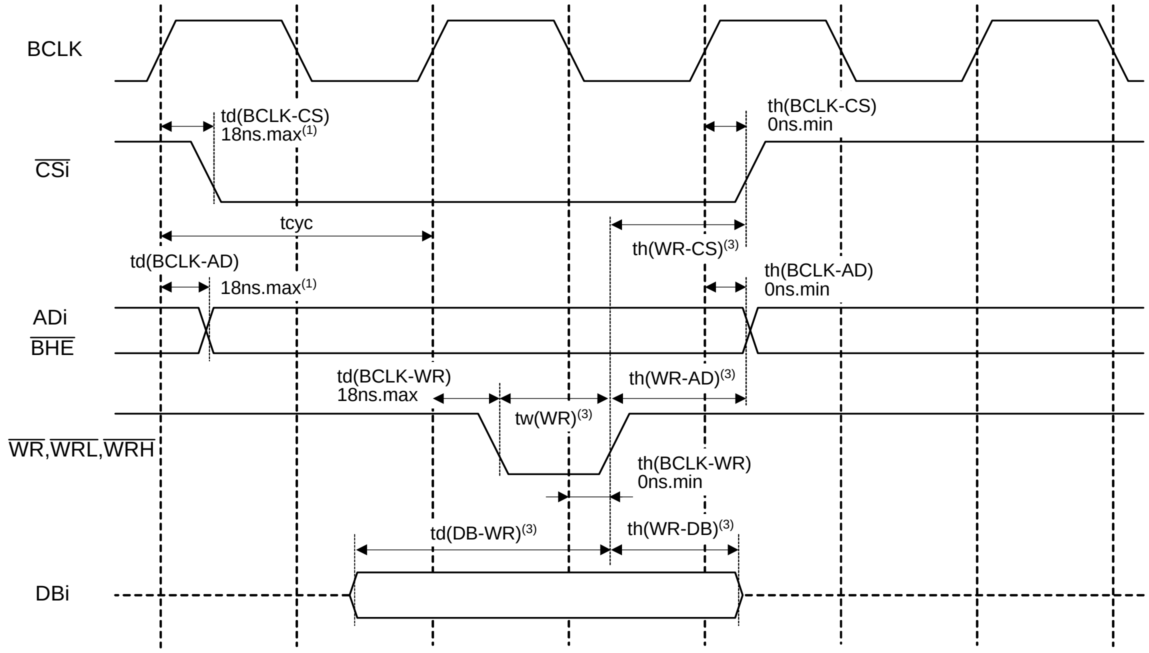
VCC1=VCC2=3.3V

Read Timing (1ϕ + 1ϕ Bus Cycle)



NOTES:
1. Values guaranteed only when the MCU is used stand-alone.
A maximum of 35 ns is guaranteed for td(BCLK-AD) + tsu(DB-BCLK).
2. Varies with operation frequency:
tac1(RD-DB) = (tcyc / 2 x m - 35) ns.max (if external bus cycle aϕ + bϕ, m = (b x 2) + 1)
tac1(AD-DB) = (tcyc x n - 35) ns.max (if external bus cycle aϕ + bϕ, n = a + b)

Write Timing (1ϕ + 1ϕ Bus Cycle)



NOTES:
3. Varies with operation frequency:
td(DB-WR) = (tcyc x m - 20) ns.min
(if external bus cycle aϕ + bϕ, m = b)
th(WR-DB) = (tcyc / 2 - 15) ns.min
th(WR-AD) = (tcyc / 2 - 10) ns.min
th(WR-CS) = (tcyc / 2 - 10) ns.min
tw(WR) = (tcyc / 2 x n - 15) ns.min
(if external bus cycle aϕ + bϕ, n = (b x 2) - 1)

Measurement Conditions:
- VCC1 = VCC2 = 3.0 to 3.6 V
- Input high and low voltage: VIH = 1.5 V, VIL = 0.5 V
- Output high and low voltage: VOH = 1.5 V, VOL = 1.5 V

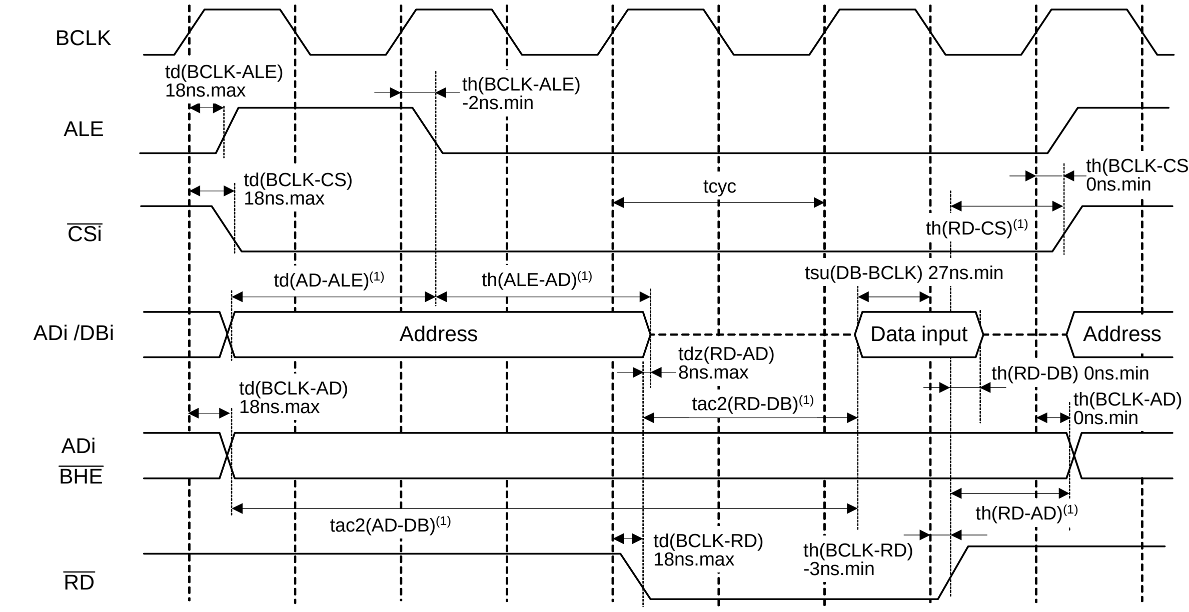
$$tcyc = \frac{10^9}{f(BCLK)}$$

Figure 24.9 VCC1 = VCC2 = 3.3 V Timing Diagram (3)

Memory Expansion Mode and Microprocessor Mode
(when accessing an external memory space with the multiplexed bus)

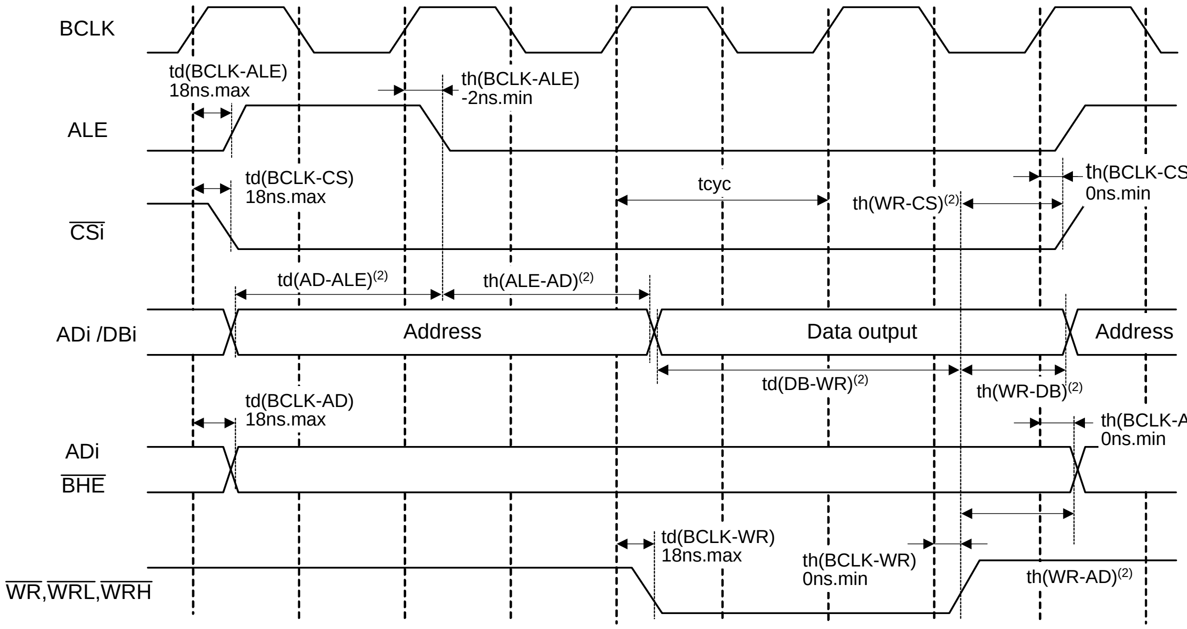
VCC1=VCC2=3.3V

Read Timing (2ϕ + 2ϕ Bus Cycle)



NOTES:
1. Varies with operation frequency:
td(AD-ALE) = (tcyc / 2 x n - 20) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(ALE-AD) = (tcyc / 2 x n - 10) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(RD-AD) = (tcyc / 2 - 10) ns.min, th(RD-CS) = (tcyc / 2 - 10) ns.min
tac2(RD-DB) = (tcyc / 2 x m - 35) ns.max (if external bus cycle aϕ + bϕ, m = (b x 2) - 1)
tac2(AD-DB) = (tcyc / 2 x p - 35) ns.max (if external bus cycle aϕ + bϕ, p = {(a + b - 1) x 2} + 1)

Write Timing (2ϕ + 2ϕ Bus Cycle)



NOTES:
1. Varies with operation frequency:
td(AD-ALE) = (tcyc / 2 x n - 20) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(ALE-AD) = (tcyc / 2 x n - 10) ns.min (if external bus cycle aϕ + bϕ, n = a)
th(WR-AD) = (tcyc / 2 - 10) ns.min, th(WR-CS) = (tcyc / 2 - 10) ns.min
th(WR-DB) = (tcyc / 2 - 15) ns.min
td(DB-WR) = (tcyc / 2 x m - 25) ns.min (if external bus cycle aϕ + bϕ, m = (b x 2) - 1)

$$\text{tcyc}=\frac{10^9}{\text{f(BCLK)}}$$

Measurement Conditions:
- VCC1 = VCC2 = 3.0 to 3.6 V
- Input high and low voltage VIH = 1.5 V, VIL = 0.5 V
- Output high and low voltage VOH = 1.5 V, VOL = 1.5 V

Figure 24.10 VCC1 = VCC2 = 3.3 V Timing Diagram (4)

25. Usage Notes

25.1 Power Supply

25.1.1 Power-on

At power-on, supply voltage applied to the VCC1 must meet the SVCC standard.
(Technical update: TN-M16C-116-0311)

Table 25.1 Supply Voltage Power-up Slope

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
SVCC	Supply voltage power-up slope (supply voltage range: 0 V to 2.0 V)	0.05			V/ms

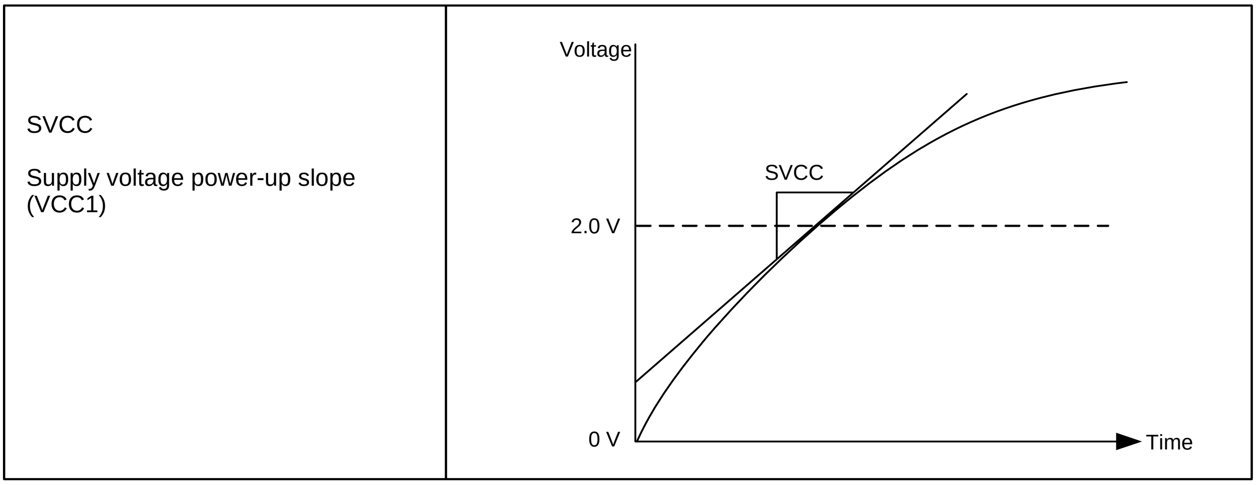


Figure 25.1 SVCC Timing

25.1.2 Power Supply Ripple

Stabilize supply voltage to meet the power supply standard listed in Table 25.2.

Table 25.2 Power Supply Ripple

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
f(ripple)	Power supply ripple tolerable frequency (VCC1)				10	kHz
Vp-p(ripple)	Power supply ripple voltage fluctuation range	(VCC1 = 5 V)			0.5	V
		(VCC1 = 3.3 V)			0.3	V
VCC(|ΔV/ΔT|)	Power supply ripple voltage fluctuation rate	(VCC1 = 5 V)			0.3	V/ms
		(VCC1 = 3.3 V)			0.3	V/ms

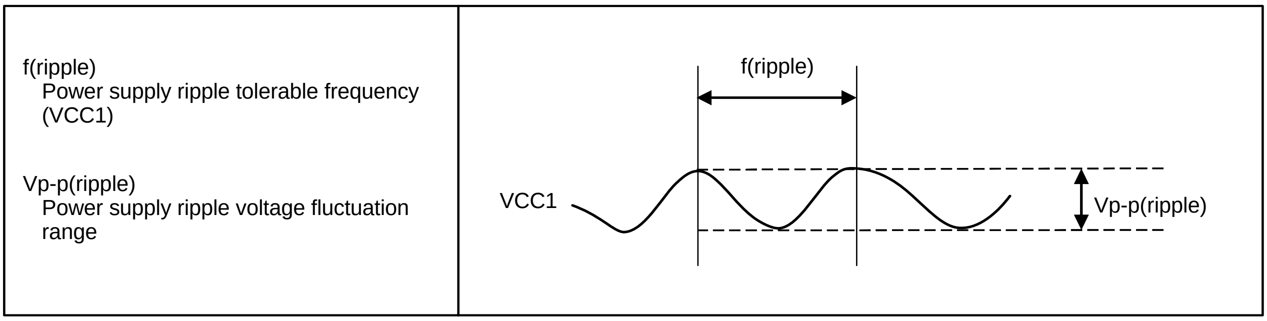

f(ripple)
Power supply ripple tolerable frequency (VCC1)
Vp-p(ripple)
Power supply ripple voltage fluctuation range
f(ripple)
VCC1
Vp-p(ripple)

Figure 25.2 Power Supply Fluctuation Timing

25.1.3 Noise

Use thick and shortest possible wiring to connect a bypass capacitor (0.1 µF or more) between VCC and VSS.

25.2 Special Function Registers (SFRs)

25.2.1 100 Pin-Package

Set addresses 03CBh, 03CEh, 03CFh, 03D2h, and 03D3h to FFh after reset when using the 100-pin package. Address 03DCh must be set to 00h after reset.

25.2.2 Register Settings

Table 25.3 lists registers containing write-only bits. Read-modify-write instructions cannot be used to set these registers. If these registers are set using a read-modify-write instruction, undefined values are read from the write-only bits in the register and written back to these bits. Table 25.4 lists read-modify-write instructions.

When establishing new values by modifying previous ones, write the previous values into RAM as well as to the register. Change the contents of the RAM and then transfer the new values to the register.

Table 25.3 Registers with Write-Only Bits

Register	Address	Register	Address
WDTS register	000Eh	U3TB register	032Bh to 032Ah
U1BRG register	02E9h	U2BRG register	0339h
U1TB register	02EBh to 02EAh	U2TB register	033Bh to 033Ah
U4BRG register	02F9h	UDF register	0344h
U4TB register	02FBh to 02FAh	TA0 register(1)	0347h to 0346h
TA11 register	0303h to 0302h	TA1 register(1)	0349h to 0348h
TA21 register	0305h to 0304h	TA2 register(1)	034Bh to 034Ah
TA41 register	0307h to 0306h	TA3 register(1)	034Dh to 034Ch
DTT register	030Ch	TA4 register(1)	034Fh to 034Eh
ICTB2 register	030Dh	U0BRG register	0369h
U3BRG register	0329h	U0TB register	036Bh to 036Ah

NOTE:

1. In one-shot timer mode and pulse width modulation mode only.

Table 25.4 Read-Modify-Write Instructions

Function	Mnemonic
Transfer	MOVDir
Bit manipulation	BCLR, BMCnd, BNOT, BSET, BTSTC, BTSTS
Shift	ROLC, RORC, ROT, SHA, SHANC, SHL, SHLNC
Arithmetic	ABS, ADC, ADCF, ADD, ADDX, DADC, DADD, DEC, DSBB, DSUB, EXTS, INC, MUL, MULEX, MULU, NEG, SBB, SUB, SUBX
Logical	AND, NOT, OR, XOR
Jump	ADJNZ, SBJNZ

25.3 Processor Mode

- When a port shares its pin with a bus control pin, such as address bus, data bus, $\overline{\text{CS}}$, or $\overline{\text{RD}}$, set its corresponding Port Pi Register (i = 0 to 15) and Port Pi Direction Register after entering single-chip mode.
(Technical update: TN-M16C-49-0004)

- Rewriting bits PM01 and PM00 in the PM0 register places the MCU in the corresponding processor mode regardless of CNVSS input level. When setting bits PM01 and PM00 to 01b (memory expansion mode) or 11b (microprocessor mode), do not set simultaneously with bits PM07 to PM02. First, set bits PM02, PM05 and PM04, and PM07 in the PM0 register, and also set bits PM11 and PM10, PM15 and PM14 in the PM1 register. Then, set bits PM01 and PM00.

- When the MCU starts up in microprocessor mode, the internal ROM cannot be accessed.

25.4 Bus

25.4.1 $\overline{\text{HOLD}}$ Input

If the $\overline{\text{HOLD}}$ input is used, set bits PD4_0 to PD4_7 in the PD4 register and bits PD5_0 to PD5_2 in the PD5 register to 0 (input mode) prior to setting bits PM01 and PM00 in the PM0 register to 01b (memory expansion mode) or to 11b (microprocessor mode) to switch from single-chip mode to memory expansion mode or microprocessor mode.

(Technical update: TN-M16C-59-0008)

25.5 Clock Generation Circuits

25.5.1 Main Clock

- If the main clock is selected as the CPU clock while an external clock is applied to the XIN pin, do not stop the external clock.
  (Technical update: TN-M16C-109-0309)

- When a clock applied to the XIN pin is used for the CPU clock, do not set the CM05 bit in the CM0 register to 1 (stopped).

25.5.2 Sub Clock

25.5.2.1 To Oscillate Sub Clock

To oscillate the sub clock, set the CM07 bit in the CM0 register to 0 (clock other than the sub clock) and the CM03 bit to 1 (XCIN-XOUT drive capability = high). Then, set the CM04 bit in the CM0 register to 1 (XCIN-XCOUT oscillation function). Once the sub clock becomes stabilized, set the CM03 bit to 0 (XCIN-XOUT drive capability = low).
After the above procedure, the sub clock can be used as the CPU clock, or the count source for timer A and timer B.
(Technical update: TN-16C-119A/EA)

25.5.2.2 Oscillation Parameter Matching

If an oscillation circuit constant matching for the sub clock oscillation circuit has only been evaluated with the drive capability = high, the constant matching for drive capability = low must also be evaluated.
Contact your oscillator manufacturer for details on the oscillation circuit constant matching.

25.5.3 Clock Dividing Ratio

To change bits MCD4 to MCD0, set the PM12 bit in the PM1 register to 0 (no wait state).

25.5.4 Power Consumption Control

Stabilize the main clock, sub clock, or PLL clock prior to switching the clock source for the CPU clock to one of these clocks.
In the flash memory version, if CPU clock frequency is set to be 20MHz or more, first set bits MCD4 to MCD0 in the MCD register to 00010b (divide-by-2 mode), and then set to 10010b (no division mode). Also, insert at least eight NOP instructions after the instruction to set to no division mode.

25.5.4.1 Wait Mode

- When entering wait mode, the instructions following the WAIT instruction are stored into the instruction queue, and the program stops. Insert at least 4 NOP instructions after the WAIT instruction.

- To enter wait mode, execute the WAIT instruction while a high-level (“H”) signal is applied to the $\overline{\text{NMI}}$ pin.

25.5.4.2 Stop Mode

- The MCU cannot enter stop mode if a low-level (“L”) signal is applied to the $\overline{\text{NMI}}$ pin. Apply an “H” signal to enter stop mode.

- To exit stop mode by reset, apply an “L” signal to $\overline{\text{RESET}}$ pin until a main clock oscillation stabilizes.

- If using the $\overline{\text{NMI}}$ interrupt to exit stop mode, use the following procedure to set the CM10 bit in the CM1 register to 1 (all clocks stopped).
  (Technical update: TN-16C-127A/EA)

  (1) Exit stop mode using the $\overline{\text{NMI}}$ interrupt.
  (2) Generate a dummy interrupt.
  (3) Set the CM10 bit to 1 (all clocks stopped).

```
e.g.,   int     #63        ; dummy interrupt
        bset    CM1        ; all clocks stopped

        /*dummy interrupt routine*/
        dummy
        reit
```

- When entering stop mode, the instructions following CM10 = 1 instruction are stored into the instruction queue, and the program stops. When stop mode is exited, the instruction lined in the queue is executed before the exit interrupt routine is handled. Insert a jmp.b instruction as follows after the instruction to set the CM10 bit to 1.
  (Technical update: TN-16C-124A/EA)

```
        fset I              ; I flag is set to 1
        bset 0,   cm1       ; all clocks stopped (stop mode)
        jmp.b LABEL_001     ; jmp.b instruction executed (no instruction between jmp.b and LABEL.)
LABEL_001:
        nop                 ; nop(1)
        nop                 ; nop(2)
        nop                 ; nop(3)
        nop                 ; nop(4)
        mov.b #0, prcr      ; protection set
                .
                .
                .
```

25.5.4.3 Suggestions to Reduce Power Consumption

The followings are suggestions to reduce power consumption when programming or designing systems.

Ports:

- Through current may flow into floating input pins. Set unassigned pins to input mode and connect them to VSS via a resistor (pull down), or set unassigned pins to output mode and leave them open.

A/D converter:

- When the A/D conversion is not performed, set the VCUT bit in the AD0CON1 register to 0 (VREF not connected). When the A/D conversion is performed, set the VCUT bit to 1 (VREF connection) and wait 1 µs or more to start the A/D conversion.

D/A converter:

- When the D/A conversion is not performed, set the DAiE bit (i = 0, 1) in the DACON register to 0 (output disabled) and the DAi register to 00h.

Peripheral function clock stop:

- When entering wait mode from main clock mode, on-chip oscillator mode, or on-chip oscillator low-power consumption mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop peripheral function clock source (fPFC). However, fC32 does not stop by setting the CM02 bit to 1.

- In low-speed mode or low-power consumption mode, set the CM02 bit to 0 (peripheral clocks do not stop in wait mode) prior to entering wait mode.
  (Technical update: TN-M16C-69-0104)

25.6 Protection

The PRC2 bit in the PRCR register becomes 0 (write disable) by a write to the SFR area after the PRC2 bit is set to 1 (write enable). Set a register protected by the PRC2 bit immediately after the PRC2 bit is set to 1. Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.

25.7 Interrupts

25.7.1 ISP Setting

After reset, ISP is initialized to 000000h. The program may go out of control if an interrupt is acknowledged before setting a value to ISP. Therefore, ISP must be set before any interrupt request is acknowledged. Setting ISP to an even address allows interrupt sequences to be executed at a higher speed.

To use the $\overline{\mathrm{NMI}}$ interrupt, set ISP at the very beginning of the program. The $\overline{\mathrm{NMI}}$ interrupt can be acknowledged after the first instruction has been executed after reset.

25.7.2 $\overline{\mathrm{NMI}}$ Interrupt

- The $\overline{\mathrm{NMI}}$ interrupt cannot be disabled. Connect the $\overline{\mathrm{NMI}}$ pin to VCC1 via a resistor (pull-up) when not in use.

- The P8_5 bit in the P8 register indicates the voltage level applied to the $\overline{\mathrm{NMI}}$ pin. Read the P8_5 bit only to determine the pin level after the $\overline{\mathrm{NMI}}$ interrupt occurs.

25.7.3 $\overline{\mathrm{INT}}$ Interrupt

- Edge Sensitive
  Each “H” or “L” width of the signal applied to pins $\overline{\mathrm{INT0}}$ to $\overline{\mathrm{INT5}}$ must be 250 ns or more regardless of the CPU clock frequency.

- Level Sensitive
  Each “H” or “L” width of the signal applied to pins $\overline{\mathrm{INT0}}$ to $\overline{\mathrm{INT5}}$ must be one CPU clock cycle + 200 ns or more. For example, each “H” or “L” width must be 234 ns or more if the CPU clock is 30 MHz.

- The IR bit in the INTiIC register (i = 0 to 5) may become 1 (interrupt requested) when the polarity settings of pins $\overline{\mathrm{INT0}}$ to $\overline{\mathrm{INT5}}$ are changed. Set the IR bit to 0 (interrupt not requested) after the polarity setting is changed.
  Figure 25.3 shows a procedure to set the $\overline{\mathrm{INTi}}$ interrupt source (i = 0 to 5).

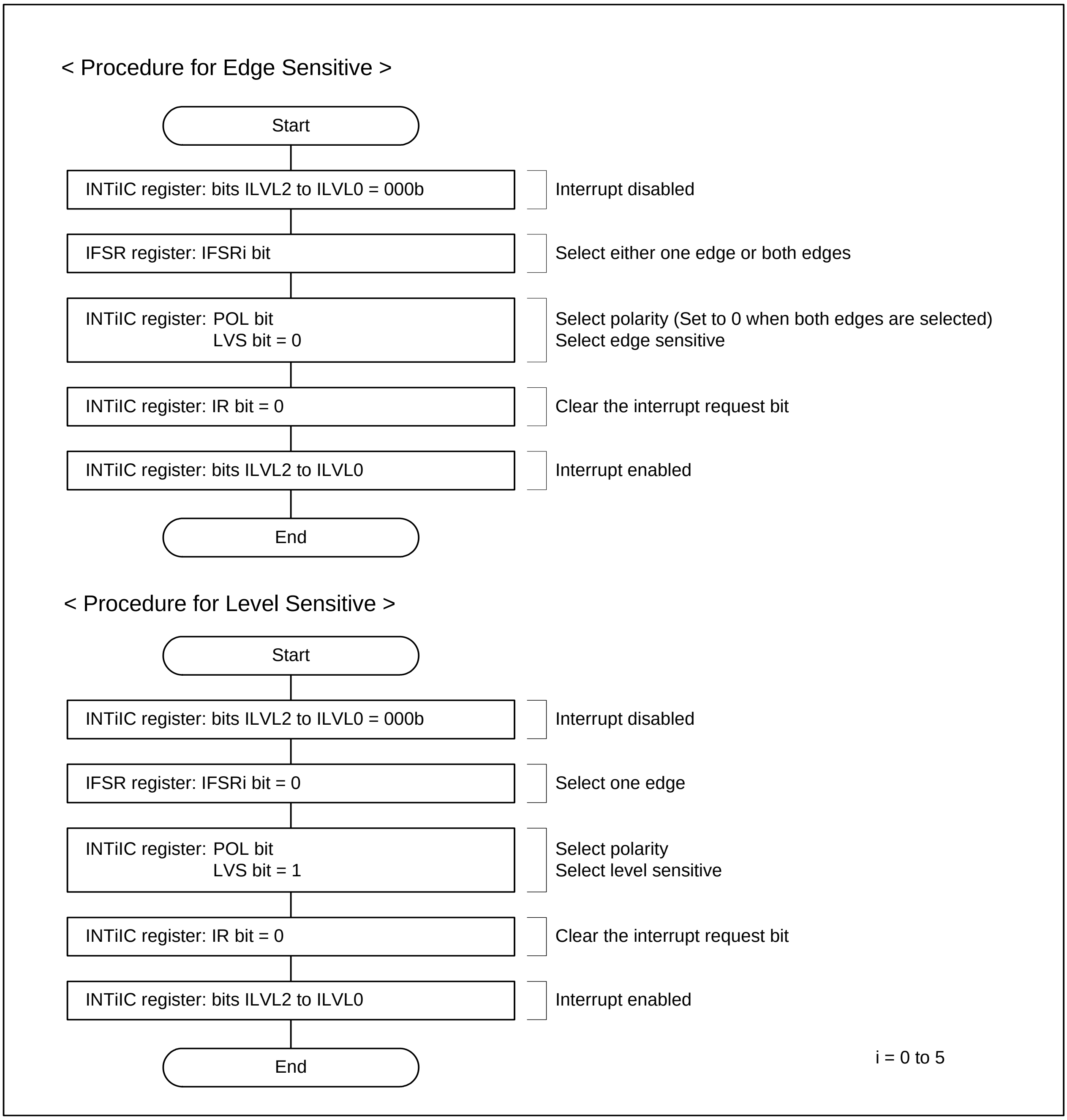

< Procedure for Edge Sensitive >
Start
INTiIC register: bits ILVL2 to ILVL0 = 000b
Interrupt disabled
IFSR register: IFSRi bit
Select either one edge or both edges
INTiIC register: POL bit
LVS bit = 0
Select polarity (Set to 0 when both edges are selected)
Select edge sensitive
INTiIC register: IR bit = 0
Clear the interrupt request bit
INTiIC register: bits ILVL2 to ILVL0
Interrupt enabled
End
< Procedure for Level Sensitive >
Start
INTiIC register: bits ILVL2 to ILVL0 = 000b
Interrupt disabled
IFSR register: IFSRi bit = 0
Select one edge
INTiIC register: POL bit
LVS bit = 1
Select polarity
Select level sensitive
INTiIC register: IR bit = 0
Clear the interrupt request bit
INTiIC register: bits ILVL2 to ILVL0
Interrupt enabled
End
i = 0 to 5

Figure 25.3 Procedure to Set the $\overline{\text{INTi}}$ Interrupt Source (i = 0 to 5)

25.7.4 Changing Interrupt Control Register

To change the Interrupt Control Register while an interrupt request is disabled, use the following instructions.

Changing IR bit:
The IR bit may not be changed to 0 (interrupt not requested) by writing, depending on which instruction is used. If this causes a problem, use MOV instruction to change the register. (Technical update: TN-M16C-85-0204)

Changing any bits other than IR bit:
If an interrupt request is generated while writing to the corresponding Interrupt Control Register with instructions such as MOV, the IR bit may not become 1 (interrupt requested) and the interrupt is not acknowledged. If this causes a problem, use the following instructions to write to the register:
AND, OR, BCLR, BSET

25.7.5 Changing RLVL Register

The DMAII bit in the RLVL register is undefined after reset. To use interrupt priority level 7 for an interrupt, set it to 0 before setting the Interrupt Control Register.

25.8 DMAC

- Set the DMAC-associated registers while bits MDi1 and MDi0 (i = 0 to 3) in the channel i are set to 00b (DMA disabled). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure, which enables the DMA request of the channel i to be acknowledged.

- Write a 1 (requested) to the DRQ bit when setting the DMiSL register.

In the M32C/80 Series, if a DMA request is generated but a receiving channel is not ready(1), a DMA transfer does not occur and the DRQ bit becomes 0.

NOTE:

1. Bits MDi1 and MDi0 are set to 00b or the DCTi register is 0000h (transferred 0 time).

- To start a DMA transfer using a software trigger, set bits DSR and DRQ in the DMiSL register to 1 simultaneously.

e.g.,

```
    OR.B #0A0h, DMiSL        ; set bits DSR and DRQ to 1 simultaneously
```

- While the DCTi register in the channel i is set to 1, do not generate a DMA request in the channel i in the timing that bits MDi1 and MDi0 in the DMDj register (j = 0, 1) corresponding to the channel i are set to 01b (single transfer) or 11b (repeat transfer). (Technical update: TN-M16C-88-0209)

- Select a peripheral function used as a DMA request source after setting the DMA-associated registers. When the $\overline{\text{INT}}$ interrupt is selected as a DMA request source, do not set the DCTi register to 1.

- Wait six CPU clock cycles or more by a program to enable DMA after setting the DMiSL register(2).

NOTE:

2. To enable DMA means changing bits MDi1 and MDi0 in the DMDj register from 00b (DMA disabled) to 01b (single transfer) or 11b (repeat transfer).

25.9 Timers

25.9.1 Timer A, Timer B

Timers are stopped after reset. Set the TAiS (i = 0 to 4) or TBjS (j = 0 to 5) bit in the TABSR or TBSR register to 1 (count starts) after setting timer operating mode, count source, and counter value.
Change the following registers and bits while the corresponding timer is stopped (the TAiS or TBjS bit is set to 0 (count stops)).

- Registers TAiMR and TBjMR
- UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- TRGSR register

25.9.2 Timer A

25.9.2.1 Timer A (Timer Mode)

- The TAiS bit (i = 0 to 4) in the TABSR register is set to 0 (count stops) after reset. Set the TAiS bit to 1 (count starts) after selecting timer operating mode and setting the TAi register.

- The TAi register indicates a counter value while counting at any given time. However, FFFFh can be read in the reload timing. When the TAi register is set while a counter is stopped, the setting value can be read until a counter is started.

25.9.2.2 Timer A (Event Counter Mode)

- The TAiS bit (i = 0 to 4) is set to 0 (count stops) after reset. Set the TAiS bit to 1 (count starts) after selecting timer operating mode and setting the TAi register.

- The TAi register indicates a counter value while counting at any given time. In the reload timing, however, FFFFh can be read if the timer underflows, or 0000h if the timer overflows. When the TAi register is set while the counter is stopped, the setting value can be read until a counter is started.

25.9.2.3 Timer A (One-Shot Timer Mode)

- The TAiS bit (i = 0 to 4) in the TABSR register is set to 0 (count stops) after reset. Set the TAiS bit to 1 (count starts) after selecting timer operating mode and setting the TAi register.

- The following occurs when the TAiS bit in the TABSR register is set to 0 (count stops) while counting.
  - The counter stops counting and the contents of the reload register is reloaded.
  - The TAiOUT pin outputs a low-level (“L”) signal.
  - The IR bit in the TAiIC register becomes 1 (interrupt requested) after one CPU clock cycle.

- One-shot timer is operated by an internal count source. When an external trigger is selected, a maximum of one count source clock delay occurs between the trigger input to the TAiIN pin and the one-shot timer output.

- The IR bit becomes 1 when one of the following procedures are used to set timer operating mode.
  - When selecting one-shot timer mode after reset.
  - When switching from timer mode to one-shot timer mode.
  - When switching from event counter mode to one-shot timer mode.

  To use the timer Ai interrupt (IR bit), set the IR bit to 0 after one of the above setting has done.

- When a retrigger occurs while counting, the contents of the reload register is reloaded after the counter decrements by one, and continues counting.
  To generate a retrigger while counting, wait 1 count source clock cycle or more after the last trigger generation.

- When an external trigger input is used to start counting in timer A one-shot timer mode, do not provide an external retrigger input for 300 ns before a timer A counter value reaches 0000h. The external retrigger may be ignored.
  (Technical update: TN-16C-125A/EA)

25.9.2.4 Timer A (Pulse Width Modulation Mode)

- The TAiS bit (i = 0 to 4) in the TABSR register is set to 0 (count stops) after reset. Set the TAiS bit to 1 (count starts) after selecting timer operating mode and setting the TAi register.

- The IR bit becomes 1 when one of the following procedures are used to set timer operating mode.
  - When selecting PWM mode after reset.
  - When switching from timer mode to PWM mode.
  - When switching from event counter mode to PWM mode.

  To use the timer Ai interrupt (IR bit), set the IR bit to 0 after one of the above setting has done.

- The following occurs when the TAiS bit is set to 0 (count stops) while PWM pulse is output.
  - The counter stops.
  - If the TAiOUT pin outputs a high-level (“H”) signal, the signal changes to “L” and the IR bit becomes 1.
  - If the TAiOUT pin outputs an “L” signal, its output signal and the IR bit remains unchanged.

25.9.3 Timer B

25.9.3.1 Timer B (Timer Mode, Event Counter Mode)

- The TBiS bit (i = 0 to 5) in the TABSR or TBSR register is set to 0 (count stops) after reset. Set the TBiS bit to 1 (count starts) after selecting timer operating mode and setting the TBi register.
  Bits TB2S to TB0S are bits 7 to 5 in the TABSR register. Bits TB5S to TB3S are bits 7 to 5 in the TBSR register.

- The TBi register indicates a counter value while counting at any given time. However, FFFFh can be read in the reload timing. When the TBi register is set while a counter is stopped, the setting value can be read until a counter is started.

25.9.3.2 Timer B (Pulse Period/Pulse Width Measurement Mode)

- To set the MR3 bit to 0 (no overflow has occurred), wait for one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1, while the TBiS bit is set to 1.
  (Technical update: TN-M16C-75-0110)

- Use the IR bit in the TBiIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt routine.

- When the first valid edge is input after the count starts, an undefined value is transferred to the reload register. At this time, the timer Bi interrupt request is not generated.

- The counter value is undefined when the count starts. Therefore, the MR3 bit may become 1 (overflow) and causes a timer Bi interrupt request to be generated before a valid edge is input.

- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBiMR register after the count starts. If the same value is written to bits MR1 and MR0, the IR bit is not changed.

- Pulse width is repeatedly measured in pulse width measurement mode. Determine by a program whether the measurement result is high (“H”) or low (“L”).

- If an overflow and a valid edge input occur simultaneously in pulse period measurement mode, an interrupt request is generated only once, which results in the valid edge not being recognized. Do not let an overflow occur.

- In pulse width measurement mode, determine whether an interrupt source is a valid edge input or an overflow by reading the port level in the TBi interrupt routine.

25.10 Three-Phase Motor Control Timer Function

- Do not write to the TAi or the TAi1 register (i = 1, 2, 4) in the timing that timer B2 underflows. If there is a possibility to write in this timing, read the value of the timer B2 register to verify that there is a sufficient time until timer B2 underflows, and then write to the TAi or the TAi1 register immediately.
  (Technical update: TN-M16C-86-0205)

25.11 Serial Interfaces

25.11.1 Changing UiBRG Register (i = 0 to 4)

Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When bits CLK1 and CLK0 are changed, set the UiBRG register again.

25.11.2 Clock Synchronous Mode

25.11.2.1 Selecting External Clock

If an external clock is selected, meet the following conditions while the external clock is held “H” when the CKPOL bit in the UiC0 register (i = 0 to 4) is set to 0 (transmit data output at the falling edge and receive data input at the rising edge of the serial clock), or while the external clock is held “L” when the CKPOL bit is set to 1 (transmit data output at the rising edge and receive data input at the falling edge of the serial clock)

- Set the TE bit in the UiC1 register to 1 (transmit operation enabled).
- Set the RE bit in the UiC1 register to 1 (receive operation enabled).
- The TI bit in the UiC1 register is 0 (data in the UiTB register).

The RE bit setting is not required for a transmit-only operation.

25.11.2.2 Receive Operation

- In clock synchronous mode, the serial clock is controlled by the transmit control circuit. Set the UARTi-associated registers for a transmit operation as well, even if the MCU is used only for receive operation. Dummy data is output from the TXDi pin while receiving if the TXDi pin is set to output mode.

- If data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. And the OER bit in the UiRB register becomes 1 (overrun error). In this case, a read from the UiRB register returns undefined values. If an overrun error occurs, the IR bit in the SiRIC register remains unchanged as 0.

- The following two conditions must be satisfied to use continuous receive mode (UiRRM bit is set to 1).
  (1) The CKDIR bit in the UiMR register is set to 1 (external clock).
  (2) The RTS function is not used.

  To receive data continuously under the other conditions, set the UiRRM bit to 0 (continuous receive mode disabled), and write dummy data to the UiTB register every time a receive operation is completed.

25.11.3 UART Mode

Set the UiERE bit in the UiC1 register after setting the UiMR register.

25.11.4 Special Mode 1 (I^{2}C Mode)

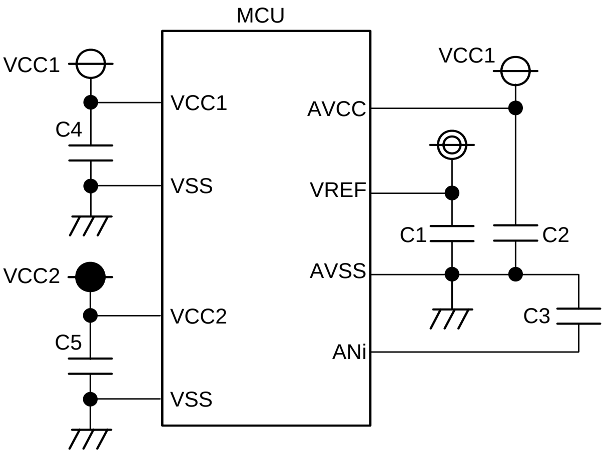
To generate the start condition, stop condition, or restart condition, set the STSPSEL bit in the USMR4 register to 0. Then, wait for a half clock cycle of the serial clock or more to change individual condition generation bit (the STAREQ bit, STPREQ bit, or RSTAREQ bit) from 0 to 1.
(Technical update: TN-16C-130A/EA)

25.12 A/D Converter

- Set the ADST bit to 1 (A/D conversion starts) after setting registers AD0CON0 (ADST bit excluded), AD0CON1, AD0CON2, AD0CON3, and AD0CON4.

- When the VCUT bit in the AD0CON1 register is changed from 0 (VREF not connected) to 1 (VREF connected), wait for 1 µs or more to start A/D conversion.
  Set the VCUT bit to 0 when A/D conversion is not used to reduce current consumption.

- To prevent latch-up and malfunction due to noise and also to minimize a conversion error, insert a capacitor between the AVSS pin and each of the following pins: the AVCC pin, VREF pin, or analog input pin ANi_j (i = none, 0, 2, 15; j = 0 to 7). Insert a capacitor between the VCC pin and the VSS pin as well. Figure 25.4 shows an example of individual pin handling.



NOTES:
1. C1 ≥ 0.47 µF, C2 ≥ 0.47 µF, C3 ≥ 10000 pF, C4 ≥ 0.1 µF, C5 ≥ 0.1 µF (reference values)
2. Use thick and shortest possible wiring to connect capacitors.

Figure 25.4 Individual Pin Handling

- Set the port direction bit in the PDk register (k = 0 to 15), which corresponds to a pin used as an analog input pin, to 0 (input mode). Also, set the port direction bit in the PDk register corresponding to the $\overline{\text{ADTRG}}$ pin, to 0 (input mode.)

- When the key input interrupt is used, do not select pins P10_4 to P10_7 (AN_4 to AN_7) as analog input pins.

- ϕAD frequency must be 16 MHz or lower when VCC1 = 4.2 V to 5.5 V, or 10 MHz or lower when VCC1 = 3.0 V to 5.5 V. When the sample and hold is not activated, ϕAD frequency must be 250 kHz or higher. When the sample and hold is activated, ϕAD frequency must be 1 MHz or higher.

- When A/D operating mode is changed, set bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register again to select analog input pins.

- The voltage applied to AN_0 to AN_7, AN15_0 to AN15_7, ANEX0, and ANEX1 must be VCC1 or below. The voltage applied to AN0_0 to AN0_7, and AN2_0 to AN2_7 must be VCC2 or below.

- If an A/D conversion in progress is forcibly aborted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion stops), the A/D conversion result will be incorrect. The AD0j (j = 0 to 7) register which is not performing A/D conversion may also be incorrect. If the ADST bit is set to 0 during A/D conversion, do not use values obtained from any of AD0j registers.

- When using DMAC operating mode in single sweep mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, or multi-port repeat sweep mode 0, do not input an external retrigger or hardware retrigger. If a retrigger is input, the sequence of A/D conversions in progress is aborted and starts over from the ANi_0 pin (i = none, 0, 2, 15). As a result, a pin and the conversion result of the pin transferred to the RAM do not correspond to each other.
  Do not read the AD00 register using instructions.

- To abort an A/D conversion in progress by setting the ADST bit in the AD0CON0 register to 0 in single sweep mode, disable interrupts before setting the ADST bit to 0.
  (Technical update: TN-16C-132A/EA)

25.13 Programmable I/O Ports

- Pins P7_2 to P7_5, P8_0, and P8_1 have the forced cutoff function of the three-phase PWM output. When these ports are set in output mode (port output, timer output, three-phase PWM output, serial interface output), they are affected by the three-phase motor control timer function and the $\overline{\text{NMI}}$ pin setting. Table 25.5 shows the INVC0 register setting, $\overline{\text{NMI}}$ pin input level, and output pin states.

Table 25.5 INVC0 Register Setting, $\overline{\text{NMI}}$ Pin Level, and Output Pin Status

Setting Value of the INVC0 Register		$\overline{\text{NMI}}$ Pin Input Level	Pin States of P7_2 to P7_5, P8_0, P8_1 (when set in output mode)
INV02 Bit	INV03 Bit		
0 (three-phase motor control timer function not used)	–	–	Output functions selected using registers PS1, PSL1, PSC, PS2, and PSL2
1 (three-phase motor control timer function used)	0 (three-phase motor control timer output disabled)	–	High-impedance states
	1 (three-phase motor control timer output enabled)(1)	H	Output functions selected using registers PS1, PSL1, PSC, PS2, and PSL2
		L (forcibly terminated)	High-impedance states

–: Not affected by the bit setting nor the pin state

NOTE:

1. The INV03 bit becomes 0 after a low-level (“L”) signal is applied to the $\overline{\text{NMI}}$ pin.

- The availability of the pull-up resistors is undefined until the internal power voltage stabilizes even if the $\overline{\text{RESET}}$ pin is held “L”.

- The input threshold level varies between the input to the port and input to the peripheral functions. If the port function and peripheral function share the same pin, the level verified by the peripheral function and the level obtained by reading the Port Pi register (i = 0 to 15) may vary during the process when the voltage applied to the pin changes from “H” to “L” or from “L” to “H”.
(Technical update: TN-M16C-102-0309)

- In the following modes, the rise time and fall time of the input/output signal to the target pins must be 10 µs or less.

< Modes >
(a) Stop mode
(b) Wait mode entered from low-power consumption mode with main voltage regulator off (the MRS bit in the VRCR register is set to 1). (note)
NOTE: It excludes when entering wait mode from other than low-power consumption mode, and when entering wait mode from low-power consumption mode with the MRS bit set to 0.

< 18 target pins >
P5_0, P5_1, P5_2, P5_3, P5_4, P5_5, P5_6, P5_7, P6_5, P6_7, P7_5,
P8_0, P8_1, P8_2, P8_3, P8_4, P9_1, P14_3

< Rise time/fall time >
Pins P6_5, P6_7, P7_5, P8_0, P8_1, P8_2, P8_3, P8_4, P9_1, P14_3: the signal transition time between 0.2VCC1 to 0.8VCC1 must be 10 µs or less.
Pins P5_0, P5_1, P5_2, P5_3, P5_4, P5_5, P5_6, P5_7: the signal transition time between 0.2VCC2 to 0.8VCC2 must be 10 µs or less.

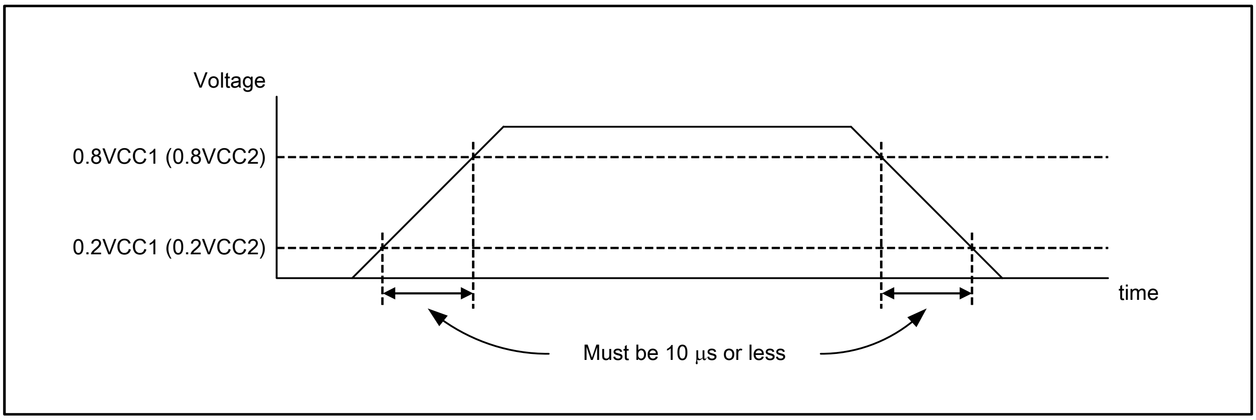


Figure 25.5 Rise Time and Fall Time

25.14 Flash Memory

25.14.1 Operating Speed

Prior to accessing registers FMR0 to FMR3 or to entering CPU rewrite mode (EW0, EW1 mode), set the CPU clock frequency to 10 MHz or lower using bits MCD4 to MCD0 in the MCD register, and also set the PM12 bit in the PM1 register to 1 (1 wait state).

25.14.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the flash memory is accessed by executing these instructions: UND, INTO, JMPS, JSRS, and BRK instructions.

25.14.3 Interrupts (EW0 Mode)

- To use peripheral function interrupts, place interrupt routine programs and the relocatable vector table in the RAM area.
- When an interrupt request is generated by the $\overline{\text{NMI}}$, watchdog timer, voltage monitor interrupt, or oscillation stop detect function, registers FMR0, FMR1, and FMR3 are forcibly initialized and the erase or program operation in progress is aborted. Now that the flash memory can be accessed, the interrupt routine will be executed.
- The address match interrupt is not available because the flash memory is accessed to process this interrupt.

25.14.4 Interrupts (EW1 Mode)

- Do not generate a peripheral function interrupt or a DMA or DMACII transfer during an erase or program operation.
- When an interrupt request is generated by the $\overline{\text{NMI}}$, watchdog timer (when the PM22 bit is set to 1), voltage monitor interrupt, or oscillation stop detect function, registers FMR0, FMR1, and FMR3 are forcibly initialized and the erase or program operation in progress is aborted. Now that the flash memory can be accessed, the interrupt routine will be executed.

25.14.5 How to Access

To set the FMR01 or FMR02 bit in the FMR0 register to 1, write a 1 immediately after writing a 0 to the bit. Write to the FMR0 register in 8-bit units. Do not generate an interrupt or a DMA or DMACII transfer between these two settings. Also, set these bits while a high-level (“H”) signal is applied to the $\overline{\text{NMI}}$ pin.
To change the FMR01 bit from 1 to 0, enter read array mode first, and then write into address 0057h in 16-bit units. Set the eight high-order bits to 00h.

25.14.6 Rewriting User ROM Area (EW0 Mode)

If the supply voltage drops while rewriting the block where a rewrite control program is stored, it may not be possible to rewrite the flash memory again, because the rewrite control program is not rewritten successfully. If this happens, use standard serial I/O mode to rewrite the block.

25.14.7 Rewriting User ROM Area (EW1 Mode)

Do not rewrite a block where the rewrite control program is stored.

25.14.8 Boot Mode

When starting up in boot mode, input pins may not be placed in high-impedance states until the internal supply voltage stabilizes. Use the following procedure to power up in boot mode.

(1) Input an “L” signal to the $\overline{\text{RESET}}$ pin and CNVSS pin
(2) Wait for *td(P-R)* (internal power supply stabilization time) or more after the voltage applied to the VCC1 pin rises above 3.0 V
(3) Input an “L” (pull-down) to the P6_5 or an “H” (pull-up) to the P6_7
(4) Input an “L” (pull-down) to the $\overline{\text{EPM}}$ (P5_5) and an “H” (pull-up) to the $\overline{\text{CE}}$ (P5_0)
(5) Input an “H” to the CNVSS pin
(6) Input an “H” to the $\overline{\text{RESET}}$ pin (out of reset)

25.14.9 Writing Command and Data

Write command codes and data to even addresses in the user ROM area.

25.14.10 Block Erase

If an erase operation in progress is aborted due to such as the $\overline{\text{NMI}}$ interrupt, hardware reset, or supply voltage drop, the lock bit or protect bit of the block which has been erased may become 0 (locked/protected). To erase the same block again, set the FMR02 bit in the FMR0 register to 1 (lock bit disabled) and then execute the block erase command.

25.14.11 Wait Mode

To enter wait mode, set the FMR01 bit in the FMR0 register to 0 (CPU rewrite mode disabled) and then execute the WAIT instruction.

25.14.12 Stop Mode

To enter stop mode, use the following procedure:

- Set the FMR01 bit to 0 (CPU rewrite mode disabled) before setting the CM10 bit to 1 (stop mode).
- Execute the JMP.B instruction right after the instruction to set the CM10 bit to 1 (stop mode).

```
e.g.,      BSET     0,        CM1           ; Stop mode
           JMP.B    L1
       L1:
           Program after exiting stop mode
```

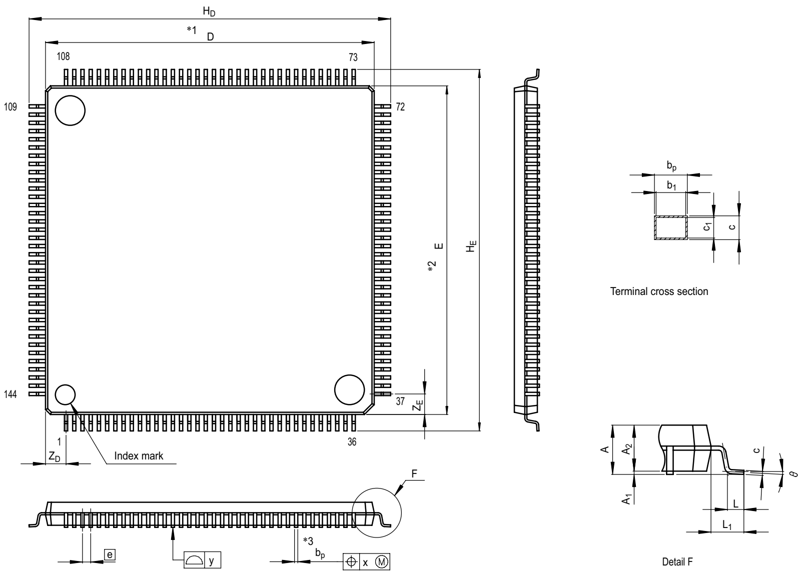
25.14.13 Low-Power Consumption Mode and On-Chip Oscillator Low-Power Consumption Mode

When the CM05 bit in the CM0 register is set to 1 (main clock stopped), do not execute the following commands:

- Program command
- Block erase command
- Lock bit program command
- Read lock bit status command
- Protect bit program command
- Read protect bit status command

Appendix 1. Package Dimensions

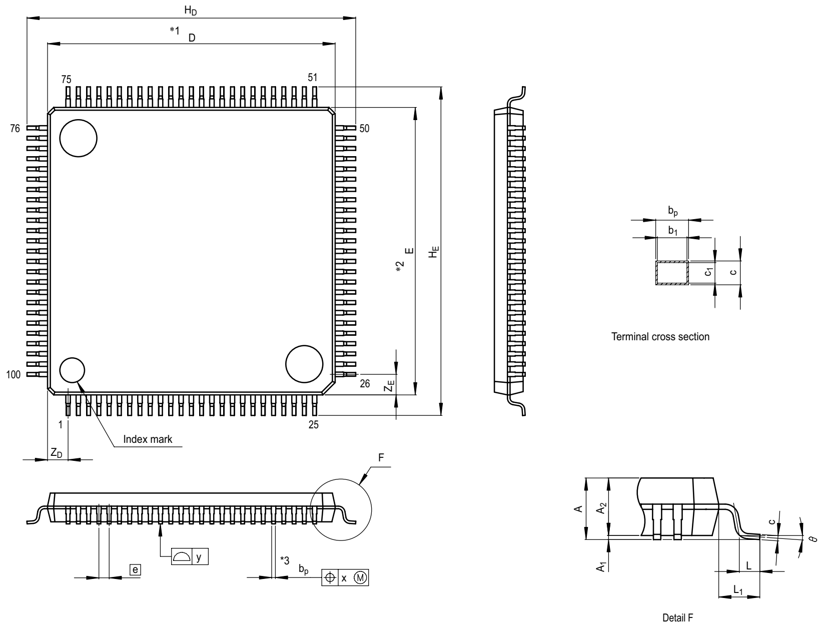
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g



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	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
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	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

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	—

Index

REVISION HISTORY	M32C/8B Group Hardware Manual

Rev.	Date	Description	
		Page	Summary
0.10	May 15, 2008	–	First Edition issued
0.50	Oct 31, 2008		**Overview**
		3, 5	• **Table 1.2, 1.4 Specifications** Current consumption added, modified
		17	• **Table 1.13 Pin Functions (3/3)** A/D converter AN0_0 to AN0_7, AN2_0 to AN2_7 supply voltage changed to VCC2
			Special Function Registers (Suffers)
		23	• **Table 4.2** Note is added to Page mode wait control register 0, 1
		27	• **Table 4.7** Address changed from “044Ch” to “034Ch”
			Reset
		35	• **Table 5.1** Note 4 changed from “P5_5($\overline{\text{EPM}}$)” to “$\overline{\text{EPM}}$(P5_5)”
			Power Supply Voltage Monitor Function
		38	• **Figure 6.3** DVCR register Note 2: Vdet(F), Vdet(R) values are added
			Clock Generation Function
		69	• **Figure 9.4** MCD Register Note 2 modified
		84	• **9.5.1.3 Low-Speed Mode** Text revised
		84	• **9.5.1.4 Low-Power Consumption Mode** Text revised
		90	• **Table 9.8** “the clock input to the CLKi pin (i = 0 to 6)” changed to “the external clock”
		93	• **Figure 9.19** Note 1 text revised
			Interrupts
		99	• **11.5.1 Fixed Vector Table** Text revised
		108	• **Figure 11.8** Diagram on the right: text changed from “Stack state before...” to “Stack state after...”
		110	• **Figure 11.10** “Interrupt request level determination output” changed to “Request signal used to wake-up from wait mode/stop mode”
			Three-Phase Motor Control Timer Function
		185	• **Figure 16.7** TB2SC register Note 1 deleted
			Serial Interfaces
		201	• **Figure 17.5** DINC bit changed from “Serial input pin..” to “Serial I/O pin..”
		202	• **Figure 17.6** UiSMR4 register Note 5 added
		218-247	• **Figure 17.11 to 17.32** Flow charts: “Initial setting start” to “Start”, “Initial setting end” to “End”
		248	• **Figure 17.33** (2) Text changed to “TXDi pin outputs “L” level since...”
			A/D Converter
		263	• **Table 18.8** Start condition “retrigger of external trigger is invalid” deleted
		267	• **Table 18.11** Note 2 revised
			• **18.2.4** Text modified, added
			Programmable I/O Ports
		281-283	• **Figures 22.1 to 22.3** Figures modified
			Flash Memory
		308	• **23.6.1** CPU Rewrite Mode text added
		311	• **Table 23.5** Read protect bit status, 2nd bus cycle changed “BA1” to “PBA”
		315	• **Figure 23.12** Note 3 and 4 added
		317	• **(9)** Text changed from “any even address” to “the protect bit address” “The flash memory enters...by reading a protect bit address.” added
			• **Figure 23.14** Text changed from “any even address” to “protect bit address”

Rev.	Date	Description	
		Page	Summary
		 326-359	**Electrical Characteristics** • Chapter added
		 379 381	**Usage Notes** • **25.12** 2nd dot, text “When using DMAC...correspond to each other.” added • **25.14.1** Text modified
0.60	Jan 31, 2009	 50 51	**Bus** • **8.2.4** “However, when the PLL clock ... the PM12 bit setting.” added • **Table 8.6** Note 1 added
		 185	**Three-Phase Motor Control Timer Function** • **Figure 16.7** TB2SC register modified
		 308 315 317	**Flash Memory** • **Table 23.4** Software command in EW1 mode modified • **23.6.1.1 (7)** “When the FMR31 bit ... in EW1 mode.” added • **23.6.1.1 (9)** “In EW1 mode, ... read protect bit status command.” added
		 329 334 337 352 353 354 355 358 359	**Electrical Characteristics** • **Table 24.5** Condition in the table title revised • **Table 24.11** Max value of “Detection voltage level accuracy” revised • **Table 24.24** RXDi input setup time for 5 V changed “30” to “80” • **Table 24.45** RXDi input setup time for 3.3 V changed “30” to “80” • **Table 24.47** Data input setup time changed to “27”, $\overline{\text{RDY}}$ input setup time to “30”, $\overline{\text{HOLD}}$ input setup time to “40” • **Table 24.48** Equation for th(WR-DB) revised • **Table 24.49** Equation for th(WR-DB) revised • **Figure 24.9** Equation for th(WR-DB) revised • **Figure 24.10** Equation for th(WR-DB) revised
		 367	**Usage Notes** • **25.5.4.3** 2nd dot of **Peripheral function clock stop** Text revised
0.61	Feb 05, 2009	 239 239	**Serial Interfaces** • **17.1.4.1** DINC bit setting changed from 1 to 0 • **17.1.4.2** DINC bit setting changed from 0 to 1
1.00	Nov 01, 2009	 -	**Throughtout the manual** • **“Preliminary”** deleted
		 6	**Overview** • **Table 1.4 Product List** “(D): Under development” deleted
		 80 81 84	**Clock** • **Figure 9.14** “Procedure to use PLL clock as CPU clock source” revised • **9.2 CPU clock and BCLK** “In the flash memory...no division mode.” added • **9.5.1.2 PLL Mode** “In the flash memory...no division mode.” added
		 332, 348 337, 352 342, 356 344, 358	**Electrical Characteristics** • **Table 24.8, 24.31** “Stop mode” and “Stop mode Topr=85°C” revised • **Table 24.23, 24.44** “tw(ADH) $\overline{\text{ADTRG}}$ input high pulse width” added • **Figure 24.3, 24.7** “tw(ADH)” added • **Figure 24.5, 24.9** $\overline{\text{RD}}$ and DBi in Read timing revised
		 365 381	**Usage Notes** • **25.5.4** “In the flash memory...no division mode.” added • **25.13** 4th dot “In the following modes...10 µs or less.” added **Figure 25.5 “Rise Time and Fall Time”** added

M32C/8B Group Hardware Manual

Publication Date : Rev.0.10 May 15, 2008
Rev.1.00 Nov 01, 2009

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

M32C/8B Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan