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

RENESAS

16/32

M32C/87 Group (M32C/87, M32C/87A, M32C/87B)

Hardware Manual

RENESAS MCU

M16C FAMILY / M32C/80 SERIES

Renesas Electronics
www.renesas.com

Rev.1.51 2008.07

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/87 Group (M32C/87, M32C/87A, M32C/87B). 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/87 Group (M32C/87, M32C/87A, M32C/87B) Datasheet	REJ03B0127-0151
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/87 Group (M32C/87, M32C/87A, M32C/87B) 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

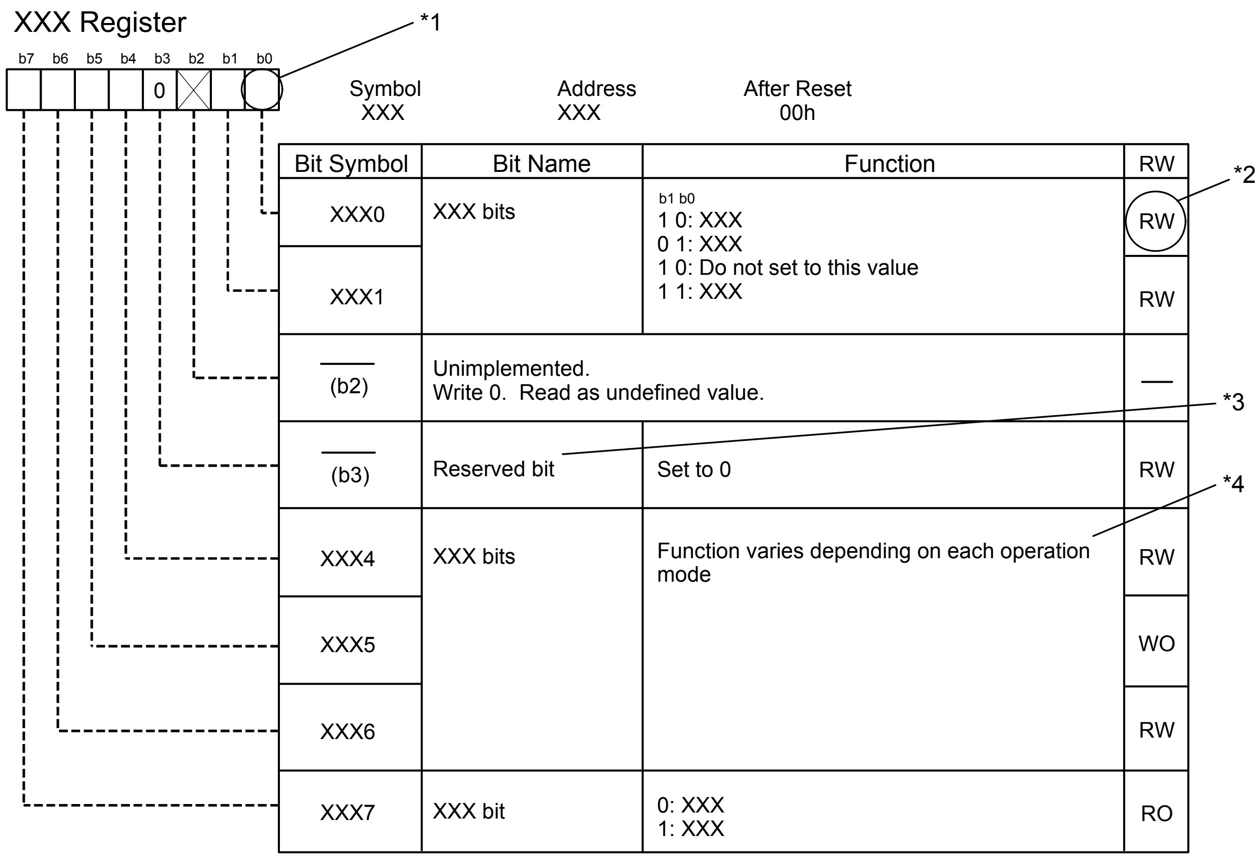
The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(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.



*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	60
0005h	Processor Mode Register 1	PM1	61
0006h	System Clock Control Register 0	CM0	82, 136
0007h	System Clock Control Register 1	CM1	83
0008h			
0009h	Address Match Interrupt Enable Register	AIER	127
000Ah	Protect Register	PRCR	105
000Bh	External Data Bus Width Control Register	DS	63
000Ch	Main Clock Division Register	MCD	84
000Dh	Oscillation Stop Detection Register	CM2	85
000Eh	Watchdog Timer Start Register	WDTS	137
000Fh	Watchdog Timer Control Register	WDC	54, 137
0010h			
0011h	Address Match Interrupt Register 0	RMAD0	127
0012h			
0013h	Processor Mode Register 2	PM2	87
0014h			
0015h	Address Match Interrupt Register 1	RMAD1	127
0016h			
0017h	Voltage Detection Register 2	VCR2	52
0018h			
0019h	Address Match Interrupt Register 2	RMAD2	127
001Ah			
001Bh	Voltage Detection Register 1	VCR1	52
001Ch			
001Dh	Address Match Interrupt Register 3	RMAD3	127
001Eh			
001Fh			
0020h			
0021h			
0022h			
0023h			
0024h			
0025h			
0026h	PLL Control Register 0	PLC0	86
0027h	PLL Control Register 1	PLC1	86
0028h			
0029h	Address Match Interrupt Register 4	RMAD4	127
002Ah			
002Bh			
002Ch			
002Dh	Address Match Interrupt Register 5	RAMD5	127
002Eh			
002Fh	Vdet4 Detection Interrupt Register	D4INT	53
0030h			
0031h			
0032h			
0033h			
0034h			
0035h			
0036h			
0037h			
0038h			
0039h	Address Match Interrupt Register 6	RMAD6	127
003Ah			
003Bh			
003Ch			
003Dh	Address Match Interrupt Register 7	RMAD7	127
003Eh			
003Fh			
0040h			
0041h			
0042h			
0043h			
0044h			
0045h			
0046h			
0047h			
0048h	External Space Wait Control Register 0	EWCR0	69
0049h	External Space Wait Control Register 1	EWCR1	69
004Ah	External Space Wait Control Register 2	EWCR2	69
004Bh	External Space Wait Control Register 3	EWCR3	69
004Ch			
004Dh			
004Eh			
004Fh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0050h			
0051h			
0052h			
0053h			
0054h			
0055h	Flash Memory Control Register 1	FMR1	500
0056h			
0057h	Flash Memory Control Register 0	FMR0	498
0058h			
0059h			
005Ah			
005Bh			
005Ch			
005Dh			
005Eh			
005Fh			
0060h			
0061h			
0062h			
0063h			
0064h			
0065h			
0066h			
0067h			
0068h	DMA0 Control Register	DM0IC	
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	114
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	II/O Interrupt Control Register 0/ CAN1 Interrupt Control Register 0	IIO0IC/ CAN3IC	
0076h	Timer B1 Interrupt Control Register	TB1IC	
0077h	II/O Interrupt Control Register 2	IIO2IC	
0078h	Timer B3 Interrupt Control Register	TB3IC	
0079h	II/O Interrupt Control Register 4	IIO4IC	
007Ah	$\overline{\text{INT5}}$ Interrupt Control Register	INT5IC	115
007Bh	II/O Interrupt Control Register 6	IIO6IC	114
007Ch	$\overline{\text{INT3}}$ Interrupt Control Register	INT3IC	115
007Dh	II/O Interrupt Control Register 8	IIO8IC	114
007Eh	$\overline{\text{INT1}}$ Interrupt Control Register	INT1IC	115
007Fh	II/O Interrupt Control Register 10/ CAN0 Interrupt Control Register 1	IIO10IC/ CAN1IC	114
0080h			
0081h	II/O Interrupt Control Register 11/ CAN0 interrupt control register 2	IIO11IC/ CAN2IC	114
0082h			
0083h			
0084h			
0085h			
0086h			
0087h			
0088h	DMA1 Interrupt Control Register	DM1IC	
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	114
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 /NACK Interrupt Control Register	S1TIC	
0093h	Key Input Interrupt Control Register	KUPIC	
0094h	Timer B0 Interrupt Control Register	TB0IC	

Blank spaces are reserved. No access is allowed.

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0095h	II/O Interrupt Control Register 1/ CAN1 Interrupt Control Register 1	IIO1IC/ CAN4IC	114
0096h	Timer B2 Interrupt Control Register	TB2IC	
0097h	II/O Interrupt Control Register 3	IIO3IC	
0098h	Timer B4 Interrupt Control Register	TB4IC	
0099h	II/O Interrupt Control Register 5/ CAN1 Interrupt Control Register 2	IIO5IC/ CAN5IC	
009Ah	$\overline{INT4}$ Interrupt Control Register	INT4IC	115
009Bh	II/O Interrupt Control Register 7	IIO7IC	114
009Ch	$\overline{INT2}$ Interrupt Control Register	INT2IC	115
009Dh	II/O Interrupt Control Register 9/ CAN0 Interrupt Control register 0	IIO9IC/ CAN0IC	114
009Eh	$\overline{INT0}$ Interrupt Control Register	INT0IC	115
009Fh	Exit Priority Register	RLVL	116, 152
00A0h	Interrupt Request Register 0	IIO0IR	129
00A1h	Interrupt Request Register 1	IIO1IR	
00A2h	Interrupt Request Register 2	IIO2IR	
00A3h	Interrupt Request Register 3	IIO3IR	
00A4h	Interrupt Request Register 4	IIO4IR	
00A5h	Interrupt Request Register 5	IIO5IR	
00A6h	Interrupt Request Register 6	IIO6IR	
00A7h	Interrupt Request Register 7	IIO7IR	
00A8h	Interrupt Request Register 8	IIO8IR	
00A9h	Interrupt Request Register 9	IIO9IR	
00AAh	Interrupt Request Register 10	IIO10IR	
00ABh	Interrupt Request Register 11	IIO11IR	
00ACh			
00ADh			
00AEh			
00AFh			
00B0h	Interrupt Enable Register 0	IIO0IE	130
00B1h	Interrupt Enable Register 1	IIO1IE	
00B2h	Interrupt Enable Register 2	IIO2IE	
00B3h	Interrupt Enable Register 3	IIO3IE	
00B4h	Interrupt Enable Register 4	IIO4IE	
00B5h	Interrupt Enable Register 5	IIO5IE	
00B6h	Interrupt Enable Register 6	IIO6IE	
00B7h	Interrupt Enable Register 7	IIO7IE	
00B8h	Interrupt Enable Register 8	IIO8IE	
00B9h	Interrupt Enable Register 9	IIO9IE	
00BAh	Interrupt Enable Register 10	IIO10IE	
00BBh	Interrupt Enable Register 11	IIO11IE	
00BCh to 00DFh			
00E0h			
00E1h			
00E2h			
00E3h			
00E4h			
00E5h			
00E6h			
00E7h			
00E8h	Group 0 SI/O Receive Buffer Register	G0RB	378
00E9h			
00EAh	Group 0 Transmit Buffer/Receive Data Register	G0TB/ G0DR	377
00EBh			
00ECh	Group 0 Receive Input Register	G0RI	378
00EDh	Group 0 SI/O Communication Mode Register	G0MR	371
00EEh	Group 0 Transmit Output Register	G0TO	378
00EFh	Group 0 SI/O Communication Control Register	G0CR	372
00F0h	Group 0 Data Compare Register 0	G0CMP0	376
00F1h	Group 0 Data Compare Register 1	G0CMP1	
00F2h	Group 0 Data Compare Register 2	G0CMP2	
00F3h	Group 0 Data Compare Register 3	G0CMP3	
00F4h	Group 0 Data Mask Register 0	G0MSK0	
00F5h	Group 0 Data Mask Register 1	G0MSK1	
00F6h	Communication Clock Select Register	CCS	370
00F7h			
00F8h	Group 0 Receive CRC Code Register	G0RCRC	376
00F9h			
00FAh	Group 0 Transmit CRC Code Register	G0TCRC	
00FBh			
00FCh	Group 0 SI/O Expansion Mode Register	G0EMR	373
00FDh	Group 0 SI/O Extended Receive Control Register	G0ERC	374
00FEh	Group 0 SI/O Special Communication Interrupt Detection Register	G0IRF	375
00FFh	Group 0 SI/O Extended Transmit Control Register	G0ETC	373

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0100h	Group 1 Time Measurement/Waveform Generation Register 0	G1TM0/ G1PO0	327/328
0101h			
0102h	Group 1 Time Measurement/Waveform Generation Register 1	G1TM1/ G1PO1	
0103h			
0104h	Group 1 Time Measurement/Waveform Generation Register 2	G1TM2/ G1PO2	
0105h			
0106h	Group 1 Time Measurement/Waveform Generation Register 3	G1TM3/ G1PO3	
0107h			
0108h	Group 1 Time Measurement/Waveform Generation Register 4	G1TM4/ G1PO4	
0109h			
010Ah	Group 1 Time Measurement/Waveform Generation Register 5	G1TM5/ G1PO5	
010Bh			
010Ch	Group 1 Time Measurement/Waveform Generation Register 6	G1TM6/ G1PO6	
010Dh			
010Eh	Group 1 Time Measurement/Waveform Generation Register 7	G1TM7/ G1PO7	
010Fh			
0110h	Group 1 Waveform Generation Control Register 0	G1POCR0	327
0111h	Group 1 Waveform Generation Control Register 1	G1POCR1	
0112h	Group 1 Waveform Generation Control Register 2	G1POCR2	
0113h	Group 1 Waveform Generation Control Register 3	G1POCR3	
0114h	Group 1 Waveform Generation Control Register 4	G1POCR4	
0115h	Group 1 Waveform Generation Control Register 5	G1POCR5	
0116h	Group 1 Waveform Generation Control Register 6	G1POCR6	
0117h	Group 1 Waveform Generation Control Register 7	G1POCR7	
0118h	Group 1 Time Measurement Control Register 0	G1TMCR0	326
0119h	Group 1 Time Measurement Control Register 1	G1TMCR1	
011Ah	Group 1 Time Measurement Control Register 2	G1TMCR2	
011Bh	Group 1 Time Measurement Control Register 3	G1TMCR3	
011Ch	Group 1 Time Measurement Control Register 4	G1TMCR4	
011Dh	Group 1 Time Measurement Control Register 5	G1TMCR5	
011Eh	Group 1 Time Measurement Control Register 6	G1TMCR6	
011Fh	Group 1 Time Measurement Control Register 7	G1TMCR7	
0120h	Group 1 Base Timer Register	G1BT	324
0121h			
0122h	Group 1 Base Timer Control Register 0	G1BCR0	324
0123h	Group 1 Base Timer Control Register 1	G1BCR1	325
0124h	Group 1 Time Measurement Prescaler Register 6	G1TPR6	326
0125h	Group 1 Time Measurement Prescaler Register 7	G1TPR7	
0126h	Group 1 Function Enable Register	G1FE	329
0127h	Group 1 Function Select Register	G1FS	
0128h	Group 1 SI/O Receive Buffer Register	G1RB	378
0129h			
012Ah	Group 1 Transmit Buffer/Receive Data Register	G1TB/ G1DR	377
012Bh			
012Ch	Group 1 Receive Input Register	G1RI	378
012Dh	Group 1 SI/O Communication Mode Register	G1MR	371
012Eh	Group 1 Transmit Output Register	G1TO	378
012Fh	Group 1 SI/O Communication Control Register	G1CR	372
0130h	Group 1 Data Compare Register 0	G1CMP0	376
0131h	Group 1 Data Compare Register 1	G1CMP1	
0132h	Group 1 Data Compare Register 2	G1CMP2	
0133h	Group 1 Data Compare Register 3	G1CMP3	
0134h	Group 1 Data Mask Register 0	G1MSK0	
0135h	Group 1 Data Mask Register 1	G1MSK1	
0136h			
0137h			
0138h	Group 1 Receive CRC Code Register	G1RCRC	376
0139h			
013Ah	Group 1 Transmit CRC Code Register	G1TCRC	
013Bh			
013Ch	Group 1 SI/O Expansion Mode Register	G1EMR	373
013Dh	Group 1 SI/O Extended Receive Control Register	G1ERC	374
013Eh	Group 1 SI/O Special Communication Interrupt Detection Register	G1IRF	375
013Fh	Group 1 SI/O Extended Transmit Control Register	G1ETC	373

Blank spaces are reserved. No access is allowed.

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0140h 0141h	Group 2 Waveform Generation Control Register 0	G2PO0	333
0142h 0143h	Group 2 Waveform Generation Control Register 1	G2PO1	
0144h 0145h	Group 2 Waveform Generation Control Register 2	G2PO2	
0146h 0147h	Group 2 Waveform Generation Control Register 3	G2PO3	
0148h 0149h	Group 2 Waveform Generation Control Register 4	G2PO4	
014Ah 014Bh	Group 2 Waveform Generation Control Register 5	G2PO5	
014Ch 014Dh	Group 2 Waveform Generation Control Register 6	G2PO6	
014Eh 014Fh	Group 2 Waveform Generation Control Register 7	G2PO7	
0150h	Group 2 Waveform Generation Control Register 0	G2POCR0	332
0151h	Group 2 Waveform Generation Control Register 1	G2POCR1	
0152h	Group 2 Waveform Generation Control Register 2	G2POCR2	
0153h	Group 2 Waveform Generation Control Register 3	G2POCR3	
0154h	Group 2 Waveform Generation Control Register 4	G2POCR4	
0155h	Group 2 Waveform Generation Control Register 5	G2POCR5	
0156h	Group 2 Waveform Generation Control Register 6	G2POCR6	
0157h	Group 2 Waveform Generation Control Register 7	G2POCR7	
0158h			
0159h			
015Ah			
015Bh			
015Ch			
015Dh			
015Eh			
015Fh			
0160h 0161h	Group 2 Base Timer Register	G2BT	330
0162h	Group 2 Base Timer Control Register 0	G2BCR0	330
0163h	Group 2 Base Timer Control Register 1	G2BCR1	331
0164h	Base Timer Start Register	BTSR	335
0165h			
0166h	Group 2 Function Enable Register	G2FE	334
0167h	Group 2 RTP Output Buffer Register	G2RTP	
0168h			
0169h			
016Ah	Group 2 SI/O Communication Mode Register	G2MR	394
016Bh	Group 2 SI/O Communication Control Register	G2CR	395
016Ch 016Dh	Group 2 SI/O Transmit Buffer Data Register	G2TB	393
016Eh 016Fh	Group 2 SI/O Receive Buffer Register	G2RB	
0170h 0171h	Group 2 IEBus Address Register	IEAR	396
0172h	Group 2 IEBus Control Register	IECR	
0173h	Group 2 IEBus Transmit Interrupt Source Detection Register	IETIF	397
0174h	Group 2 IEBus Receive Interrupt Source Detection Register	IERIF	
0175h			
0176h			
0177h	Input Function Select Register B	IPSB	486
0178h	Input Function Select Register	IPS	485
0179h	Input Function Select Register A	IPSA	486
017Ah			
017Bh			
017Ch			
017Dh to 01BFh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
01C0h	UART5 Transmit/Receive Mode Register	U5MR	274
01C1h	UART5 Baud Rate Register	U5BRG	275
01C2h 01C3h	UART5 Transmit Buffer Register	U5TB	277
01C4h	UART5 Transmit/Receive Control Register 0	U5C0	275
01C5h	UART5 Transmit/Receive Control Register 1	U5C1	276
01C6h 01C7h	UART5 Receive Buffer Register	U5RB	277
01C8h	UART6 Transmit/Receive Mode Register	U6MR	274
01C9h	UART6 Baud Rate Register	U6BRG	275
01CAh 01CBh	UART6 Transmit Buffer Register	U6TB	277
01CCh	UART6 Transmit/Receive Control Register 0	U6C0	275
01CDh	UART6 Transmit/Receive Control Register 1	U6C1	276
01CEh 01CFh	UART5 Receive Buffer Register	U6RB	277
01D0h	UART5, UART6 Transmit/Receive Control Register	U56CON	276
01D1h	UART5, UART6 Input Pin Function Select Register	U56IS	273
01D2h			
01D3h			
01D4h			
01D5h			
01D6h			
01D7h			
01D8h	RTP Output Buffer Register 0	RTP0R	459
01D9h	RTP Output Buffer Register 1	RTP1R	
01DAh	RTP Output Buffer Register 2	RTP2R	
01DBh	RTP Output Buffer Register 3	RTP3R	
01DCh			
01DDh			
01DEh			
01DFh			
01E0h	CAN0 Message Slot Buffer 0 Standard ID0	C0SLOT0_0	443
01E1h	CAN0 Message Slot Buffer 0 Standard ID1	C0SLOT0_1	
01E2h	CAN0 Message Slot Buffer 0 Extended ID0	C0SLOT0_2	444
01E3h	CAN0 Message Slot Buffer 0 Extended ID1	C0SLOT0_3	
01E4h	CAN0 Message Slot Buffer 0 Extended ID2	C0SLOT0_4	445
01E5h	CAN0 Message Slot Buffer 0 Data Length Code	C0SLOT0_5	
01E6h	CAN0 Message Slot Buffer 0 Data 0	C0SLOT0_6	446
01E7h	CAN0 Message Slot Buffer 0 Data 1	C0SLOT0_7	
01E8h	CAN0 Message Slot Buffer 0 Data 2	C0SLOT0_8	
01E9h	CAN0 Message Slot Buffer 0 Data 3	C0SLOT0_9	
01EAh	CAN0 Message Slot Buffer 0 Data 4	C0SLOT0_10	
01EBh	CAN0 Message Slot Buffer 0 Data 5	C0SLOT0_11	
01ECh	CAN0 Message Slot Buffer 0 Data 6	C0SLOT0_12	
01EDh	CAN0 Message Slot Buffer 0 Data 7	C0SLOT0_13	
01EEh	CAN0 Message Slot Buffer 0 Time Stamp High-Order	C0SLOT0_14	
01EFh	CAN0 Message Slot Buffer 0 Time Stamp Low-Order	C0SLOT0_15	
01F0h	CAN0 Message Slot Buffer 1 Standard ID0	C0SLOT1_0	443
01F1h	CAN0 Message Slot Buffer 1 Standard ID1	C0SLOT1_1	
01F2h	CAN0 Message Slot Buffer 1 Extended ID0	C0SLOT1_2	444
01F3h	CAN0 Message Slot Buffer 1 Extended ID1	C0SLOT1_3	
01F4h	CAN0 Message Slot Buffer 1 Extended ID2	C0SLOT1_4	445
01F5h	CAN0 Message Slot Buffer 1 Data Length Code	C0SLOT1_5	
01F6h	CAN0 Message Slot Buffer 1 Data 0	C0SLOT1_6	446
01F7h	CAN0 Message Slot Buffer 1 Data 1	C0SLOT1_7	
01F8h	CAN0 Message Slot Buffer 1 Data 2	C0SLOT1_8	
01F9h	CAN0 Message Slot Buffer 1 Data 3	C0SLOT1_9	
01FAh	CAN0 Message Slot Buffer 1 Data 4	C0SLOT1_10	
01FBh	CAN0 Message Slot Buffer 1 Data 5	C0SLOT1_11	
01FCh	CAN0 Message Slot Buffer 1 Data 6	C0SLOT1_12	
01FDh	CAN0 Message Slot Buffer 1 Data 7	C0SLOT1_13	
01FEh	CAN0 Message Slot Buffer 1 Time Stamp High-Order	C0SLOT1_14	
01FFh	CAN0 Message Slot Buffer 1 Time Stamp Low-Order	C0SLOT1_15	

Blank spaces are reserved. No access is allowed

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0200h	CAN0 Control Register 0	C0CTLR0	405
0201h			
0202h	CAN0 Status Register	C0STR	410
0203h			
0204h	CAN0 Extended ID Register	C0IDR	413
0205h			
0206h	CAN0 Configuration Register	C0CONR	414
0207h			
0208h	CAN0 Time Stamp Register	C0TSR	417
0209h			
020Ah	CAN0 Transmit Error Count Register	C0TEC	418
020Bh	CAN0 Receive Error Count Register	C0REC	
020Ch	CAN0 Slot Interrupt Status Register	C0SISTR	419
020Dh			
020Eh			
020Fh			
0210h	CAN0 Slot Interrupt Mask Register	C0SIMKR	421
0211h			
0212h			
0213h			
0214h	CAN0 Error Interrupt Mask Register	C0EIMKR	422
0215h	CAN0 Error Interrupt Status Register	C0EISTR	423
0216h	CAN0 Error Source Register	C0EFR	424
0217h	CAN0 Baud Rate Prescaler	C0BRP	416
0218h			
0219h	CAN0 Mode Register	C0MDR	426
021Ah			
021Bh			
021Ch			
021Dh			
021Eh			
021Fh			
0220h	CAN0 Single Shot Control Register	C0SSCTLR	428
0221h			
0222h			
0223h			
0224h	CAN0 Single Shot Status Register	C0SSSTR	430
0225h			
0226h			
0227h			
0228h	CAN0 Global Mask Register Standard ID0	C0GMR0	432
0229h	CAN0 Global Mask Register Standard ID1	C0GMR1	433
022Ah	CAN0 Global Mask Register Extended ID0	C0GMR2	434
022Bh	CAN0 Global Mask Register Extended ID1	C0GMR3	435
022Ch	CAN0 Global Mask Register Extended ID2	C0GMR4	436
022Dh			
022Eh			
022Fh			
0230h	CAN0 Message Slot 0 Control Register/ CAN0 Local Mask Register A Standard ID0	C0MCTL0/ C0LMAR0	438/432
0231h	CAN0 Message Slot 1 Control Register/ CAN0 Local Mask Register A Standard ID1	C0MCTL1/ C0LMAR1	438/433
0232h	CAN0 Message Slot 2 Control Register/ CAN0 Local Mask Register A Extended ID0	C0MCTL2/ C0LMAR2	438/434
0233h	CAN0 Message Slot 3 Control Register/ CAN0 Local Mask Register A Extended ID1	C0MCTL3/ C0LMAR3	438/435
0234h	CAN0 Message Slot 4 Control Register/ CAN0 Local Mask Register A Extended ID2	C0MCTL4/ C0LMAR4	438/436
0235h	CAN0 Message Slot 5 Control Register	C0MCTL5	438
0236h	CAN0 Message Slot 6 Control Register	C0MCTL6	
0237h	CAN0 Message Slot 7 Control Register	C0MCTL7	
0238h	CAN0 Message Slot 8 Control Register/ CAN0 Local Mask Register B Standard ID0	C0MCTL8/ C0LMBR0	438/432
0239h	CAN0 Message Slot 9 Control Register/ CAN0 Local Mask Register B Standard ID1	C0MCTL9/ C0LMBR1	438/433
023Ah	CAN0 Message Slot 10 Control Register/ CAN0 Local Mask Register B Extended ID0	C0MCTL10/ C0LMBR2	438/434
023Bh	CAN0 Message Slot 11 Control Register/ CAN0 Local Mask Register B Extended ID1	C0MCTL11/ C0LMBR3	438/435
023Ch	CAN0 Message Slot 12 Control Register/ CAN0 Local Mask Register B Extended ID2	C0MCTL12/ C0LMBR4	438/436
023Dh	CAN0 Message Slot 13 Control Register	C0MCTL13	438
023Eh	CAN0 Message Slot 14 Control Register	C0MCTL14	
023Fh	CAN0 Message Slot 15 Control Register	C0MCTL15	
0240h	CAN0 Slot Buffer Select Register	C0SBS	442
0241h	CAN0 Control Register 1	C0CTLR1	408
0242h	CAN0 Sleep Control Register	C0SLPR	409
0243h			
0244h	CAN0 Acceptance Filter Support Register	C0AFS	447
0245h			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0246h			
0247h			
0248h			
0249h			
024Ah			
024Bh			
024Ch			
024Dh			
024Eh			
024Fh			
0250h	CAN1 Slot Buffer Select Register	C1SBS	442
0251h	CAN1 Control Register 1	C1CTLR1	408
0252h	CAN1 Sleep Control Register	C1SLPR	409
0253h			
0254h	CAN1 Acceptance Filter Support Register	C1AFS	447
0255h			
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh			
025Fh			
0260h	CAN1 Message Slot Buffer 0 Standard ID0	C1SLOT0_0	443
0261h	CAN1 Message Slot Buffer 0 Standard ID1	C1SLOT0_1	
0262h	CAN1 Message Slot Buffer 0 Extended ID0	C1SLOT0_2	444
0263h	CAN1 Message Slot Buffer 0 Extended ID1	C1SLOT0_3	
0264h	CAN1 Message Slot Buffer 0 Extended ID2	C1SLOT0_4	445
0265h	CAN1 Message Slot Buffer 0 Data Length Code	C1SLOT0_5	
0266h	CAN1 Message Slot Buffer 0 Data 0	C1SLOT0_6	446
0267h	CAN1 Message Slot Buffer 0 Data 1	C1SLOT0_7	
0268h	CAN1 Message Slot Buffer 0 Data 2	C1SLOT0_8	
0269h	CAN1 Message Slot Buffer 0 Data 3	C1SLOT0_9	
026Ah	CAN1 Message Slot Buffer 0 Data 4	C1SLOT0_10	
026Bh	CAN1 Message Slot Buffer 0 Data 5	C1SLOT0_11	
026Ch	CAN1 Message Slot Buffer 0 Data 6	C1SLOT0_12	
026Dh	CAN1 Message Slot Buffer 0 Data 7	C1SLOT0_13	
026Eh	CAN1 Message Slot Buffer 0 Time Stamp High-Order	C1SLOT0_14	
026Fh	CAN1 Message Slot Buffer 0 Time Stamp Low-Order	C1SLOT0_15	
0270h	CAN1 Message Slot Buffer 1 Standard ID0	C1SLOT1_0	443
0271h	CAN1 Message Slot Buffer 1 Standard ID1	C1SLOT1_1	
0272h	CAN1 Message Slot Buffer 1 Extended ID0	C1SLOT1_2	444
0273h	CAN1 Message Slot Buffer 1 Extended ID1	C1SLOT1_3	
0274h	CAN1 Message Slot Buffer 1 Extended ID2	C1SLOT1_4	445
0275h	CAN1 Message Slot Buffer 1 Data Length Code	C1SLOT1_5	
0276h	CAN1 Message Slot Buffer 1 Data 0	C1SLOT1_6	446
0277h	CAN1 Message Slot Buffer 1 Data 1	C1SLOT1_7	
0278h	CAN1 Message Slot Buffer 1 Data 2	C1SLOT1_8	
0279h	CAN1 Message Slot Buffer 1 Data 3	C1SLOT1_9	
027Ah	CAN1 Message Slot Buffer 1 Data 4	C1SLOT1_10	
027Bh	CAN1 Message Slot Buffer 1 Data 5	C1SLOT1_11	
027Ch	CAN1 Message Slot Buffer 1 Data 6	C1SLOT1_12	
027Dh	CAN1 Message Slot Buffer 1 Data 7	C1SLOT1_13	
027Eh	CAN1 Message Slot Buffer 1 Time Stamp High-Order	C1SLOT1_14	
027Fh	CAN1 Message Slot Buffer 1 Time Stamp Low-Order	C1SLOT1_15	
0280h	CAN1 Control Register 0	C1CTLR0	405
0281h			
0282h	CAN1 Status Register	C1STR	410
0283h			
0284h	CAN1 Extended ID Register	C1IDR	413
0285h			
0286h	CAN1 Configuration Register	C1CONR	414
0287h			
0288h	CAN1 Time Stamp Register	C1TSR	417
0289h			
028Ah	CAN1 Transmit Error Count Register	C1TEC	418
028Bh	CAN1 Receive Error Count Register	C1REC	
028Ch	CAN1 Slot Interrupt Status Register	C1SISTR	419
028Dh			
028Eh			
028Fh			

Blank spaces are reserved. No access is allowed.

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0290h 0291h	CAN1 Slot Interrupt Mask Register	C1SIMKR	421
0292h			
0293h			
0294h	CAN1 Error Interrupt Mask Register	C1EIMKR	422
0295h	CAN1 Error Interrupt Status Register	C1EISTR	423
0296h	CAN1 Error Source Register	C1EFR	424
0297h	CAN1 Baud Rate Prescaler	C1BRP	416
0298h			
0299h	CAN1 Mode Register	C1MDR	426
029Ah			
029Bh			
029Ch			
029Dh			
029Eh			
029Fh			
02A0h 02A1h	CAN1 Single Shot Control Register	C1SSCTLR	428
02A2h			
02A3h			
02A4h 02A5h	CAN1 Single Shot Status Register	C1SSSTR	430
02A6h			
02A7h			
02A8h	CAN1 Global Mask Register Standard ID0	C1GMR0	432
02A9h	CAN1 Global Mask Register Standard ID1	C1GMR1	433
02AAh	CAN1 Global Mask Register Extended ID0	C1GMR2	434
02ABh	CAN1 Global Mask Register Extended ID1	C1GMR3	435
02ACh	CAN1 Global Mask Register Extended ID2	C1GMR4	436
02ADh			
02AEh			
02AFh			
02B0h	CAN1 Message Slot 0 Control Register/ CAN1 Local Mask Register A Standard ID0	C1MCTL0/ C1LMAR0	438/432
02B1h	CAN1 Message Slot 1 Control Register/ CAN1 Local Mask Register A Standard ID1	C1MCTL1/ C1LMAR1	438/433
02B2h	CAN1 Message Slot 2 Control Register/ CAN1 Local Mask Register A Extended ID0	C1MCTL2/ C1LMAR2	438/434
02B3h	CAN1 Message Slot 3 Control Register/ CAN1 Local Mask Register A Extended ID1	C1MCTL3/ C1LMAR3	438/435
02B4h	CAN1 Message Slot 4 Control Register/ CAN1 Local Mask Register A Extended ID2	C1MCTL4/ C1LMAR4	438/436
02B5h	CAN1 Message Slot 5 Control Register	C1MCTL5	438
02B6h	CAN1 Message Slot 6 Control Register	C1MCTL6	
02B7h	CAN1 Message Slot 7 Control Register	C1MCTL7	
02B8h	CAN1 Message Slot 8 Control Register/ CAN1 Local Mask Register B Standard ID0	C1MCTL8/ C1LMBR0	438/432
02B9h	CAN1 Message Slot 9 Control Register/ CAN1 Local Mask Register B Standard ID1	C1MCTL9/ C1LMBR1	438/433
02BAh	CAN1 Message Slot 10 Control Register/ CAN1 Local Mask Register B Extended ID0	C1MCTL10/ C1LMBR2	438/434
02BBh	CAN1 Message Slot 11 Control Register/ CAN1 Local Mask Register B Extended ID1	C1MCTL11/ C1LMBR3	438/435
02BCh	CAN1 Message Slot 11 Control Register/ CAN1 Local Mask Register B Extended ID1	C1MCTL12/ C1LMBR4	438/436
02BDh	CAN1 Message Slot 13 Control Register	C1MCTL13	438
02BEh	CAN1 Message Slot 14 Control Register	C1MCTL14	
02BFh	CAN1 Message Slot 15 Control Register	C1MCTL15	
02C0h 02C1h	X0 Register, Y0 Register	X0R, Y0R	318
02C2h 02C3h	X1 Register, Y1 Register	X1R, Y1R	
02C4h 02C5h	X2 Register, Y2 Register	X2R, Y2R	
02C6h 02C7h	X3 Register, Y3 Register	X3R, Y3R	
02C8h 02C9h	X4 Register, Y4 Register	X4R, Y4R	
02CAh 02CBh	X5 Register, Y5 Register	X5R, Y5R	
02CCh 02CDh	X6 Register, Y6 Register	X6R, Y6R	
02CEh 02CFh	X7 Register, Y7 Register	X7R, Y7R	
02D0h 02D1h	X8 Register, Y8 Register	X8R, Y8R	
02D2h 02D3h	X9 Register, Y9 Register	X9R, Y9R	

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
02D4h 02D5h	X10 Register, Y10 Register	X10R, Y10R	318
02D6h 02D7h	X11 Register, Y11 Register	X11R, Y11R	
02D8h 02D9h	X12 Register, Y12 Register	X12R, Y12R	
02DAh 02DBh	X13 Register, Y13 Register	X13R, Y13R	
02DCh 02DDh	X14 Register, Y14 Register	X14R, Y14R	
02DEh 02DFh	X15 Register, Y15 Register	X15R, Y15R	
02E0h	X/Y Control Register	XYC	318
02E1h			
02E2h			
02E3h			
02E4h	UART1 Special Mode Register 4	U1SMR4	220
02E5h	UART1 Special Mode Register 3	U1SMR3	219
02E6h	UART1 Special Mode Register 2	U1SMR2	218
02E7h	UART1 Special Mode Register	U1SMR	217
02E8h	UART1 Transmit/Receive Mode Register	U1MR	216
02E9h	UART1 Baud Rate Register	U1BRG	222
02EAh 02EBh	UART1 Transmit Buffer Register	U1TB	224
02ECh	UART1 Transmit/Receive Control Register 0	U1C0	221
02EDh	UART1 Transmit/Receive Control Register 1	U1C1	222
02EEh 02EFh	UART1 Receive Buffer Register	U1RB	224
02F0h			
02F1h			
02F2h			
02F3h			
02F4h	UART4 Special Mode Register 4	U4SMR4	220
02F5h	UART4 Special Mode Register 3	U4SMR3	219
02F6h	UART4 Special Mode Register 2	U4SMR2	218
02F7h	UART4 Special Mode Register	U4SMR	217
02F8h	UART4 Transmit/Receive Mode Register	U4MR	216
02F9h	UART4 Baud Rate Register	U4BRG	222
02FAh 02FBh	UART4 Transmit Buffer Register	U4TB	224
02FCh	UART4 Transmit/Receive Control Register 0	U4C0	221
02FDh	UART4 Transmit/Receive Control Register 1	U4C1	222
02FEh 02FFh	UART4 Receive Buffer Register	U4RB	224
0300h	Timer B3, B4, B5 Count Start Flag	TBSR	189
0301h			
0302h 0303h	Timer A11 Register	TA11	205
0304h 0305h	Timer A21 Register	TA21	
0306h 0307h	Timer A41 Register	TA41	
0308h	Three-Phase PWM Control Register 0	INVC0	198
0309h	Three-Phase PWM Control Register 1	INVC1	199
030Ah	Three-Phase Output Buffer Register 0	IDB0	205
030Bh	Three-Phase Output Buffer Register 1	IDB1	205
030Ch	Dead Time Timer	DTT	204
030Dh	Timer B2 Interrupt Generation Frequency Set Counter	ICTB2	203
030Eh			
030Fh			
0310h 0311h	Timer B3 Register	TB3	188
0312h 0313h	Timer B4 Register	TB4	
0314h 0315h	Timer B5 Register	TB5	
0316h			
0317h			
0318h			
0319h			
031Ah			
031Bh	Timer B3 Mode Register	TB3MR	185, 186, 187
031Ch	Timer B4 Mode Register	TB4MR	
031Dh	Timer B5 Mode Register	TB5MR	
031Eh	External Interrupt Source Select Register 1	IFSRA	125
031Fh	External Interrupt Source Select Register	IFSR	124, 223

Blank spaces are reserved. No access is allowed.

Special Function Register (SFR) Page Reference

Address	Register	Symbol	Page
0320h			
0321h			
0322h			
0323h			
0324h	UART3 Special Mode Register 4	U3SMR4	220
0325h	UART3 Special Mode Register 3	U3SMR3	219
0326h	UART3 Special Mode Register 2	U3SMR2	218
0327h	UART3 Special Mode Register	U3SMR	217
0328h	UART3 Transmit/Receive Mode Register	U3MR	216
0329h	UART3 Baud Rate Register	U3BRG	222
032Ah	UART3 Transmit Buffer Register	U3TB	224
032Bh			
032Ch	UART3 Transmit/Receive Control Register 0	U3C0	221
032Dh	UART3 Transmit/Receive Control Register 1	U3C1	222
032Eh	UART3 Receive Buffer Register	U3RB	224
032Fh			
0330h			
0331h			
0332h			
0333h			
0334h	UART2 Special Mode Register 4	U2SMR4	220
0335h	UART2 Special Mode Register 3	U2SMR3	219
0336h	UART2 Special Mode Register 2	U2SMR2	218
0337h	UART2 Special Mode Register	U2SMR	217
0338h	UART2 Transmit/Receive Mode Register	U2MR	216
0339h	UART2 Baud Rate Register	U2BRG	222
033Ah	UART2 Transmit Buffer Register	U2TB	224
033Bh			
033Ch	UART2 Transmit/Receive Control Register 0	U2C0	221
033Dh	UART2 Transmit/Receive Control Register 1	U2C1	222
033Eh	UART2 Receive Buffer Register	U2RB	224
033Fh			
0340h	Count Start Register	TABSR	170, 189, 206
0341h	Clock Prescaler Reset Register	CPSRF	88
0342h	One-Shot Start Register	ONSF	171
0343h	Trigger Select Register	TRGSR	169, 202
0344h	Up/Down Select Register	UDF	168
0345h			
0346h	Timer A0 Register	TA0	167
0347h			
0348h	Timer A1 Register	TA1	167, 205
0349h			
034Ah	Timer A2 Register	TA2	167, 205
034Bh			
044Ch	Timer A3 Register	TA3	167
034Dh			
034Eh	Timer A4 Register	TA4	167, 205
034Fh			
0350h	Timer B0 Register	TB0	188
0351h			
0352h	Timer B1 Register	TB1	188
0353h			
0354h	Timer B2 Register	TB2	188, 204
0355h			
0356h	Timer A0 Mode Register	TA0MR	163, 164, 165, 166
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	185, 186, 187
035Ch	Timer B1 Mode Register	TB1MR	
035Dh	Timer B2 Mode Register	TB2MR	
035Eh	Timer B2 Special Mode Register	TB2SC	203
035Fh	Count Source Prescaler Register	TCSPR	88, 162
0360h			
0361h			
0362h			
0363h			
0364h	UART0 Special Mode Register 4	U0SMR4	220
0365h	UART0 Special Mode Register 3	U0SMR3	219
0366h	UART0 Special Mode Register 2	U0SMR2	218
0367h	UART0 Special Mode Register	U0SMR	217
0368h	UART0 Transmit/Receive Mode Register	U0MR	216
0369h	UART0 Baud Rate Register	U0BRG	222
036Ah	UART0 Transmit Buffer Register	U0TB	224
036Bh			
036Ch	UART0 Transmit/Receive Control Register 0	U0C0	221
036Dh	UART0 Transmit/Receive Control Register 1	U0C1	222
036Eh	UART0 Receive Buffer Register	U0RB	224
036Fh			

Blank spaces are reserved. No access is allowed.

Address	Register	Symbol	Page
0370h			
0371h			
0372h	IrDA Control Register	IRCON	270
0373h			
0374h			
0375h			
0376h			
0377h			
0378h	DMA0 Request Source Select Register	DM0SL	140
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	316
037Dh			
037Eh	CRC Input Register	CRCIN	316
037Fh			
0380h	A/D0 Register 0	AD00	299
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	299
0393h			
0394h	A/D0 Control Register 2	AD0CON2	297
0395h	A/D0 Control Register 3	AD0CON3	298
0396h	A/D0 Control Register 0	AD0CON0	295
0397h	A/D0 Control Register 1	AD0CON1	296
0398h	D/A Register 0	DA0	314
0399h			
039Ah	D/A Register 1	DA1	314
039Bh			
039Ch	D/A Control Register	DACON	314
039Dh	D/A Control Register 1	DACON1	314
039Eh			
039Fh			
03A0h	Function Select Register A8	PS8	472
03A1h	Function Select Register A9	PS9	
03A2h			
03A3h	Function Select Register B9	PSL9	476
03A4h	Function Select Register E2	PSE2	480
03A5h			
03A6h			
03A7h	Function Select Register D1	PSD1	479
03A8h	Function Select Register D2	PSD2	
03A9h			
03AAh	Function Select Register C6	PSC6	478
03ABh	Function Select Register E1	PSE1	480
03ACh	Function Select Register C2	PSC2	477
03ADh	Function Select Register C3	PSC3	478
03AEh			
03AFh	Function Select Register C	PSC	477
03B0h	Function Select Register A0	PS0	468
03B1h	Function Select Register A1	PS1	
03B2h	Function Select Register B0	PSL0	473
03B3h	Function Select Register B1	PSL1	
03B4h	Function Select Register A2	PS2	469
03B5h	Function Select Register A3	PS3	
03B6h	Function Select Register B2	PSL2	474
03B7h	Function Select Register B3	PSL3	
03B8h	Function Select Register A4	PS4	470
03B9h	Function Select Register A5	PS5	
03BAh			
03BBh	Function Select Register B5	PSL5	475
03BCh	Function Select Register A6	PS6	471
03BDh	Function Select Register A7	PS7	471
03BEh	Function Select Register B6	PSL6	475
03BFh	Function Select Register B7	PSL7	476

Blank spaces are reserved. No access is allowed

Special Function Register (SFR) Page Reference

..

Address	Register	Symbol	Page
03C0h	Port P6 Register	P6	467
03C1h	Port P7 Register	P7	
03C2h	Port P6 Direction Register	PD6	466
03C3h	Port P7 Direction Register	PD7	
03C4h	Port P8 Register	P8	467
03C5h	Port P9 Register	P9	
03C6h	Port P8 Direction Register	PD8	466
03C7h	Port P9 Direction Register	PD9	
03C8h	Port P10 Register	P10	467
03C9h	Port P11 Register	P11	
03CAh	Port P10 Direction Register	PD10	466
03CBh	Port P11 Direction Register	PD11	
03CCh	Port P12 Register	P12	467
03CDh	Port P13 Register	P13	
03CEh	Port P12 Direction Register	PD12	466
03CFh	Port P13 Direction Register	PD13	
03D0h	Port P14 Register	P14	467
03D1h	Port P15 Register	P15	
03D2h	Port P14 Direction Register	PD14	466
03D3h	Port P15 Direction Register	PD15	
03D4h			
03D5h			
03D6h			
03D7h			
03D8h			
03D9h			
03DAh	Pull-Up Control Register 2	PUR2	482
03DBh	Pull-Up Control Register 3	PUR3	483
03DCh	Pull-Up Control Register 4	PUR4	484
03DDh			
03DEh			
03DFh			
03E0h	Port P0 Register	P0	467
03E1h	Port P1 Register	P1	
03E2h	Port P0 Direction Register	PD0	466
03E3h	Port P1 Direction Register	PD1	
03E4h	Port P2 Register	P2	467
03E5h	Port P3 Register	P3	
03E6h	Port P2 Direction Register	PD2	466
03E7h	Port P3 Direction Register	PD3	
03E8h	Port P4 Register	P4	467
03E9h	Port P5 Register	P5	
03EAh	Port P4 Direction Register	PD4	466
03EBh	Port P5 Direction Register	PD5	
03ECh			
03EDh			
03EEh			
03EFh			
03F0h	Pull-Up Control Register 0	PUR0	481
03F1h	Pull-Up Control Register 1	PUR1	
03F2h			
03F3h			
03F4h			
03F5h			
03F6h			
03F7h			
03F8h			
03F9h			
03FAh			
03FBh			
03FCh			
03FDh			
03FEh			
03FFh	Port Control Register	PCR	485

Blank spaces are reserved. No access is allowed.

1. Overview

1.1 Features

The M32C/87 Group (M32C/87, M32C/87A, M32C/87B) is a single-chip control MCU, fabricated using high-performance silicon gate CMOS technology, embedding the M32C/80 Series CPU core. The M32C/87 Group (M32C/87, M32C/87A, M32C/87B) is housed in 144-pin and 100-pin plastic molded LQFP/QFP 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/87 Group (M32C/87, M32C/87A, M32C/87B) 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 components, cameras, office equipment, communication devices, mobile devices, etc.

1.1.2 Specifications

Tables 1.1 to 1.4 list the specifications of the M32C/87 Group (M32C/87, M32C/87A, M32C/87B).

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 = 4.2 to 5.5 V) 41.7 ns (f(CPU) = 24 MHz, VCC1 = 3.0 to 5.5 V) • Operating modes: Single-chip mode, memory expansion mode, and microprocessor mode
Memory	ROM, RAM, data flash	See **Tables 1.5 to 1.7 Product List**.
Power Supply Voltage Detection		Vdet3 detection function, Vdet4 detection function, cold start/warm start determination function
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 detection 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: 14 ($\overline{NMI}$, $\overline{INT}$ × 9, key input × 4) • Interrupt priority levels: 7
Watchdog Timer		15-bit × 1 channel (with prescaler)
DMA	DMAC	• 4 channels, cycle steal method • Trigger sources: 43 • 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

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, IrDA mode[2], IEBus (optional)[1][3]
	UART5, UART6	Clock synchronous/asynchronous × 2
A/D Converter		10-bit resolution × 34 channels (in single-chip mode) 10-bit resolution × 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
Intelligent I/O		16-bit timer × 2 • Time measurement function (input capture): 8 channels • Waveform generation function (output compare): 16 channels • Communication function: Clock synchronous mode, clock asynchronous mode, HDLC data processing mode, IEBus (optional)[1][3] • 2-phase pulse signal processing (2-phase encoder input) × 1
ROM Correction Function		Address match interrupt × 8
CAN modules		Supporting CAN 2.0B specification M32C/87: 16 slots × 2 channels, M32C/87A: 16 slots × 1 channel M32C/87B: none
I/O Ports	Programmable I/O ports	• Input only: 1 • CMOS I/O: 121 with selectable pull-up resistor • N channel open drain ports: 2
Flash Memory		• Erase and program voltage: 3.3 V ± 0.3 V or 5.0 V ± 0.5 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 = 4.2 to 5.5 V, VCC2 = 3.0 V to VCC1 24 MHz: VCC1 = 3.0 to 5.5 V, VCC2 = 3.0 V to VCC1
Current Consumption		32 mA (32 MHz, VCC1 = VCC2 = 5 V) 23 mA (24 MHz, VCC1 = VCC2 = 3.3 V) 45 μA (approx. 1 MHz, VCC1 = VCC2 = 3.3 V, on-chip oscillator low-power consumption mode → wait mode) 0.8 μA (VCC1 = VCC2 = 3.3 V, stop mode)
Operating Ambient Temperature (°C)		-20 to 85°C, -40 to 85°C (optional)[3]
Package		144-pin LQFP (PLQP0144KA-A)

NOTES:
1. IEBus is a registered trademark of NEC Electronics Corporation.
2. Available in UART0.
3. 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 = 4.2 to 5.5 V) 41.7 ns (f(CPU) = 24 MHz, VCC1 = 3.0 to 5.5 V) • Operating mode: Single-chip mode, memory expansion mode, and microprocessor mode
Memory	ROM, RAM, data flash	See **Tables 1.5 to 1.7 Product List**.
Power Supply Voltage Detection		Vdet3 detection function, Vdet4 detection function, cold start/warm start determination function
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 detection 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 ($\overline{NMI}$, $\overline{INT}$ × 6, key input × 4) • Interrupt priority levels: 7
Watchdog Timer		15-bit × 1 channel (with prescaler)
DMA	DMAC	• 4 channels, cycle steal method • Trigger sources: 43 • 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

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, IrDA mode(2), IEBus (optional)(1)(3)
	UART5	Clock synchronous/asynchronous × 1
A/D Converter		10-bit resolution × 26 channels (in single-chip mode) 10-bit resolution × 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
Intelligent I/O		16-bit timer × 2 • Time measurement function (input capture): 8 channels • Waveform generation function (output compare): 10 channels • Communication function: Clock synchronous mode, clock asynchronous mode, HDLC data processing mode, IEBus (optional)(1)(3) • 2-phase pulse signal processing (2-phase encoder input) × 1
ROM Correction Function		Address match interrupt × 8
CAN modules		Supporting CAN 2.0B specification M32C/87: 16 slots × 2 channels, M32C/87A: 16 slots × 1 channel M32C/87B: none
I/O Ports	Programmable I/O ports	• Input only: 1 • CMOS I/O: 85, selectable pull-up resistor • N channel open drain ports: 2
Flash Memory		• Erase and program voltage: 3.3 V ± 0.3 V or 5.0 V ± 0.5 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 = 4.2 to 5.5 V, VCC2 = 3.0 V to VCC1 24 MHz: VCC1 = 3.0 to 5.5 V, VCC2 = 3.0 V to VCC1
Current Consumption		32 mA (32 MHz, VCC1 = VCC2 = 5 V) 23 mA (24 MHz, VCC1 = VCC2 = 3.3 V) 45 μA (approx. 1 MHz, VCC1 = VCC2 = 3.3 V, on-chip oscillator low-power consumption mode → wait mode) 0.8 μA (VCC1 = VCC2 = 3.3 V, stop mode)
Operating Ambient Temperature (°C)		-20 to 85°C, -40 to 85°C (optional)(3)
Package		100-pin LQFP (PLQP0100KB-A) 100-pin QFP (PRQP0100JB-A)

NOTES:
1. IEBus is a registered trademark of NEC Electronics Corporation.
2. Available in UART0.
3. Please contact a Renesas sales office for optional features.

1.2 Product List

Tables 1.5 to 1.7 list product information. Figure 1.1 shows product numbering system.

Table 1.5 M32C/87 Group (1) (M32C/87: 2-channel CAN module) Current as of Jul. 2008

Part Number	Package Code	ROM Capacity	RAM Capacity	Remarks
M3087BFLGP	PLQP0144KA-A (144P6Q-A)	1 MB + 4 KB(1)	48 KB	Flash memory
M30879FLFP	PRQP0100JB-A (100P6S-A)			
M30879FLGP	PLQP0100KB-A (100P6Q-A)			
M3087BFKGP	PLQP0144KA-A (144P6Q-A)	768 KB + 4 KB(1)		
M30879FKGP	PLQP0100KB-A (100P6Q-A)			
M30878FJGP	PLQP0144KA-A (144P6Q-A)	512 KB + 4 KB(1)	31 KB	
M30876FJGP	PLQP0100KB-A (100P6Q-A)			
M30875FHGP	PLQP0144KA-A (144P6Q-A)	384 KB + 4 KB(1)	24 KB	
M30873FHGP	PLQP0100KB-A (100P6Q-A)			
M30878MJ-XXXGP	PLQP0144KA-A (144P6Q-A)	512 KB	31 KB	Mask ROM
M30876MJ-XXXFP	PRQP0100JB-A (100P6S-A)			
M30876MJ-XXXGP	PLQP0100KB-A (100P6Q-A)			
M30875MH-XXXGP	PLQP0144KA-A (144P6Q-A)	384 KB	24 KB	
M30873MH-XXXGP	PLQP0100KB-A (100P6Q-A)			

NOTE:
1. Additional 4-Kbyte space is available for data flash memory.

Table 1.6 M32C/87 Group (2) (M32C/87A: 1-channel CAN module) Current as of Jul. 2008

Part Number	Package Code	ROM Capacity	RAM Capacity	Remarks
M3087BFLAGP	PLQP0144KA-A (144P6Q-A)	1 MB + 4 KB(1)	48 KB	Flash memory
M30879FLAFP	PRQP0100JB-A (100P6S-A)			
M30879FLAGP	PLQP0100KB-A (100P6Q-A)			
M3087BFKAGP	PLQP0144KA-A (144P6Q-A)	768 KB + 4 KB(1)		
M30879FKAGP	PLQP0100KB-A (100P6Q-A)			
M30878FJAGP	PLQP0144KA-A (144P6Q-A)	512 KB + 4 KB(1)	31 KB	
M30876FJAGP	PLQP0100KB-A (100P6Q-A)			
M30875FHAGP	PLQP0144KA-A (144P6Q-A)	384 KB + 4 KB(1)	24 KB	
M30873FHAGP	PLQP0100KB-A (100P6Q-A)			
M30878MJA-XXXGP	PLQP0144KA-A (144P6Q-A)	512 KB	31 KB	Mask ROM
M30876MJA-XXXFP	PRQP0100JB-A (100P6S-A)			
M30876MJA-XXXGP	PLQP0100KB-A (100P6Q-A)			
M30875MHA-XXXGP	PLQP0144KA-A (144P6Q-A)	384 KB	24 KB	
M30873MHA-XXXGP	PLQP0100KB-A (100P6Q-A)			

NOTE:
1. Additional 4-Kbyte space is available for data flash memory.

Table 1.7 M32C/87 Group (3) (M32C/87B: no CAN module) Current as of Jul. 2008

Part Number	Package Code	ROM Capacity	RAM Capacity	Remarks
M3087BFLBGP	PLQP0144KA-A (144P6Q-A)	1 MB + 4 KB(1)	48 KB	Flash memory
M30879FLBFP	PRQP0100JB-A (100P6S-A)			
M30879FLBGP	PLQP0100KB-A (100P6Q-A)			
M3087BFKBGP	PLQP0144KA-A (144P6Q-A)	768 KB + 4 KB(1)		
M30879FKBGP	PLQP0100KB-A (100P6Q-A)			
M30878FJBGP	PLQP0144KA-A (144P6Q-A)	512 KB + 4 KB(1)	31 KB	
M30876FJBGP	PLQP0100KB-A (100P6Q-A)			
M30875FHBGP	PLQP0144KA-A (144P6Q-A)	384 KB + 4 KB(1)	24 KB	
M30873FHBGP	PLQP0100KB-A (100P6Q-A)			
M30878MJB-XXXGP	PLQP0144KA-A (144P6Q-A)	512 KB	31 KB	Mask ROM
M30876MJB-XXXFP	PRQP0100JB-A (100P6S-A)			
M30876MJB-XXXGP	PLQP0100KB-A (100P6Q-A)			
M30875MHB-XXXGP	PLQP0144KA-A (144P6Q-A)	384 KB	24 KB	
M30873MHB-XXXGP	PLQP0100KB-A (100P6Q-A)			

NOTE:
1. Additional 4-Kbyte space is available for data flash memory.

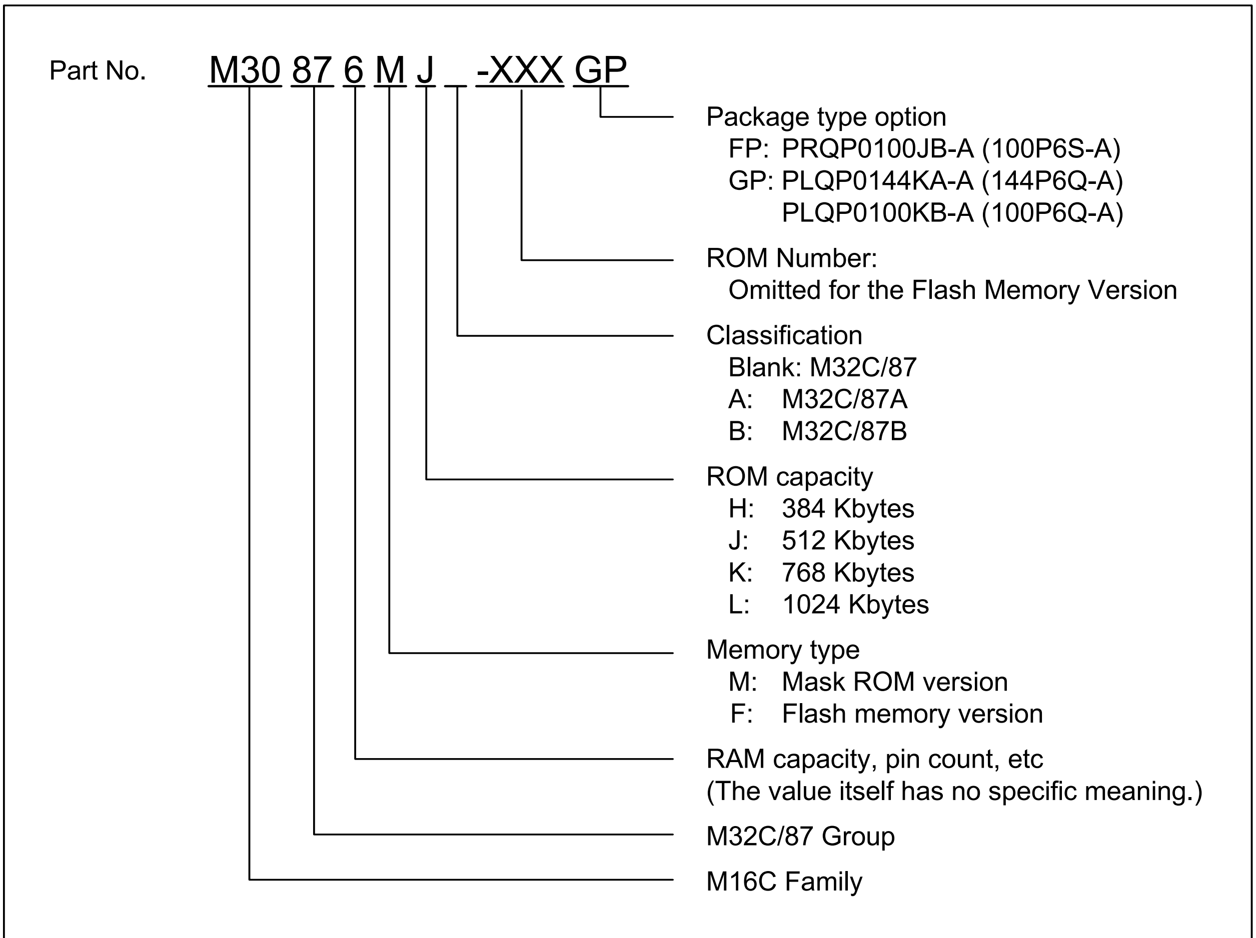
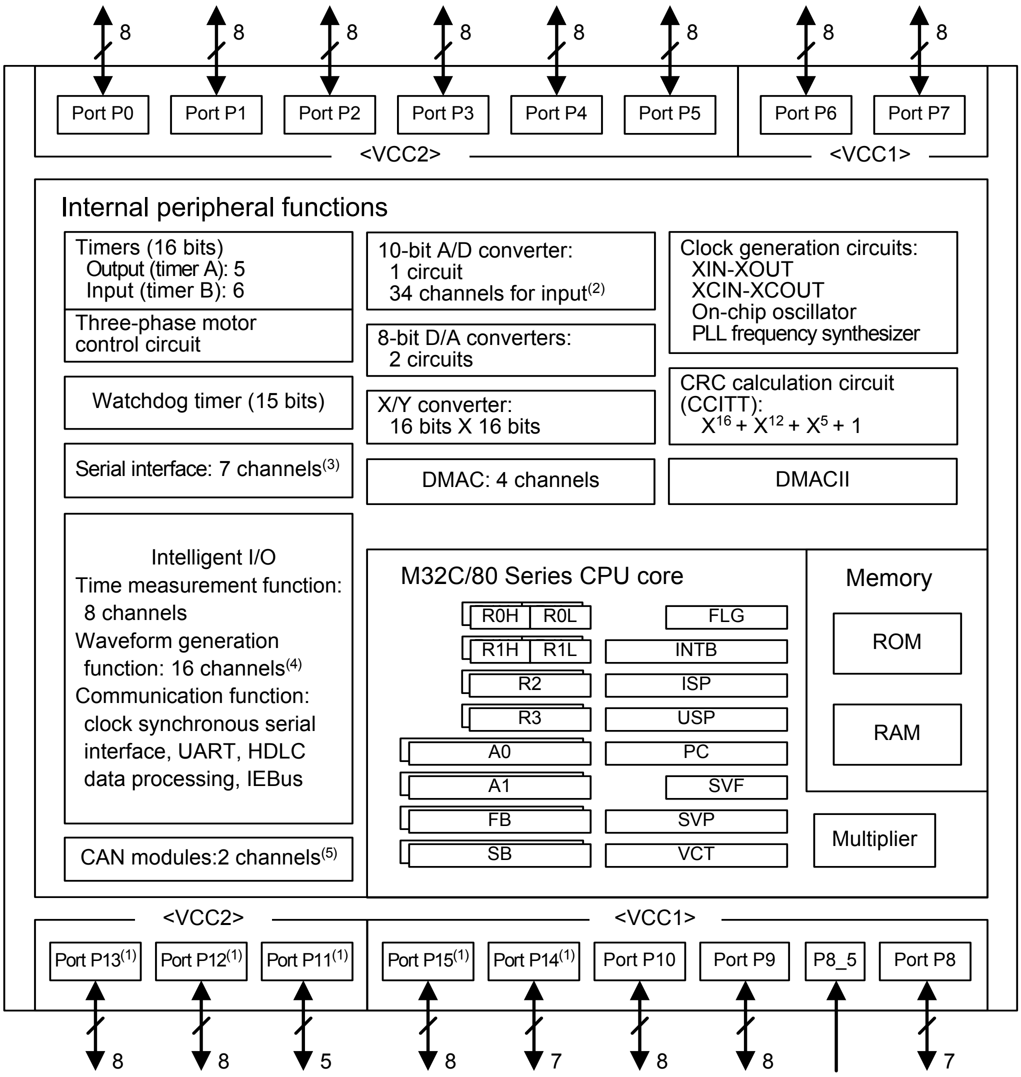


Figure 1.1 Product Numbering System

1.3 Block Diagram

Figure 1.2 shows a block diagram of the M32C/87 Group (M32C/87, M32C/87A, M32C/87B).

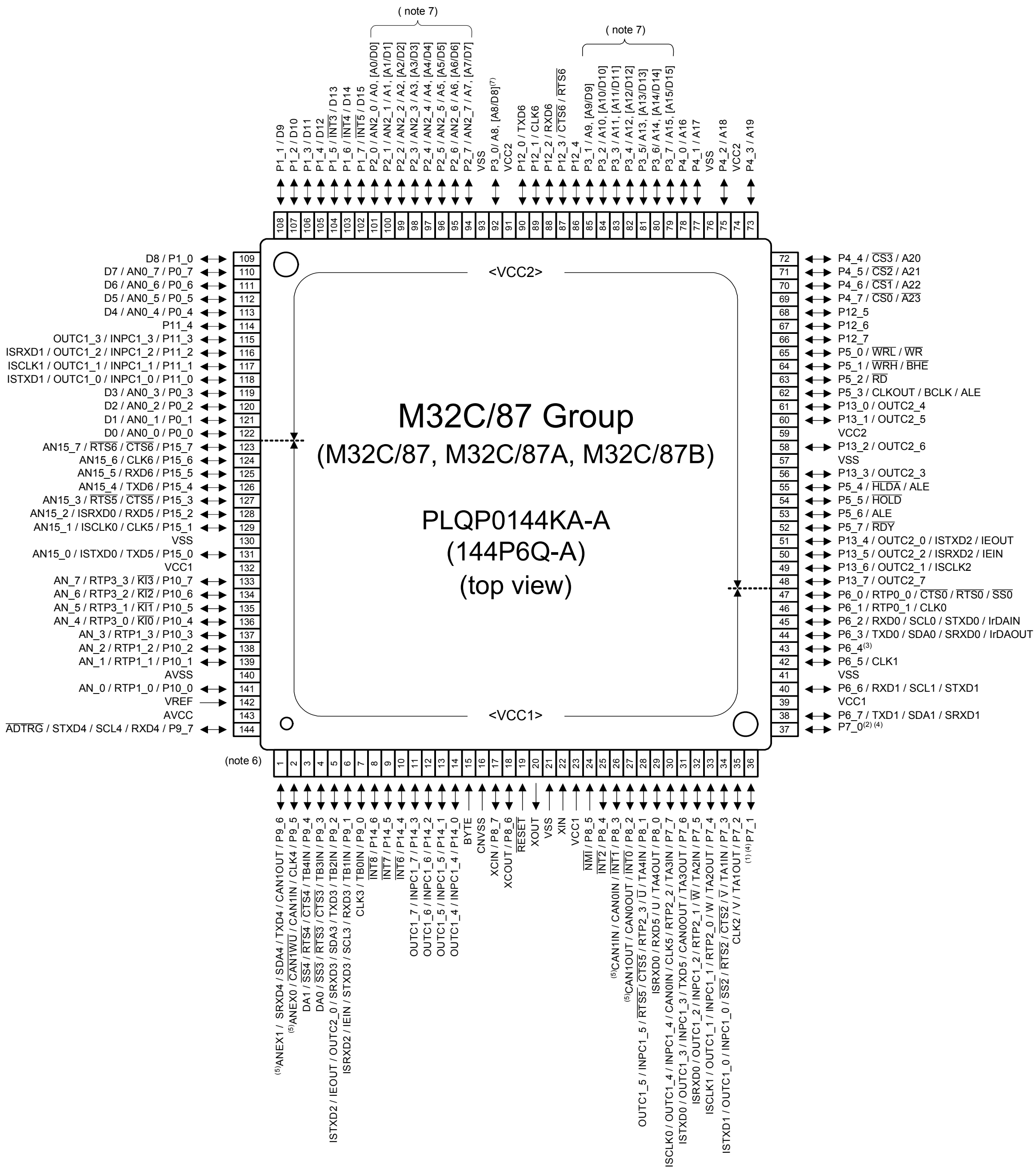


NOTES:
1. Ports P11 to P15 are provided in the 144-pin package only.
2. 34 channels are available in the 144-pin package. 26 channels are available in the 100-pin package.
3. 6 channels are available in the 100-pin package.
4. 10 channels are available in the 100-pin package.
5. M32C/87A has 1 channel. M32C/87B has no CAN module.

Figure 1.2 M32C/87 Group (M32C/87, M32C/87A, M32C/87B) Block Diagram

1.4 Pin Assignments

Figures 1.3 to 1.5 show pin assignments (top view).



NOTES:

1. P7_1 / TA0IN / TB5IN / RTP0_3 / RXD2 / SCL2 / STXD2 / INPC1_7 / OUTC1_7 / OUTC2_2 / ISRXD2 / IEIN
2. P7_0 / TA0OUT / RTP0_2 / TXD2 / SDA2 / SRXD2 / INPC1_6 / OUTC1_6 / OUTC2_0 / ISTXD2 / IEOUT
3. P6_4 / CTS1 / RTS1 / SS1 / OUTC2_1 / ISCLK2
4. P7_0 and P7_1 are N-channel open drain output ports.
5. The CAN pins cannot be used in M32C/87B. Only CAN0 pins can be used in M32C/87A.
6. Refer to **Package Dimensions** for the pin1 position on the package.
7. 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 1.8 144-Pin Package List of Pin Names (1/4)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin[1]	Intelligent I/O Pin	Analog Pin	Bus Control Pin
1		P9_6			TXD4/SDA4/SRXD4/CAN1OUT		ANEX1	
2		P9_5			CLK4/CAN1IN/$\overline{CAN1WU}$		ANEX0	
3		P9_4		TB4IN	$\overline{CTS4}$/$\overline{RTS4}$/$\overline{SS4}$		DA1	
4		P9_3		TB3IN	$\overline{CTS3}$/$\overline{RTS3}$/$\overline{SS3}$		DA0	
5		P9_2		TB2IN	TXD3/SDA3/SRXD3	OUTC2_0/IEOUT/ISTXD2		
6		P9_1		TB1IN	RXD3/SCL3/STXD3	IEIN/ISRXD2		
7		P9_0		TB0IN	CLK3			
8		P14_6	$\overline{INT8}$					
9		P14_5	$\overline{INT7}$					
10		P14_4	$\overline{INT6}$					
11		P14_3				INPC1_7/OUTC1_7		
12		P14_2				INPC1_6/OUTC1_6		
13		P14_1				INPC1_5/OUTC1_5		
14		P14_0				INPC1_4/OUTC1_4		
15	BYTE							
16	CNVSS							
17	XCIN	P8_7						
18	XCOUT	P8_6						
19	$\overline{RESET}$							
20	XOUT							
21	VSS							
22	XIN							
23	VCC1							
24		P8_5	$\overline{NMI}$					
25		P8_4	$\overline{INT2}$					
26		P8_3	$\overline{INT1}$		CAN0IN/CAN1IN			
27		P8_2	$\overline{INT0}$		CAN0OUT/CAN1OUT			
28		P8_1		TA4IN/$\overline{U}$/RTP2_3	$\overline{CTS5}$/$\overline{RTS5}$	INPC1_5/OUTC1_5		
29		P8_0		TA4OUT/U	RXD5	ISRXD0		
30		P7_7		TA3IN/RTP2_2	CLK5/CAN0IN	INPC1_4/OUTC1_4/ISCLK0		
31		P7_6		TA3OUT	TXD5/CAN0OUT	INPC1_3/OUTC1_3/ISTXD0		
32		P7_5		TA2IN/$\overline{W}$/RTP2_1		INPC1_2/OUTC1_2/ISRXD1		
33		P7_4		TA2OUT/W/RTP2_0		INPC1_1/OUTC1_1/ISCLK1		
34		P7_3		TA1IN/$\overline{V}$	$\overline{CTS2}$/$\overline{RTS2}$/$\overline{SS2}$	INPC1_0/OUTC1_0/ISTXD1		
35		P7_2		TA1OUT/V	CLK2			
36		P7_1		TA0IN/TB5IN/RTP0_3	RXD2/SCL2/STXD2	INPC1_7/OUTC1_7/OUTC2_2/ISRXD2/IEIN		
37		P7_0		TA0OUT/RTP0_2	TXD2/SDA2/SRXD2	INPC1_6/OUTC1_6/OUTC2_0/ISTXD2/IEOUT		
38		P6_7			TXD1/SDA1/SRXD1			
39	VCC1							
40		P6_6			RXD1/SCL1/STXD1			

NOTE:
1. The CAN pins cannot be used in M32C/87B. Only CAN0 pins can be used in M32C/87A.

Table 1.9 144-Pin Package List of Pin Names (2/4)

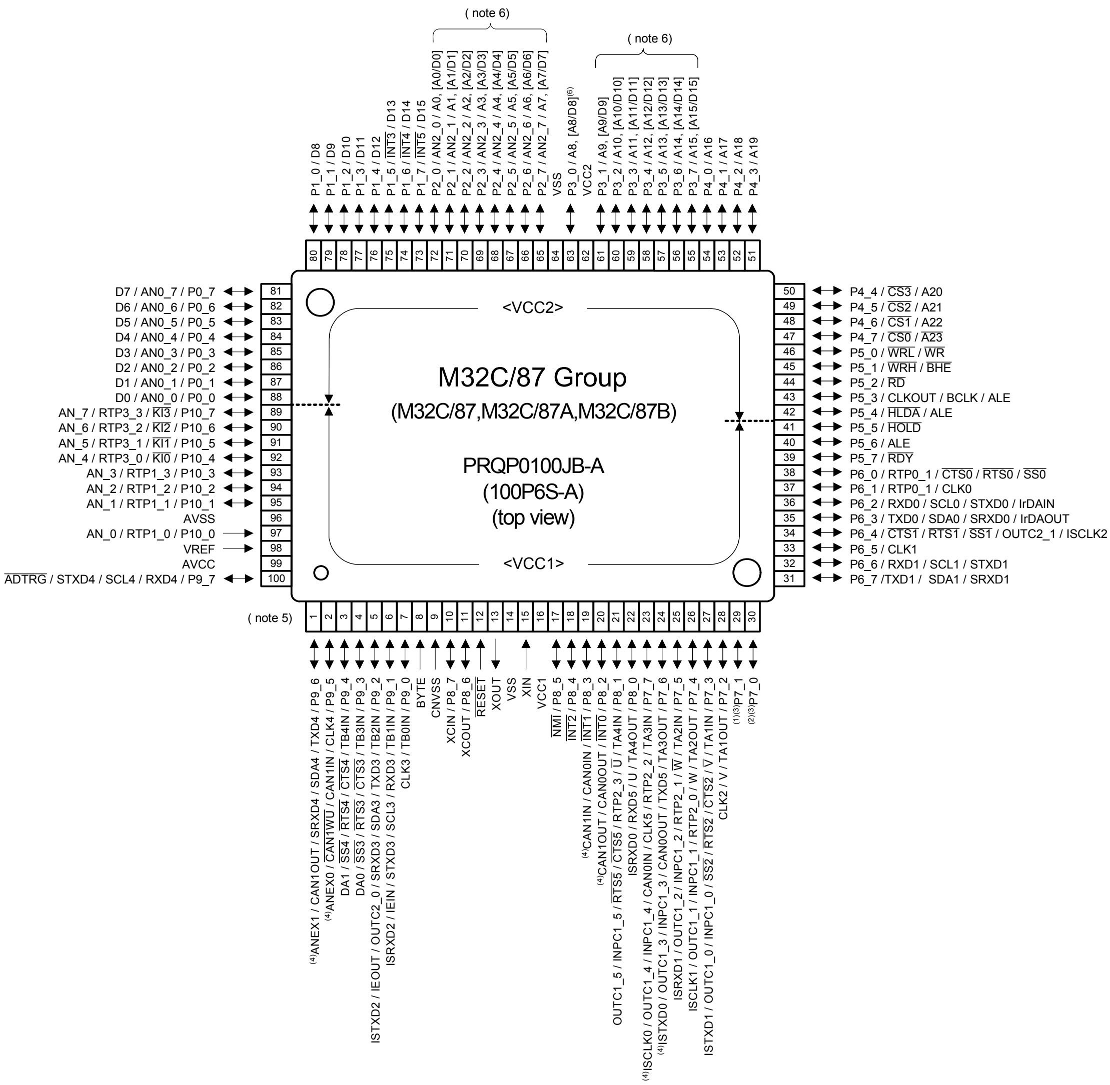
Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
41	VSS							
42		P6_5			CLK1			
43		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2		
44		P6_3			TXD0/SDA0/SRXD0/ IrDAOUT			
45		P6_2			RXD0/SCL0/STXD0/ IrDAIN			
46		P6_1		RTP0_1	CLK0			
47		P6_0		RTP0_0	CTS0/RTS0/SS0			
48		P13_7				OUTC2_7		
49		P13_6				OUTC2_1/ISCLK2		
50		P13_5				OUTC2_2/ISRXD2/ IEIN		
51		P13_4				OUTC2_0/ISTXD2/ IEOUT		
52		P5_7						RDY
53		P5_6						ALE
54		P5_5						HOLD
55		P5_4						HLDA/ALE
56		P13_3				OUTC2_3		
57	VSS							
58		P13_2				OUTC2_6		
59	VCC2							
60		P13_1				OUTC2_5		
61		P13_0				OUTC2_4		
62	CLKOUT	P5_3						BCLK/ALE
63		P5_2						RD
64		P5_1						WRH/BHE
65		P5_0						WRL/WR
66		P12_7						
67		P12_6						
68		P12_5						
69		P4_7						CS0/A23
70		P4_6						CS1/A22
71		P4_5						CS2/A21
72		P4_4						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]

Table 1.10 144-Pin Package List of Pin Names (3/4)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
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			$\overline{CTS6}$/$\overline{RTS6}$			
88		P12_2			RXD6			
89		P12_1			CLK6			
90		P12_0			TXD6			
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]
101		P2_0					AN2_0	A0,[A0/D0]
102		P1_7	$\overline{INT5}$					D15
103		P1_6	$\overline{INT4}$					D14
104		P1_5	$\overline{INT3}$					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				INPC1_3/OUTC1_3		
116		P11_2				INPC1_2/OUTC1_2/ ISRXD1		
117		P11_1				INPC1_1/OUTC1_1/ ISCLK1		
118		P11_0				INPC1_0/OUTC1_0/ ISTXD1		
119		P0_3					AN0_3	D3
120		P0_2					AN0_2	D2

Table 1.11 144-Pin Package List of Pin Names (4/4)

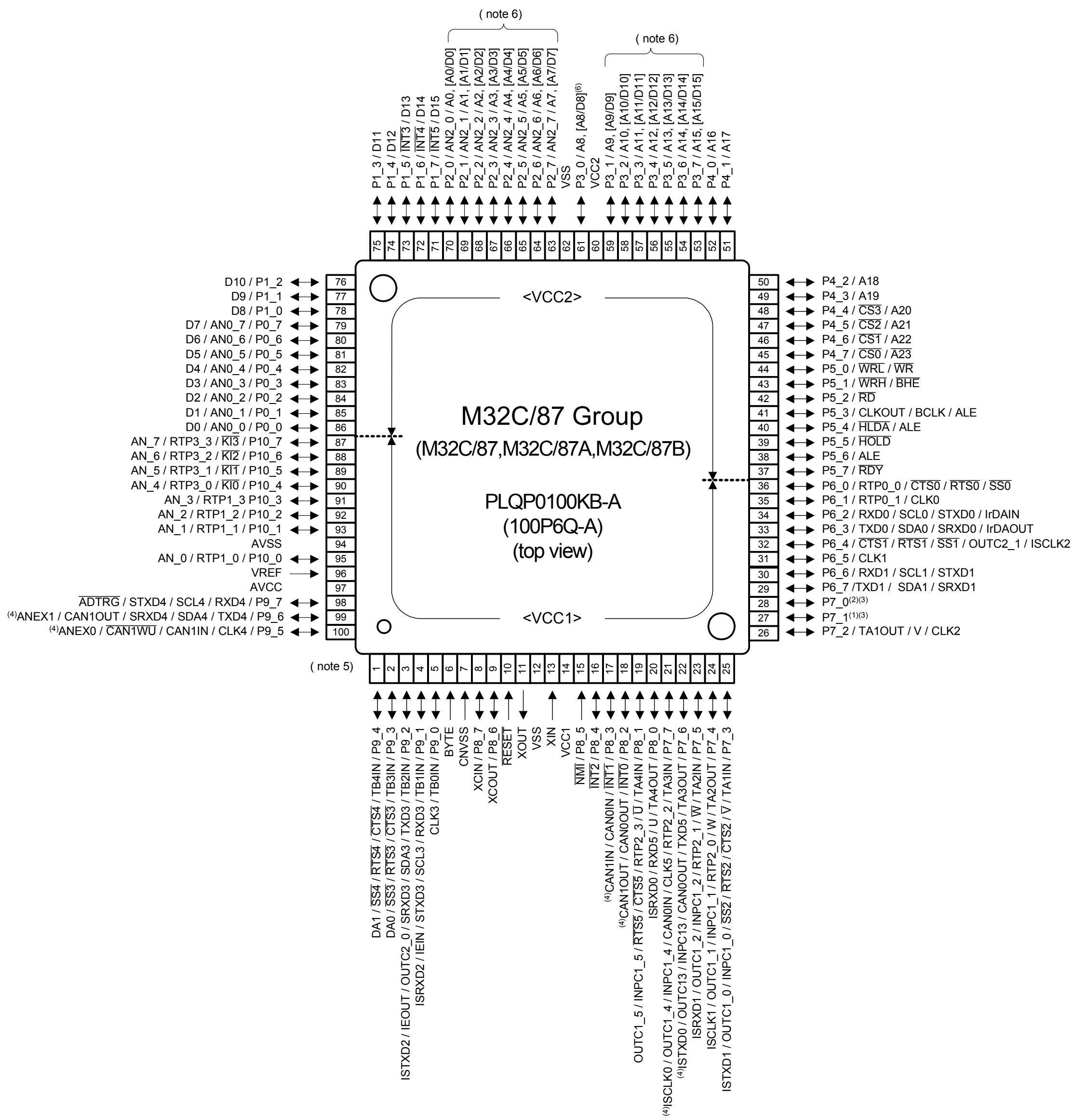
Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
121		P0_1					AN0_1	D1
122		P0_0					AN0_0	D0
123		P15_7			$\overline{CTS6}$/$\overline{RTS6}$		AN15_7	
124		P15_6			CLK6		AN15_6	
125		P15_5			RXD6		AN15_5	
126		P15_4			TXD6		AN15_4	
127		P15_3			$\overline{CTS5}$/$\overline{RTS5}$		AN15_3	
128		P15_2			RXD5	ISRXD0	AN15_2	
129		P15_1			CLK5	ISCLK0	AN15_1	
130	VSS							
131		P15_0			TXD5	ISTXD0	AN15_0	
132	VCC1							
133		P10_7	$\overline{KI3}$	RTP3_3			AN_7	
134		P10_6	$\overline{KI2}$	RTP3_2			AN_6	
135		P10_5	$\overline{KI1}$	RTP3_1			AN_5	
136		P10_4	$\overline{KI0}$	RTP3_0			AN_4	
137		P10_3		RTP1_3			AN_3	
138		P10_2		RTP1_2			AN_2	
139		P10_1		RTP1_1			AN_1	
140	AVSS							
141		P10_0		RTP1_0			AN_0	
142	VREF							
143	AVCC							
144		P9_7			RXD4/SCL4/STXD4		$\overline{ADTRG}$	



NOTES:

1. P7_1 / TA0IN / TB5IN / RTP0_3 / RXD2 / SCL2 / STXD2 / INPC1_7 / OUTC1_7 / OUTC2_2 / ISRXD2 / IEIN
2. P7_0 / TA0OUT / RTP0_2 / TXD2 / SDA2 / SRXD2 / INPC1_6 / OUTC1_6 / OUTC2_0 / ISTXD2 / IEOUT
3. P7_0 and P7_1 are N-channel open drain output ports.
4. The CAN pins cannot be used in M32C/87B. Only CAN0 pins can be used in M32C/87A.
5. Refer to **Package Dimensions** for the pin1 position on the package.
6. 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



NOTES:
1. P7_1 / TA0IN / TB5IN / RTP0_3 / RXD2 / SCL2 / STXD2 / INPC1_7 / OUTC1_7 / OUTC2_2 / ISRXD2 / IEIN
2. P7_0 / TA0OUT / RTP0_2 / TXD2 / SDA2 / SRXD2 / INPC1_6 / OUTC1_6 / OUTC2_0 / ISTXD2 / IEOUT
3. P7_0 and P7_1 are N-channel open drain output ports.
4. The CAN pins cannot be used in M32C/87B. Only CAN0 pins can be used in M32C/87A.
5. Refer to **Package Dimensions** for the pin1 position on the package.
6. Pin names in brackets [ ] represent a single functional signal. They should not be considered as two separate functional signals.

Figure 1.5 Pin Assignment for 100-Pin Package

Table 1.12 100-Pin Package List of Pin Names (1/3)

Pin No. FP	Pin No. GP	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin[1]	Intelligent I/O Pin	Analog Pin	Bus Control Pin
1	99		P9_6			TXD4/SDA4/SRXD4/ CAN1OUT		ANEX1	
2	100		P9_5			CLK4/CAN1IN/ CAN1WU		ANEX0	
3	1		P9_4		TB4IN	CTS4/RTS4/SS4		DA1	
4	2		P9_3		TB3IN	CTS3/RTS3/SS3		DA0	
5	3		P9_2		TB2IN	TXD3/SDA3/SRXD3	OUTC2_0/IEOUT/ISTXD2		
6	4		P9_1		TB1IN	RXD3/SCL3/STXD3	IEIN/ISRXD2		
7	5		P9_0		TB0IN	CLK3			
8	6	BYTE							
9	7	CNVSS							
10	8	XCIN	P8_7						
11	9	XCOUT	P8_6						
12	10	RESET							
13	11	XOUT							
14	12	VSS							
15	13	XIN							
16	14	VCC1							
17	15		P8_5	NMI					
18	16		P8_4	INT2					
19	17		P8_3	INT1		CAN0IN/CAN1IN			
20	18		P8_2	INT0		CAN0OUT/CAN1OUT			
21	19		P8_1		TA4IN/U/RTP2_3	CTS5/RTS5	INPC1_5/OUTC1_5		
22	20		P8_0		TA4OUT/U	RXD5	ISRXD0		
23	21		P7_7		TA3IN/RTP2_2	CLK5/CAN0IN	INPC1_4/OUTC1_4/ ISCLK0		
24	22		P7_6		TA3OUT	TXD5/CAN0OUT	INPC1_3/OUTC1_3/ ISTXD0		
25	23		P7_5		TA2IN/W/RTP2_1		INPC1_2/OUTC1_2 ISRXD1		
26	24		P7_4		TA2OUT/W/ RTP2_0		INPC1_1/OUTC1_1/ ISCLK1		
27	25		P7_3		TA1IN/V	CTS2/RTS2/SS2	INPC1_0/OUTC1_0/ ISTXD1		
28	26		P7_2		TA1OUT/V	CLK2			
29	27		P7_1		TA0IN/TB5IN/ RTP0_3	RXD2/SCL2/STXD2	INPC1_7/OUTC1_7/ OUTC2_2/ISRXD2/IEIN		
30	28		P7_0		TA0OUT/RTP0_2	TXD2/SDA2/SRXD2	INPC1_6/OUTC1_6/ OUTC2_0/ISTXD2/IEOUT		
31	29		P6_7			TXD1/SDA1/SRXD1			
32	30		P6_6			RXD1/SCL1/STXD1			
33	31		P6_5			CLK1			
34	32		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2		
35	33		P6_3			TXD0/SDA0/SRXD0/ IrDAOUT			
36	34		P6_2			RXD0/SCL0/STXD0/ IrDAIN			
37	35		P6_1		RTP0_1	CLK0			
38	36		P6_0		RTP0_0	CTS0/RTS0/SS0			
39	37		P5_7						RDY
40	38		P5_6						ALE

NOTE:
1. The CAN pins cannot be used in M32C/87B. Only CAN0 pins can be used in M32C/87A.

Table 1.13 100-Pin Package List of Pin Names (2/3)

Pin No. FP	Pin No. GP	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
41	39		P5_5						$\overline{\text{HOLD}}$
42	40		P5_4						$\overline{\text{HLDA}}$/ALE
43	41	CLKOUT	P5_3						BCLK/ALE
44	42		P5_2						$\overline{\text{RD}}$
45	43		P5_1						$\overline{\text{WRH}}$/$\overline{\text{BHE}}$
46	44		P5_0						$\overline{\text{WRL}}$/$\overline{\text{WR}}$
47	45		P4_7						$\overline{\text{CS0}}$/$\overline{\text{A23}}$
48	46		P4_6						$\overline{\text{CS1}}$/A22
49	47		P4_5						$\overline{\text{CS2}}$/A21
50	48		P4_4						$\overline{\text{CS3}}$/A20
51	49		P4_3						A19
52	50		P4_2						A18
53	51		P4_1						A17
54	52		P4_0						A16
55	53		P3_7						A15,[A15/D15]
56	54		P3_6						A14,[A14/D14]
57	55		P3_5						A13,[A13/D13]
58	56		P3_4						A12,[A12/D12]
59	57		P3_3						A11,[A11/D11]
60	58		P3_2						A10,[A10/D10]
61	59		P3_1						A9,[A9/D9]
62	60	VCC2							
63	61		P3_0						A8,[A8/D8]
64	62	VSS							
65	63		P2_7					AN2_7	A7,[A7/D7]
66	64		P2_6					AN2_6	A6,[A6/D6]
67	65		P2_5					AN2_5	A5,[A5/D5]
68	66		P2_4					AN2_4	A4,[A4/D4]
69	67		P2_3					AN2_3	A3,[A3/D3]
70	68		P2_2					AN2_2	A2,[A2/D2]
71	69		P2_1					AN2_1	A1,[A1/D1]
72	70		P2_0					AN2_0	A0,[A0/D0]

Table 1.14 100-Pin Package List of Pin Names (3/3)

Pin No.		Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN Pin	Intelligent I/O Pin	Analog Pin	Bus Control Pin
FP	GP								
73	71		P1_7	$\overline{INT5}$					D15
74	72		P1_6	$\overline{INT4}$					D14
75	73		P1_5	$\overline{INT3}$					D13
76	74		P1_4						D12
77	75		P1_3						D11
78	76		P1_2						D10
79	77		P1_1						D9
80	78		P1_0						D8
81	79		P0_7					AN0_7	D7
82	80		P0_6					AN0_6	D6
83	81		P0_5					AN0_5	D5
84	82		P0_4					AN0_4	D4
85	83		P0_3					AN0_3	D3
86	84		P0_2					AN0_2	D2
87	85		P0_1					AN0_1	D1
88	86		P0_0					AN0_0	D0
89	87		P10_7	$\overline{KI3}$	RTP3_3			AN_7	
90	88		P10_6	$\overline{KI2}$	RTP3_2			AN_6	
91	89		P10_5	$\overline{KI1}$	RTP3_1			AN_5	
92	90		P10_4	$\overline{KI0}$	RTP3_0			AN_4	
93	91		P10_3		RTP1_3			AN_3	
94	92		P10_2		RTP1_2			AN_2	
95	93		P10_1		RTP1_1			AN_1	
96	94	AVSS							
97	95		P10_0		RTP1_0			AN_0	
98	96	VREF							
99	97	AVCC							
100	98		P9_7			RXD4/SCL4/STXD4		$\overline{ADTRG}$	

1.5 Pin Functions

Table 1.15 Pin Functions (100-Pin and 144-Pin Packages) (1/4)

Type	Symbol	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. The input condition of VCC1 ≥ VCC2 must be met.
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	RESET	I	VCC1	The MCU is placed in the reset state while applying an “L” signal to the 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 (mask ROM, flash memory version) and boot mode (flash memory version).
External data bus width select input	BYTE	I	VCC1	This pin switches a 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.
	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.
	CS0 to CS3	O	VCC2	Chip-select signal output pins used to specify external devices.
	WRL/WR WRH/BHE RD	O	VCC2	WRL, WRH, (WR, BHE) and RD signal output pins. WRL and WRH can be switched with WR and BHE by a program. • WRL, WRH and 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 WRL pin. Data is written to an odd address while an “L” is output from the WRH pin. Data is read while an “L” is output from the RD pin. • WR, BHE and RD are selected: Data is written while an “L” is output from the WR pin. Data is read while an “L” is output from the RD pin. Data in odd address is accessed while an “L” is output from the BHE pin. Select WR, BHE and 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.
	HOLD	I	VCC2	The MCU is placed in a hold state while an “L” signal is applied to the HOLD pin.
	HLDA	O	VCC2	The HLDA pin outputs an “L” while the MCU is placed in a hold state.
	RDY	I	VCC2	Bus is placed in a wait state while an “L” signal is applied to the RDY pin.

I: Input O: Output I/O: Input and output

Table 1.16 Pin Functions (100-Pin and 144-Pin Packages) (2/4)

Type	Symbol	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.
	$\overline{\text{INT3}}$ 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 $\overline{\text{CTS5}}$	I	VCC1	Input pins to control data transmission.
	$\overline{\text{RTS0}}$ to $\overline{\text{RTS5}}$	O	VCC1	Output pins to control data reception.
	CLK0 to CLK5	I/O	VCC1	Serial clock input/output pins.
	RXD0 to RXD5	I	VCC1	Serial data input pins.
	TXD0 to TXD5	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.
IrDA	IrDAIN	I	VCC1	IrDA serial data input pin.
	IrDAOUT	O	VCC1	IrDA serial data output pin.
CAN[1]	CAN0IN, CAN1IN	I	VCC1	Received data input pins for the CAN communication function.
	CAN0OUT, CAN1OUT	O	VCC1	Transmit data output pins for the CAN communication function.
	$\overline{\text{CAN1WU}}$	I	VCC1	CAN wake-up interrupt input pin.

I: Input O: Output I/O: Input and output

NOTE:

1. The CAN pins cannot be used in M32C/87B. Only CAN0 pins can be used in M32C/87A.

Table 1.17 Pin Functions (100-Pin and 144-Pin Package) (3/4)

Type	Symbol	I/O Type	Supply Voltage	Description
Intelligent I/O	INPC1_0 to INPC1_3	I	VCC1/ VCC2(1)	Input pins for the time measurement function.
	INPC1_4 to INPC1_7	I	VCC1	
	OUTC1_0 to OUTC1_3	O	VCC1/ VCC2(1)	Output pins for the waveform generation function. (OUTC1_6/OUTC2_0 and OUTC1_7/OUTC2_2 assigned to ports 7_0 and 7_1 are N-channel open drain output.)
	OUTC1_4 to OUTC1_7	O	VCC1	
	OUTC2_0 to OUTC2_2	O	VCC1/ VCC2(1)	
	ISCLK0	I/O	VCC1	Clock input/output pins for the intelligent I/O communication function.
	ISCLK1, ISCLK2	I/O	VCC1/ VCC2(1)	
	ISRXD0	I	VCC1	Data input pins for the intelligent I/O communication function.
	ISRXD1, ISRXD2	I	VCC1/ VCC2(1)	
	ISTXD0	O	VCC1	Data output pins for the intelligent I/O communication function. (ISTXD2 assigned to port 7_0 is N-channel open drain output.)
	ISTXD1, ISTXD2	O	VCC1/ VCC2(1)	
	IEIN	I	VCC1/ VCC2(1)	Data input pin for the intelligent I/O communication function.
	IEOUT	O	VCC1/ VCC2(1)	Data output pin for the intelligent I/O communication function. (IEOUT assigned to port 7_0 is N-channel open drain output.)
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	
	$\overline{ADTRG}$	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.
Real-time port	RTP0_0 to RTP0_3 RTP1_0 to RTP1_3 RTP2_0 to RTP2_3 RTP3_0 to RTP3_3	O	VCC1	These pins function as real-time ports. (RTP0_2 and RTP0_3 are N-channel open drain output.)

I: Input O: Output I/O: Input and output
NOTE:
1. Only VCC1 can be used in the 100-pin package.

Table 1.18 Pin Functions (100-Pin and 144-Pin Package) (4/4)

Type	Symbol	I/O Type	Supply Voltage	Description
I/O port	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7	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	I/O	VCC1	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.

I: Input O: Output I/O: Input and output

Table 1.19 Pin Functions (144-Pin Package Only)

Type	Symbol	I/O Type	Supply Voltage	Description
$\overline{\text{INT}}$ Interrupt Input	$\overline{\text{INT6}}$ to $\overline{\text{INT8}}$	I	VCC1	$\overline{\text{INT}}$ interrupt input pins.
Serial interface	$\overline{\text{CTS6}}$	I	VCC1/ VCC2	Input pin to control data transmission.
	$\overline{\text{RTS6}}$	O	VCC1/ VCC2	Output pin to control data reception.
	CLK6	I/O	VCC1/ VCC2	Serial clock input/output pin.
	RXD6	I	VCC1/ VCC2	Serial data input pin.
	TXD6	O	VCC1/ VCC2	Serial data output pin.
Intelligent I/O	OUTC2_3 to OUTC2_7	O	VCC2	Output pins for the waveform generation function.
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	These I/O ports are functionally equivalent to P0.

I: Input O: Output I/O: Input and output

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.

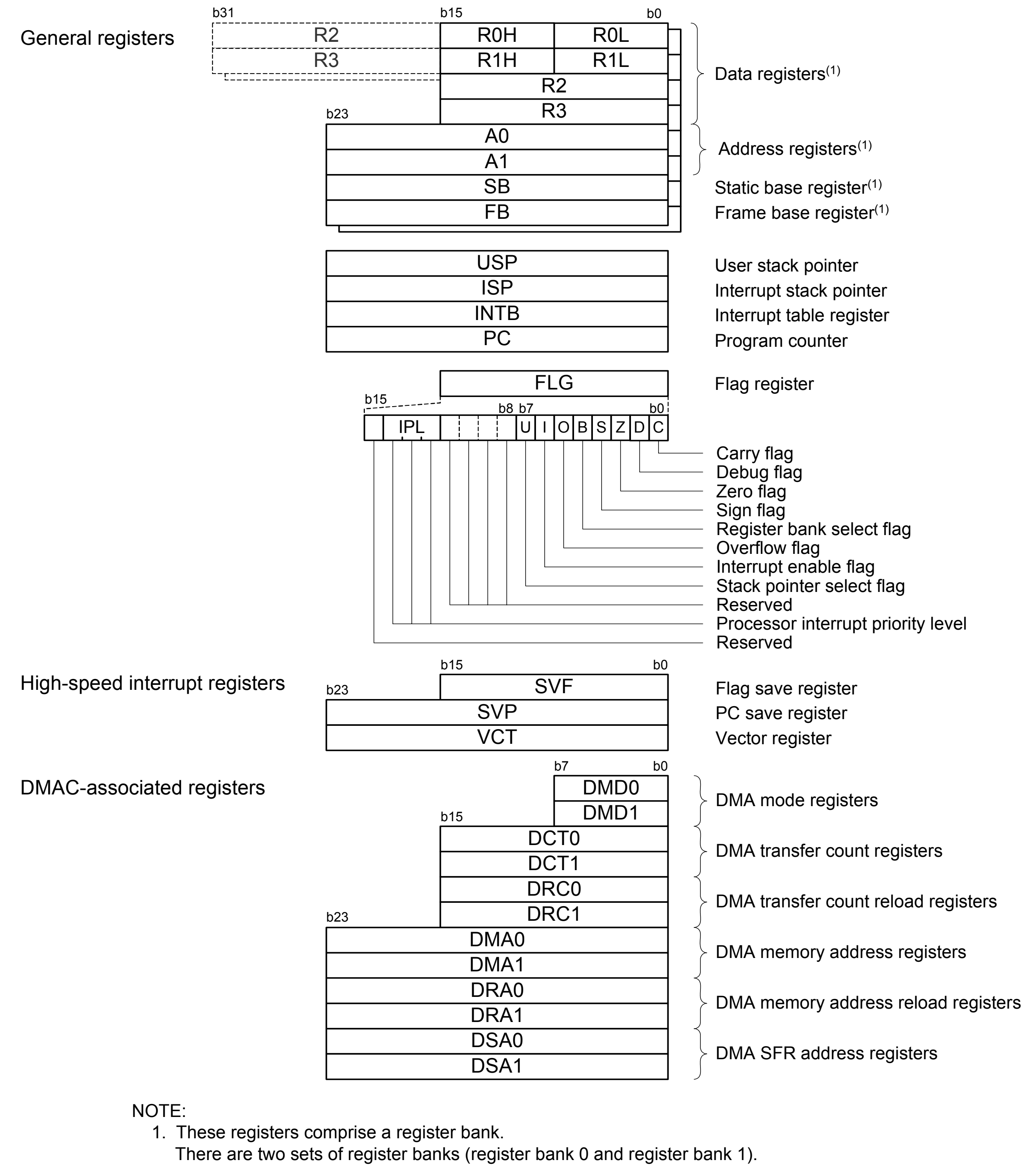


Figure 2.1 CPU Register

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)

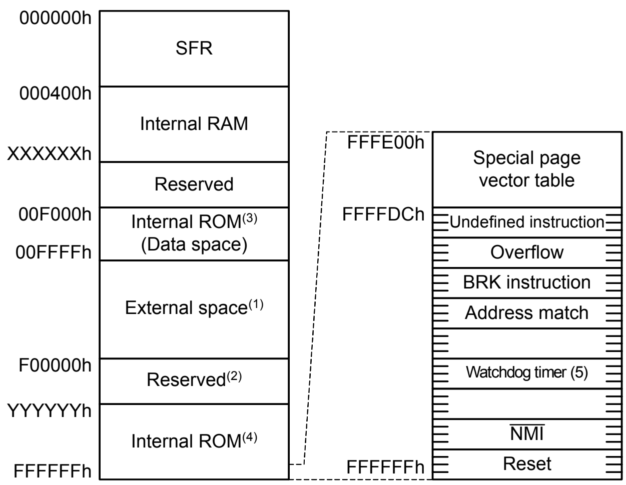
Refer to **13. DMAC** for details.

3. Memory

Figure 3.1 shows a memory map of the M32C/87 Group (M32C/87, M32C/87A, M32C/87B).
The M32C/87 Group (M32C/87, M32C/87A, M32C/87B) 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 512-Kbyte internal ROM area is allocated in addresses F80000h 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 48-Kbyte internal RAM area is allocated in addresses 000400h to 00C3FFh. 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
24 Kbytes	0063FFh
31 Kbytes	007FFFh
48 Kbytes	00C3FFh

Internal ROM	
Capacity	YYYYYYh
384 Kbytes	FA0000h
512 Kbytes	F80000h
768 Kbytes	F40000h
1024 Kbytes	F00000h



NOTES:
1. The space is used as the external space in memory expansion mode and in microprocessor mode. It is reserved in single-ship mode.
2. The space is reserved in memory expansion mode. It is used as the external space in microprocessor mode.
3. Additional 4-Kbyte space is 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 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 Vdet4 detection interrupt use 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 4.20 list SFR address maps.

Table 4.1 SFR Address Map (1/20)

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	00XX XXXXb
0010h 0011h 0012h	Address Match Interrupt Register 0	RMAD0	000000h
0013h	Processor Mode Register 2	PM2	00h
0014h 0015h 0016h	Address Match Interrupt Register 1	RMAD1	000000h
0017h	Voltage Detection Register 2	VCR2	00h
0018h 0019h 001Ah	Address Match Interrupt Register 2	RMAD2	000000h
001Bh	Voltage Detection Register 1	VCR1	0000 1000b
001Ch 001Dh 001Eh	Address Match Interrupt Register 3	RMAD3	000000h
001Fh			
0020h			
0021h			
0022h			
0023h			
0024h			
0025h			
0026h	PLL Control Register 0	PLC0	0001 X010b
0027h	PLL Control Register 1	PLC1	000X 0000b
0028h 0029h 002Ah	Address Match Interrupt Register 4	RMAD4	000000h
002Bh			
002Ch 002Dh 002Eh	Address Match Interrupt Register 5	RMAD5	000000h
002Fh	Vdet4 Detection Interrupt Register	D4INT	XX00 0000b

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 (2/20)

Address	Register	Symbol	After Reset
0030h			
0031h			
0032h			
0033h			
0034h			
0035h			
0036h			
0037h			
0038h			
0039h	Address Match Interrupt Register 6	RMAD6	000000h
003Ah			
003Bh			
003Ch			
003Dh	Address Match Interrupt Register 7	RMAD7	000000h
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			
004Dh			
004Eh			
004Fh			
0050h			
0051h			
0052h			
0053h			
0054h			
0055h	Flash Memory Control Register 1	FMR1	0000 0X0Xb
0056h			
0057h	Flash Memory Control Register 0	FMR0	0000 0001b(Flash Memory) XXXX XXX0b(Mask ROM)
0058h			
0059h			
005Ah			
005Bh			
005Ch			
005Dh			
005Eh			
005Fh			

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

Table 4.3 SFR Address Map (3/20)

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	II/O Interrupt Control Register 0 / CAN1 interrupt Control Register 0	IIO0IC/CAN3IC	XXXX X000b
0076h	Timer B1 Interrupt Control Register	TB1IC	XXXX X000b
0077h	II/O Interrupt Control Register 2	IIO2IC	XXXX X000b
0078h	Timer B3 Interrupt Control Register	TB3IC	XXXX X000b
0079h	II/O Interrupt Control Register 4	IIO4IC	XXXX X000b
007Ah	$\overline{\text{INT5}}$ Interrupt Control Register	INT5IC	XX00 X000b
007Bh	II/O Interrupt Control Register 6	IIO6IC	XXXX X000b
007Ch	$\overline{\text{INT3}}$ Interrupt Control Register	INT3IC	XX00 X000b
007Dh	II/O Interrupt Control Register 8	IIO8IC	XXXX X000b
007Eh	$\overline{\text{INT1}}$ Interrupt Control Register	INT1IC	XX00 X000b
007Fh	II/O Interrupt Control Register 10 / CAN0 Interrupt Control Register 1	IIO10IC/CAN1IC	XXXX X000b
0080h			
0081h	II/O Interrupt Control Register 11 / CAN0 Interrupt Control Register 2	IIO11IC/CAN2IC	XXXX X000b
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 (4/20)

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	II/O Interrupt Control Register 1 / CAN1 Interrupt Control Register 1	IIO1IC/CAN4IC	XXXX X000b
0096h	Timer B2 Interrupt Control Register	TB2IC	XXXX X000b
0097h	II/O Interrupt Control Register 3	IIO3IC	XXXX X000b
0098h	Timer B4 Interrupt Control Register	TB4IC	XXXX X000b
0099h	II/O Interrupt Control Register 5 /CAN1 Interrupt Control Register 2	IIO5IC/CAN5IC	XXXX X000b
009Ah	$\overline{\text{INT4}}$ Interrupt Control Register	INT4IC	XX00 X000b
009Bh	II/O Interrupt Control Register 7	IIO7IC	XXXX X000b
009Ch	$\overline{\text{INT2}}$ Interrupt Control Register	INT2IC	XX00 X000b
009Dh	II/O Interrupt Control Register 9 / CAN0 Interrupt Control Register 0	IIO9IC/CAN0IC	XXXX X000b
009Eh	$\overline{\text{INT0}}$ Interrupt Control Register	INT0IC	XX00 X000b
009Fh	Exit Priority Register	RLVL	XXXX 0000b
00A0h	Interrupt Request Register 0	IIO0IR	0000 000Xb
00A1h	Interrupt Request Register 1	IIO1IR	0000 000Xb
00A2h	Interrupt Request Register 2	IIO2IR	0000 000Xb
00A3h	Interrupt Request Register 3	IIO3IR	0000 000Xb
00A4h	Interrupt Request Register 4	IIO4IR	0000 000Xb
00A5h	Interrupt Request Register 5	IIO5IR	0000 000Xb
00A6h	Interrupt Request Register 6	IIO6IR	0000 000Xb
00A7h	Interrupt Request Register 7	IIO7IR	0000 000Xb
00A8h	Interrupt Request Register 8	IIO8IR	0000 000Xb
00A9h	Interrupt Request Register 9	IIO9IR	0000 000Xb
00AAh	Interrupt Request Register 10	IIO10IR	0000 000Xb
00ABh	Interrupt Request Register 11	IIO11IR	0000 000Xb
00ACh			
00ADh			
00AEh			
00AFh			
00B0h	Interrupt Enable Register 0	IIO0IE	00h
00B1h	Interrupt Enable Register 1	IIO1IE	00h
00B2h	Interrupt Enable Register 2	IIO2IE	00h
00B3h	Interrupt Enable Register 3	IIO3IE	00h
00B4h	Interrupt Enable Register 4	IIO4IE	00h
00B5h	Interrupt Enable Register 5	IIO5IE	00h
00B6h	Interrupt Enable Register 6	IIO6IE	00h
00B7h	Interrupt Enable Register 7	IIO7IE	00h
00B8h	Interrupt Enable Register 8	IIO8IE	00h
00B9h	Interrupt Enable Register 9	IIO9IE	00h
00BAh	Interrupt Enable Register 10	IIO10IE	00h
00BBh	Interrupt Enable Register 11	IIO11IE	00h
00BCh			
00BDh			
00BEh			
00BFh to 00DFh			

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

Table 4.5 SFR Address Map (5/20)

Address	Register	Symbol	After Reset
00E0h			
00E1h			
00E2h			
00E3h			
00E4h			
00E5h			
00E6h			
00E7h			
00E8h	Group 0 SI/O Receive Buffer Register	G0RB	XXXX XXXXb
00E9h			XXX0 XXXXb
00EAh	Group 0 Transmit Buffer/Receive Data Register	G0TB/G0DR	XXh
00EBh			
00ECh	Group 0 Receive Input Register	G0RI	XXh
00EDh	Group 0 SI/O Communication Mode Register	G0MR	00h
00EEh	Group 0 Transmit Output Register	G0TO	XXh
00EFh	Group 0 SI/O Communication Control Register	G0CR	0000 X011b
00F0h	Group 0 Data Compare Register 0	G0CMP0	XXh
00F1h	Group 0 Data Compare Register 1	G0CMP1	XXh
00F2h	Group 0 Data Compare Register 2	G0CMP2	XXh
00F3h	Group 0 Data Compare Register 3	G0CMP3	XXh
00F4h	Group 0 Data Mask Register 0	G0MSK0	XXh
00F5h	Group 0 Data Mask Register 1	G0MSK1	XXh
00F6h	Communication Clock Select Register	CCS	XXXX 0000b
00F7h			
00F8h 00F9h	Group 0 Receive CRC Code Register	G0RCRC	XXXXh
00FAh 00FBh	Group 0 Transmit CRC Code Register	G0TCRC	0000h
00FCh	Group 0 SI/O Expansion Mode Register	G0EMR	00h
00FDh	Group 0 SI/O Extended Receive Control Register	G0ERC	00h
00FEh	Group 0 SI/O Special Communication Interrupt Detection Register	G0IRF	0000 XXXXb
00FFh	Group 0 SI/O Extended Transmit Control Register	G0ETC	0000 0XXXb
0100h 0101h	Group 1 Time Measurement/Waveform Generation Register 0	G1TM0/G1PO0	XXXXh
0102h 0103h	Group 1 Time Measurement/Waveform Generation Register 1	G1TM1/G1PO1	XXXXh
0104h 0105h	Group 1 Time Measurement/Waveform Generation Register 2	G1TM2/G1PO2	XXXXh
0106h 0107h	Group 1 Time Measurement/Waveform Generation Register 3	G1TM3/G1PO3	XXXXh
0108h 0109h	Group 1 Time Measurement/Waveform Generation Register 4	G1TM4/G1PO4	XXXXh
010Ah 010Bh	Group 1 Time Measurement/Waveform Generation Register 5	G1TM5/G1PO5	XXXXh
010Ch 010Dh	Group 1 Time Measurement/Waveform Generation Register 6	G1TM6/G1PO6	XXXXh
010Eh 010Fh	Group 1 Time Measurement/Waveform Generation Register 7	G1TM7/G1PO7	XXXXh
0110h	Group 1 Waveform Generation Control Register 0	G1POCR0	0000 X000b
0111h	Group 1 Waveform Generation Control Register 1	G1POCR1	0X00 X000b
0112h	Group 1 Waveform Generation Control Register 2	G1POCR2	0X00 X000b
0113h	Group 1 Waveform Generation Control Register 3	G1POCR3	0X00 X000b
0114h	Group 1 Waveform Generation Control Register 4	G1POCR4	0X00 X000b
0115h	Group 1 Waveform Generation Control Register 5	G1POCR5	0X00 X000b
0116h	Group 1 Waveform Generation Control Register 6	G1POCR6	0X00 X000b
0117h	Group 1 Waveform Generation Control Register 7	G1POCR7	0X00 X000b
0118h	Group 1 Time Measurement Control Register 0	G1TMCR0	00h
0119h	Group 1 Time Measurement Control Register 1	G1TMCR1	00h

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

Table 4.6 SFR Address Map (6/20)

Address	Register	Symbol	After Reset
011Ah	Group 1 Time Measurement Control Register 2	G1TMCR2	00h
011Bh	Group 1 Time Measurement Control Register 3	G1TMCR3	00h
011Ch	Group 1 Time Measurement Control Register 4	G1TMCR4	00h
011Dh	Group 1 Time Measurement Control Register 5	G1TMCR5	00h
011Eh	Group 1 Time Measurement Control Register 6	G1TMCR6	00h
011Fh	Group 1 Time Measurement Control Register 7	G1TMCR7	00h
0120h 0121h	Group 1 Base Timer Register	G1BT	XXXXh
0122h	Group 1 Base Timer Control Register 0	G1BCR0	00h
0123h	Group 1 Base Timer Control Register 1	G1BCR1	X000 000Xb
0124h	Group 1 Time Measurement Prescaler Register 6	G1TPR6	00h
0125h	Group 1 Time Measurement Prescaler Register 7	G1TPR7	00h
0126h	Group 1 Function Enable Register	G1FE	00h
0127h	Group 1 Function Select Register	G1FS	00h
0128h 0129h	Group 1 SI/O Receive Buffer Register	G1RB	XXXX XXXXb X000 XXXXb
012Ah	Group 1 Transmit Buffer/Receive Data Register	G1TB/G1DR	XXh
012Bh			
012Ch	Group 1 Receive Input Register	G1RI	XXh
012Dh	Group 1 SI/O Communication Mode Register	G1MR	00h
012Eh	Group 1 Transmit Output Register	G1TO	XXh
012Fh	Group 1 SI/O Communication Control Register	G1CR	0000 X011b
0130h	Group 1 Data Compare Register 0	G1CMP0	XXh
0131h	Group 1 Data Compare Register 1	G1CMP1	XXh
0132h	Group 1 Data Compare Register 2	G1CMP2	XXh
0133h	Group 1 Data Compare Register 3	G1CMP3	XXh
0134h	Group 1 Data Mask Register 0	G1MSK0	XXh
0135h	Group 1 Data Mask Register 1	G1MSK1	XXh
0136h			
0137h			
0138h 0139h	Group 1 Receive CRC Code Register	G1RCRC	XXXXh
013Ah 013Bh	Group 1 Transmit CRC Code Register	G1TCRC	0000h
013Ch	Group 1 SI/O Expansion Mode Register	G1EMR	00h
013Dh	Group 1 SI/O Extended Receive Control Register	G1ERC	00h
013Eh	Group 1 SI/O Special Communication Interrupt Detection Register	G1IRF	0000 XXXXb
013Fh	Group 1 SI/O Extended Transmit Control Register	G1ETC	0000 0XXXb
0140h 0141h	Group 2 Waveform Generation Register 0	G2PO0	XXXXh
0142h 0143h	Group 2 Waveform Generation Register 1	G2PO1	XXXXh
0144h 0145h	Group 2 Waveform Generation Register 2	G2PO2	XXXXh
0146h 0147h	Group 2 Waveform Generation Register 3	G2PO3	XXXXh
0148h 0149h	Group 2 Waveform Generation Register 4	G2PO4	XXXXh
014Ah 014Bh	Group 2 Waveform Generation Register 5	G2PO5	XXXXh
014Ch 014Dh	Group 2 Waveform Generation Register 6	G2PO6	XXXXh
014Eh 014Fh	Group 2 Waveform Generation Register 7	G2PO7	XXXXh

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

Table 4.7 SFR Address Map (7/20)

Address	Register	Symbol	After Reset
0150h	Group 2 Waveform Generation Control Register 0	G2POCR0	00h
0151h	Group 2 Waveform Generation Control Register 1	G2POCR1	00h
0152h	Group 2 Waveform Generation Control Register 2	G2POCR2	00h
0153h	Group 2 Waveform Generation Control Register 3	G2POCR3	00h
0154h	Group 2 Waveform Generation Control Register 4	G2POCR4	00h
0155h	Group 2 Waveform Generation Control Register 5	G2POCR5	00h
0156h	Group 2 Waveform Generation Control Register 6	G2POCR6	00h
0157h	Group 2 Waveform Generation Control Register 7	G2POCR7	00h
0158h			
0159h			
015Ah			
015Bh			
015Ch			
015Dh			
015Eh			
015Fh			
0160h 0161h	Group 2 Base Timer Register	G2BT	XXXXh
0162h	Group 2 Base Timer Control Register 0	G2BCR0	00h
0163h	Group 2 Base Timer Control Register 1	G2BCR1	00h
0164h	Base Timer Start Register	BTSR	XXXX 0000b
0165h			
0166h	Group 2 Function Enable Register	G2FE	00h
0167h	Group 2 RTP Output Buffer Register	G2RTP	00h
0168h			
0169h			
016Ah	Group 2 SI/O Communication Mode Register	G2MR	00XX X000b
016Bh	Group 2 SI/O Communication Control Register	G2CR	0000 X000b
016Ch 016Dh	Group 2 SI/O Transmit Buffer Register	G2TB	XXXXh
016Eh 016Fh	Group 2 SI/O Receive Buffer Register	G2RB	XXXXh
0170h 0171h	Group 2 IEBus Address Register	IEAR	XXXXh
0172h	Group 2 IEBus Control Register	IECR	00XX X000b
0173h	Group 2 IEBus Transmit Interrupt Source Detection Register	IETIF	XXX0 0000b
0174h	Group 2 IEBus Receive Interrupt Source Detection Register	IERIF	XXX0 0000b
0175h			
0176h			
0177h	Input Function Select Register B	IPSB	00h
0178h	Input Function Select Register	IPS	00h
0179h	Input Function Select Register A	IPSA	00h
017Ah			
017Bh			
017Ch			
017Dh to 01BFh			

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

Table 4.8 SFR Address Map (8/20)

Address	Register	Symbol	After Reset
01C0h	UART5 Transmit/Receive Mode Register	U5MR	00h
01C1h	UART5 Baud Rate Register	U5BRG	XXh
01C2h	UART5 Transmit Buffer Register	U5TB	XXXXh
01C3h			
01C4h	UART5 Transmit/Receive Control Register 0	U5C0	0000 1000b
01C5h	UART5 Transmit/Receive Control Register 1	U5C1	XXXX 0010b
01C6h	UART5 Receive Buffer Register	U5RB	XXXXh
01C7h			
01C8h	UART6 Transmit/Receive Mode Register	U6MR	00h
01C9h	UART6 Baud Rate Register	U6BRG	XXh
01CAh	UART6 Transmit Buffer Register	U6TB	XXXXh
01CBh			
01CCh	UART6 Transmit/Receive Control Register 0	U6C0	0000 1000b
01CDh	UART6 Transmit/Receive Control Register 1	U6C1	XXXX 0010b
01CEh	UART6 Receive Buffer Register	U6RB	XXXXh
01CFh			
01D0h	UART5, UART6 Transmit/Receive Control Register	U56CON	X000 0000b
01D1h	UART5, UART6 Input Pin Function Select Register	U56IS	X000 X000b
01D2h			
01D3h			
01D4h			
01D5h			
01D6h			
01D7h			
01D8h	RTP Output Buffer Register 0	RTP0R	XXh
01D9h	RTP Output Buffer Register 1	RTP1R	XXh
01DAh	RTP Output Buffer Register 2	RTP2R	XXh
01DBh	RTP Output Buffer Register 3	RTP3R	XXh
01DCh			
01DDh			
01DEh			
01DFh			
01E0h	CAN0 Message Slot Buffer 0 Standard ID0(1)(2)	C0SLOT0_0	XXh
01E1h	CAN0 Message Slot Buffer 0 Standard ID1(1)(2)	C0SLOT0_1	XXh
01E2h	CAN0 Message Slot Buffer 0 Extended ID0(1)(2)	C0SLOT0_2	XXh
01E3h	CAN0 Message Slot Buffer 0 Extended ID1(1)(2)	C0SLOT0_3	XXh
01E4h	CAN0 Message Slot Buffer 0 Extended ID2(1)(2)	C0SLOT0_4	XXh
01E5h	CAN0 Message Slot Buffer 0 Data Length Code(1)(2)	C0SLOT0_5	XXh
01E6h	CAN0 Message Slot Buffer 0 Data 0(1)(2)	C0SLOT0_6	XXh
01E7h	CAN0 Message Slot Buffer 0 Data 1(1)(2)	C0SLOT0_7	XXh
01E8h	CAN0 Message Slot Buffer 0 Data 2(1)(2)	C0SLOT0_8	XXh
01E9h	CAN0 Message Slot Buffer 0 Data 3(1)(2)	C0SLOT0_9	XXh
01EAh	CAN0 Message Slot Buffer 0 Data 4(1)(2)	C0SLOT0_10	XXh
01EBh	CAN0 Message Slot Buffer 0 Data 5(1)(2)	C0SLOT0_11	XXh
01ECh	CAN0 Message Slot Buffer 0 Data 6(1)(2)	C0SLOT0_12	XXh
01EDh	CAN0 Message Slot Buffer 0 Data 7(1)(2)	C0SLOT0_13	XXh
01EEh	CAN0 Message Slot Buffer 0 Time Stamp High-Order(1)(2)	C0SLOT0_14	XXh
01EFh	CAN0 Message Slot Buffer 0 Time Stamp Low-Order(1)(2)	C0SLOT0_15	XXh

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:
1. The CAN-associated registers (allocated in addresses 01E0h to 02BFh) cannot be used in M32C/87B. In M32C/87A, only CAN0-associated registers can be used.
2. Set the PM13 bit in the PM1 register to 1 (2 wait states for SFR area) before accessing the CAN-associated registers.

Table 4.9 SFR Address Map (9/20)

Address	Register(2)(3)	Symbol	After Reset
01F0h	CAN0 Message Slot Buffer 1 Standard ID0	C0SLOT1_0	XXh
01F1h	CAN0 Message Slot Buffer 1 Standard ID1	C0SLOT1_1	XXh
01F2h	CAN0 Message Slot Buffer 1 Extended ID0	C0SLOT1_2	XXh
01F3h	CAN0 Message Slot Buffer 1 Extended ID1	C0SLOT1_3	XXh
01F4h	CAN0 Message Slot Buffer 1 Extended ID2	C0SLOT1_4	XXh
01F5h	CAN0 Message Slot Buffer 1 Data Length Code	C0SLOT1_5	XXh
01F6h	CAN0 Message Slot Buffer 1 Data 0	C0SLOT1_6	XXh
01F7h	CAN0 Message Slot Buffer 1 Data 1	C0SLOT1_7	XXh
01F8h	CAN0 Message Slot Buffer 1 Data 2	C0SLOT1_8	XXh
01F9h	CAN0 Message Slot Buffer 1 Data 3	C0SLOT1_9	XXh
01FAh	CAN0 Message Slot Buffer 1 Data 4	C0SLOT1_10	XXh
01FBh	CAN0 Message Slot Buffer 1 Data 5	C0SLOT1_11	XXh
01FCh	CAN0 Message Slot Buffer 1 Data 6	C0SLOT1_12	XXh
01FDh	CAN0 Message Slot Buffer 1 Data 7	C0SLOT1_13	XXh
01FEh	CAN0 Message Slot Buffer 1 Time Stamp High-Order	C0SLOT1_14	XXh
01FFh	CAN0 Message Slot Buffer 1 Time Stamp Low-Order	C0SLOT1_15	XXh
0200h	CAN0 Control Register 0	C0CTLR0	XX01 0X01b(1)
0201h			XXXX 0000b(1)
0202h	CAN0 Status Register	C0STR	0000 0000b(1)
0203h			X000 0X01b(1)
0204h 0205h	CAN0 Extended ID Register	C0IDR	0000h(1)
0206h	CAN0 Configuration Register	C0CONR	0000 XXXXb(1)
0207h			0000 0000b(1)
0208h 0209h	CAN0 Time Stamp Register	C0TSR	0000h(1)
020Ah	CAN0 Transmit Error Count Register	C0TEC	00h(1)
020Bh	CAN0 Receive Error Count Register	C0REC	00h(1)
020Ch 020Dh	CAN0 Slot Interrupt Status Register	C0SISTR	0000h(1)
020Eh			
020Fh			
0210h 0211h	CAN0 Slot Interrupt Mask Register	C0SIMKR	0000h(1)
0212h			
0213h			
0214h	CAN0 Error Interrupt Mask Register	C0EIMKR	XXXX X000b(1)
0215h	CAN0 Error Interrupt Status Register	C0EISTR	XXXX X000b(1)
0216h	CAN0 Error Source Register	C0EFR	00h(1)
0217h	CAN0 Baud Rate Prescaler	C0BRP	0000 0001b(1)
0218h			
0219h	CAN0 Mode Register	C0MDR	XXXX XX00b(1)
021Ah			
021Bh			
021Ch			
021Dh			
021Eh			
021Fh			

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:
1. Values are obtained by setting the SLEEP bit in the C0SLPR register to “1” (sleep mode exited) after reset and supplying a clock to the CAN module.
2. The CAN-associated registers (allocated in addresses 01E0h to 02BFh) cannot be used in M32C/87B. In M32C/87A, only CAN0-associated registers can be used.
3. Set the PM13 bit in the PM1 register to 1 (2 wait states for SFR area) before accessing the CAN-associated registers.

Table 4.10 SFR Address Map (10/20)

Address	Register(3)(4)	Symbol	After Reset
0220h 0221h	CAN0 Single Shot Control Register	C0SSCTLR	0000h(1)(2)
0222h			
0223h			
0224h 0225h	CAN0 Single Shot Status Register	C0SSSTR	0000h(1)(2)
0226h			
0227h			
0228h	CAN0 Global Mask Register Standard ID0	C0GMR0	XXX0 0000b(1)(2)
0229h	CAN0 Global Mask Register Standard ID1	C0GMR1	XX00 0000b(1)(2)
022Ah	CAN0 Global Mask Register Extended ID0	C0GMR2	XXXX 0000b(1)(2)
022Bh	CAN0 Global Mask Register Extended ID1	C0GMR3	00h(1)(2)
022Ch	CAN0 Global Mask Register Extended ID2	C0GMR4	XX00 0000b(1)(2)
022Dh			
022Eh			
022Fh			
0230h	CAN0 Message Slot 0 Control Register / CAN0 Local Mask Register A Standard ID0	C0MCTL0 / C0LMAR0	0000 0000b(1)(2)/ XXX0 0000b(1)(2)
0231h	CAN0 Message Slot 1 Control Register / CAN0 Local Mask Register A Standard ID1	C0MCTL1 / C0LMAR1	0000 0000b(1)(2)/ XX00 0000b(1)(2)
0232h	CAN0 Message Slot 2 Control Register / CAN0 Local Mask Register A Extended ID0	C0MCTL2 / C0LMAR2	0000 0000b(1)(2)/ XXXX 0000b(1)(2)
0233h	CAN0 Message Slot 3 Control Register / CAN0 Local Mask Register A Extended ID1	C0MCTL3 / C0LMAR3	00h(1)(2)/ 00h(1)(2)
0234h	CAN0 Message Slot 4 Control Register / CAN0 Local Mask Register A Extended ID2	C0MCTL4 / C0LMAR4	0000 0000b(1)(2)/ XX00 0000b(1)(2)
0235h	CAN0 Message Slot 5 Control Register	C0MCTL5	00h(1)(2)
0236h	CAN0 Message Slot 6 Control Register	C0MCTL6	00h(1)(2)
0237h	CAN0 Message Slot 7 Control Register	C0MCTL7	00h(1)(2)
0238h	CAN0 Message Slot 8 Control Register / CAN0 Local Mask Register B Standard ID0	C0MCTL8 / C0LMBR0	0000 0000b(1)(2)/ XXX0 0000b(1)(2)
0239h	CAN0 Message Slot 9 Control Register / CAN0 Local Mask Register B Standard ID1	C0MCTL9 / C0LMBR1	0000 0000b(1)(2)/ XX00 0000b(1)(2)
023Ah	CAN0 Message Slot 10 Control Register / CAN0 Local Mask Register B Extended ID0	C0MCTL10 / C0LMBR2	0000 0000b(1)(2)/ XXXX 0000b(1)(2)
023Bh	CAN0 Message Slot 11 Control Register / CAN0 Local Mask Register B Extended ID1	C0MCTL11 / C0LMBR3	00h(1)(2)/ 00h(1)(2)
023Ch	CAN0 Message Slot 12 Control Register / CAN0 Local Mask Register B Extended ID2	C0MCTL12 / C0LMBR4	0000 0000b(1)(2)/ XX00 0000b(1)(2)
023Dh	CAN0 Message Slot 13 Control Register	C0MCTL13	00h(1)(2)
023Eh	CAN0 Message Slot 14 Control Register	C0MCTL14	00h(1)(2)
023Fh	CAN0 Message Slot 15 Control Register	C0MCTL15	00h(1)(2)
0240h	CAN0 Slot Buffer Select Register	C0SBS	00h(2)
0241h	CAN0 Control Register 1	C0CTLR1	X000 00XXb(2)
0242h	CAN0 Sleep Control Register	C0SLPR	XXXX XXX0b
0243h			
0244h 0245h	CAN0 Acceptance Filter Support Register	C0AFS	0000 0000b(2) 0000 0001b(2)
0246h			
0247h			
0248h			
0249h			
024Ah to 024Fh			

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:
1. The BANKSEL bit in the C0CTLR1 register can switch functions for addresses 0220h to 023Fh.
2. Values are obtained by setting the SLEEP bit in the C0SLPR register to “1” (sleep mode exited) after reset and supplying a clock to the CAN module.
3. The CAN-associated registers (allocated in addresses 01E0h to 02BFh) cannot be used in M32C/87B. In M32C/87A, only CAN0-associated registers can be used.
4. Set the PM13 bit in the PM1 register to 1 (2 wait states for SFR area) before accessing the CAN-associated registers.

Table 4.11 SFR Address Map (11/20)

Address	Register[2][3]	Symbol	After Reset
0250h	CAN1 Slot Buffer Select Register	C1SBS	00h[1]
0251h	CAN1 Control Register 1	C1CTLR1	X000 00XXb[1]
0252h	CAN1 Sleep Control Register	C1SLPR	XXXX XXX0b[1]
0253h			
0254h	CAN1 Acceptance Filter Support Register	C1AFS	0000 0000b[1]
0255h			0000 0001b[1]
0256h			
0257h			
0258h			
0259h			
025Ah			
025Bh			
025Ch			
025Dh			
025Eh			
025Fh			
0260h	CAN1 Message Slot Buffer 0 Standard ID0	C1SLOT0_0	XXh
0261h	CAN1 Message Slot Buffer 0 Standard ID1	C1SLOT0_1	XXh
0262h	CAN1 Message Slot Buffer 0 Extended ID0	C1SLOT0_2	XXh
0263h	CAN1 Message Slot Buffer 0 Extended ID1	C1SLOT0_3	XXh
0264h	CAN1 Message Slot Buffer 0 Extended ID2	C1SLOT0_4	XXh
0265h	CAN1 Message Slot Buffer 0 Data Length Code	C1SLOT0_5	XXh
0266h	CAN1 Message Slot Buffer 0 Data 0	C1SLOT0_6	XXh
0267h	CAN1 Message Slot Buffer 0 Data 1	C1SLOT0_7	XXh
0268h	CAN1 Message Slot Buffer 0 Data 2	C1SLOT0_8	XXh
0269h	CAN1 Message Slot Buffer 0 Data 3	C1SLOT0_9	XXh
026Ah	CAN1 Message Slot Buffer 0 Data 4	C1SLOT0_10	XXh
026Bh	CAN1 Message Slot Buffer 0 Data 5	C1SLOT0_11	XXh
026Ch	CAN1 Message Slot Buffer 0 Data 6	C1SLOT0_12	XXh
026Dh	CAN1 Message Slot Buffer 0 Data 7	C1SLOT0_13	XXh
026Eh	CAN1 Message Slot Buffer 0 Time Stamp High-Order	C1SLOT0_14	XXh
026Fh	CAN1 Message Slot Buffer 0 Time Stamp Low-Order	C1SLOT0_15	XXh
0270h	CAN1 Message Slot Buffer 1 Standard ID0	C1SLOT1_0	XXh
0271h	CAN1 Message Slot Buffer 1 Standard ID1	C1SLOT1_1	XXh
0272h	CAN1 Message Slot Buffer 1 Extended ID0	C1SLOT1_2	XXh
0273h	CAN1 Message Slot Buffer 1 Extended ID1	C1SLOT1_3	XXh
0274h	CAN1 Message Slot Buffer 1 Extended ID2	C1SLOT1_4	XXh
0275h	CAN1 Message Slot Buffer 1 Data Length Code	C1SLOT1_5	XXh
0276h	CAN1 Message Slot Buffer 1 Data 0	C1SLOT1_6	XXh
0277h	CAN1 Message Slot Buffer 1 Data 1	C1SLOT1_7	XXh
0278h	CAN1 Message Slot Buffer 1 Data 2	C1SLOT1_8	XXh
0279h	CAN1 Message Slot Buffer 1 Data 3	C1SLOT1_9	XXh
027Ah	CAN1 Message Slot Buffer 1 Data 4	C1SLOT1_10	XXh
027Bh	CAN1 Message Slot Buffer 1 Data 5	C1SLOT1_11	XXh
027Ch	CAN1 Message Slot Buffer 1 Data 6	C1SLOT1_12	XXh
027Dh	CAN1 Message Slot Buffer 1 Data 7	C1SLOT1_13	XXh
027Eh	CAN1 Message Slot Buffer 1 Time Stamp High-Order	C1SLOT1_14	XXh
027Fh	CAN1 Message Slot Buffer 1 Time Stamp Low-Order	C1SLOT1_15	XXh

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:
1. Values are obtained by setting the SLEEP bit in the C1SLPR register to “1” (sleep mode exited) after reset and supplying a clock to the CAN module.
2. The CAN-associated registers (allocated in addresses 01E0h to 02BFh) cannot be used in M32C/87B. In M32C/87A, only CAN0-associated registers can be used.
3. Set the PM13 bit in the PM1 register to 1 (2 wait states for SFR area) before accessing the CAN-associated registers.

Table 4.12 SFR Address Map (12/20)

Address	Register(3)(4)	Symbol	After Reset
0280h	CAN1 Control Register 0	C1CTLR0	XX01 0X01b(2)
0281h			XXXX 0000b(2)
0282h	CAN1 Status Register	C1STR	0000 0000b(2)
0283h			X000 0X01b(2)
0284h	CAN1 Extended ID Register	C1IDR	0000h(2)
0285h			
0286h	CAN1 Configuration Register	C1CONR	0000 XXXXb(2)
0287h			0000 0000b(2)
0288h	CAN1 Time Stamp Register	C1TSR	0000h(2)
0289h			
028Ah	CAN1 Transmit Error Count Register	C1TEC	00h(2)
028Bh	CAN1 Receive Error Count Register	C1REC	00h(2)
028Ch	CAN1 Slot Interrupt Status Register	C1SISTR	0000h(2)
028Dh			
028Eh			
028Fh			
0290h	CAN1 Slot Interrupt Mask Register	C1SIMKR	0000h(2)
0291h			
0292h			
0293h			
0294h	CAN1 Error Interrupt Mask Register	C1EIMKR	XXXX X000b(2)
0295h	CAN1 Error Interrupt Status Register	C1EISTR	XXXX X000b(2)
0296h	CAN1 Error Source Register	C1EFR	00h(2)
0297h	CAN1 Baud Rate Prescaler	C1BRP	0000 0001b(2)
0298h			
0299h	CAN1 Mode Register	C1MDR	XXXX XX00b(2)
029Ah			
029Bh			
029Ch			
029Dh			
029Eh			
029Fh			
02A0h	CAN1 Single Shot Control Register	C1SSCTLR	0000h(1)(2)
02A1h			
02A2h			
02A3h			
02A4h	CAN1 Single Shot Status Register	C1SSSTR	0000h(1)(2)
02A5h			
02A6h			
02A7h			
02A8h	CAN1 Global Mask Register Standard ID0	C1GMR0	XXX0 0000b(1)(2)
02A9h	CAN1 Global Mask Register Standard ID1	C1GMR1	XX00 0000b(1)(2)
02AAh	CAN1 Global Mask Register Extended ID0	C1GMR2	XXXX 0000b(1)(2)
02ABh	CAN1 Global Mask Register Extended ID1	C1GMR3	00h(1)(2)
02ACh	CAN1 Global Mask Register Extended ID2	C1GMR4	XX00 0000b(1)(2)
02ADh			
02AEh			
02AFh			

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:
1. The BANKSEL bit in the C0CTLR1 register can switch functions for addresses 02A0h to 02BFh.
2. Values are obtained by setting the SLEEP bit in the C1SLPR register to “1” (sleep mode exited) after reset and supplying a clock to the CAN module.
3. The CAN-associated registers (allocated in addresses 01E0h to 02BFh) cannot be used in M32C/87B. In M32C/87A, only CAN0-associated registers can be used.
4. Set the PM13 bit in the PM1 register to 1 (2 wait states for SFR area) before accessing the CAN-associated registers.

Table 4.13 SFR Address Map (13/20)

Address	Register(3)(4)	Symbol	After Reset
02B0h	CAN1 Message Slot 0 Control Register / CAN1 Local Mask Register A Standard ID0	C1MCTL0 / C1LMAR0	0000 0000b(1)(2)/ XXX0 0000b(1)(2)
02B1h	CAN1 Message Slot 1 Control Register / CAN1 Local Mask Register A Standard ID1	C1MCTL1 / C1LMAR1	0000 0000b(1)(2)/ XX00 0000b(1)(2)
02B2h	CAN1 Message Slot 2 Control Register / CAN1 Local Mask Register A Extended ID0	C1MCTL2 / C1LMAR2	0000 0000b(1)(2)/ XXXX 0000b(1)(2)
02B3h	CAN1 Message Slot 3 Control Register / CAN1 Local Mask Register A Extended ID1	C1MCTL3 / C1LMAR3	00h(1)(2)/ 00h(1)(2)
02B4h	CAN1 Message Slot 4 Control Register / CAN1 Local Mask Register A Extended ID2	C1MCTL4 / C1LMAR4	0000 0000b(1)(2)/ XX00 0000b(1)(2)
02B5h	CAN1 Message Slot 5 Control Register	C1MCTL5	00h(1)(2)
02B6h	CAN1 Message Slot 6 Control Register	C1MCTL6	00h(1)(2)
02B7h	CAN1 Message Slot 7 Control Register	C1MCTL7	00h(1)(2)
02B8h	CAN1 Message Slot 8 Control Register / CAN1 Local Mask Register B Standard ID0	C1MCTL8 / C1LMBR0	0000 0000b(1)(2)/ XXX0 0000b(1)(2)
02B9h	CAN1 Message Slot 9 Control Register / CAN1 Local Mask Register B Standard ID1	C1MCTL9 / C1LMBR1	0000 0000b(1)(2)/ XX00 0000b(1)(2)
02BAh	CAN1 Message Slot 10 Control Register / CAN1 Local Mask Register B Extended ID0	C1MCTL10 / C1LMBR2	0000 0000b(1)(2)/ XXXX 0000b(1)(2)
02BBh	CAN1 Message Slot 11 Control Register / CAN1 Local Mask Register B Extended ID1	C1MCTL11 / C1LMBR3	00h(1)(2)/ 00h(1)(2)
02BCh	CAN1 Message Slot 12 Control Register / CAN1 Local Mask Register B Extended ID2	C1MCTL12 / C1LMBR4	0000 0000b(1)(2)/ XX00 0000b(1)(2)
02BDh	CAN1 Message Slot 13 Control Register	C1MCTL13	00h(1)(2)
02BEh	CAN1 Message Slot 14 Control Register	C1MCTL14	00h(1)(2)
02BFh	CAN1 Message Slot 15 Control Register	C1MCTL15	00h(1)(2)

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTES:
1. The BANKSEL bit in the C1CTLR1 register can switch functions for addresses 02A0h to 02BFh.
2. Values are obtained by setting the SLEEP bit in the C1SLPR register to “1” (sleep mode exited) after reset and supplying a clock to the CAN module.
3. The CAN-associated registers (allocated in addresses 01E0h to 02BFh) cannot be used in M32C/87B. In M32C/87A, only CAN0-associated registers can be used.
4. Set the PM13 bit in the PM1 register to 1 (2 wait states for SFR area) before accessing the CAN-associated registers.

Table 4.14 SFR Address Map (14/20)

Address	Register	Symbol	After Reset
02C0h 02C1h	X0 Register, Y0 Register	X0R, Y0R	XXXXh
02C2h 02C3h	X1 Register, Y1 Register	X1R, Y1R	XXXXh
02C4h 02C5h	X2 Register, Y2 Register	X2R, Y2R	XXXXh
02C6h 02C7h	X3 Register, Y3 Register	X3R, Y3R	XXXXh
02C8h 02C9h	X4 Register, Y4 Register	X4R, Y4R	XXXXh
02CAh 02CBh	X5 Register, Y5 Register	X5R, Y5R	XXXXh
02CCh 02CDh	X6 Register, Y6 Register	X6R, Y6R	XXXXh
02CEh 02CFh	X7 Register, Y7 Register	X7R, Y7R	XXXXh
02D0h 02D1h	X8 Register, Y8 Register	X8R, Y8R	XXXXh
02D2h 02D3h	X9 Register, Y9 Register	X9R, Y9R	XXXXh
02D4h 02D5h	X10 Register, Y10 Register	X10R, Y10R	XXXXh
02D6h 02D7h	X11 Register, Y11 Register	X11R, Y11R	XXXXh
02D8h 02D9h	X12 Register, Y12 Register	X12R, Y12R	XXXXh
02DAh 02DBh	X13 Register, Y13 Register	X13R, Y13R	XXXXh
02DCh 02DDh	X14 Register, Y14 Register	X14R, Y14R	XXXXh
02DEh 02DFh	X15 Register, Y15 Register	X15R, Y15R	XXXXh
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 02EBh	UART1 Transmit Buffer Register	U1TB	XXXXh
02ECh	UART1 Transmit/Receive Control Register 0	U1C0	0000 1000b
02EDh	UART1 Transmit/Receive Control Register 1	U1C1	0000 0010b
02EEh 02EFh	UART1 Receive Buffer Register	U1RB	XXXXh

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

Table 4.15 SFR Address Map (15/20)

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 02FBh	UART4 Transmit Buffer Register	U4TB	XXXXh
02FCh	UART4 Transmit/Receive Control Register 0	U4C0	0000 1000b
02FDh	UART4 Transmit/Receive Control Register 1	U4C1	0000 0010b
02FEh 02FFh	UART4 Receive Buffer Register	U4RB	XXXXh
0300h	Timer B3, B4, B5 Count Start Register	TBSR	000X XXXXb
0301h			
0302h 0303h	Timer A11 Register	TA11	XXXXh
0304h 0305h	Timer A21 Register	TA21	XXXXh
0306h 0307h	Timer A41 Register	TA41	XXXXh
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 0311h	Timer B3 Register	TB3	XXXXh
0312h 0313h	Timer B4 Register	TB4	XXXXh
0314h 0315h	Timer B5 Register	TB5	XXXXh
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	External Interrupt Source Select Register 1(1)	IFSRA	00h
031Fh	External Interrupt Source Select Register	IFSR	00h

X: Undefined
Blank spaces are all reserved. No access is allowed.
NOTE:
1. The IFSRA register is included in the 144-pin package only.

Table 4.16 SFR Address Map (16/20)

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			
044Ch	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 4.17 SFR Address Map (17/20)

Address	Register	Symbol	After Reset
0350h 0351h	Timer B0 Register	TB0	XXXXh
0352h 0353h	Timer B1 Register	TB1	XXXXh
0354h 0355h	Timer B2 Register	TB2	XXXXh
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 036Bh	UART0 Transmit Buffer Register	U0TB	XXXXh
036Ch	UART0 Transmit/Receive Control Register 0	U0C0	0000 1000b
036Dh	UART0 Transmit/Receive Control Register 1	U0C1	0000 0010b
036Eh 036Fh	UART0 Receive Buffer Register	U0RB	XXXXh
0370h			
0371h			
0372h	IrDA Control Register	IRCON	X000 0000b
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 037Dh	CRC Data Register	CRCD	XXXXh
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 4.18 SFR Address Map (18/20)

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	D/A Control Register 1	DACON1	XXXX 0000b
039Eh			
039Fh			

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

Table 4.19 SFR Address Map (19/20)

Address	Register	Symbol	After Reset
03A0h	Function Select Register A8(1)	PS8	X000 0000b
03A1h	Function Select Register A9(1)	PS9	00h
03A2h			
03A3h	Function Select Register B9(1)	PSL9	XXX0 XX00b
03A4h	Function Select Register E2	PSE2	XXXX XX0Xb
03A5h			
03A6h			
03A7h	Function Select Register D1	PSD1	00X0 XX00b
03A8h	Function Select Register D2	PSD2	XXXX XX0Xb
03A9h			
03AAh	Function Select Register C6(1)	PSC6	XXXX 0X00b
03ABh	Function Select Register E1	PSE1	00XX XX00b
03ACh	Function Select Register C2	PSC2	XXXX X00Xb
03ADh	Function Select Register C3	PSC3	X0XX XXXXb
03AEh			
03AFh	Function Select Register C	PSC	00h
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	Function Select Register A4	PS4	00h
03B9h	Function Select Register A5(1)	PS5	XXX0 0000b
03BAh			
03BBh	Function Select Register B5(1)	PSL5	XXX0 0000b
03BCh	Function Select Register A6(1)	PS6	00h
03BDh	Function Select Register A7(1)	PS7	00h
03BEh	Function Select Register B6(1)	PSL6	00h
03BFh	Function Select Register B7(1)	PSL7	00h
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 4.20 SFR Address Map (20/20)

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 1, hardware reset 2 (Vdet3 detection function), software reset and watchdog timer reset are implemented to reset the MCU.

5.1 Hardware Reset 1

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 WDC5 bit in the WDC register and the internal RAM are not reset by hardware reset 1. 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.

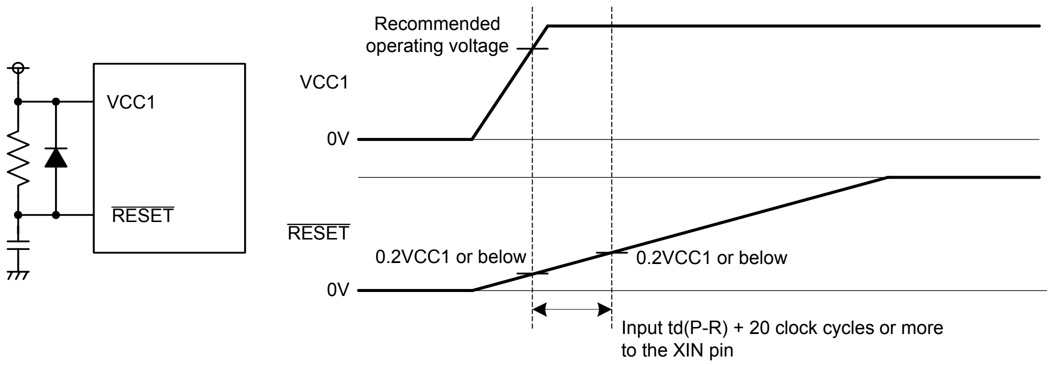
Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows a reset sequence. Table 5.1 lists pin states while the RESET pin is held "L".

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.



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

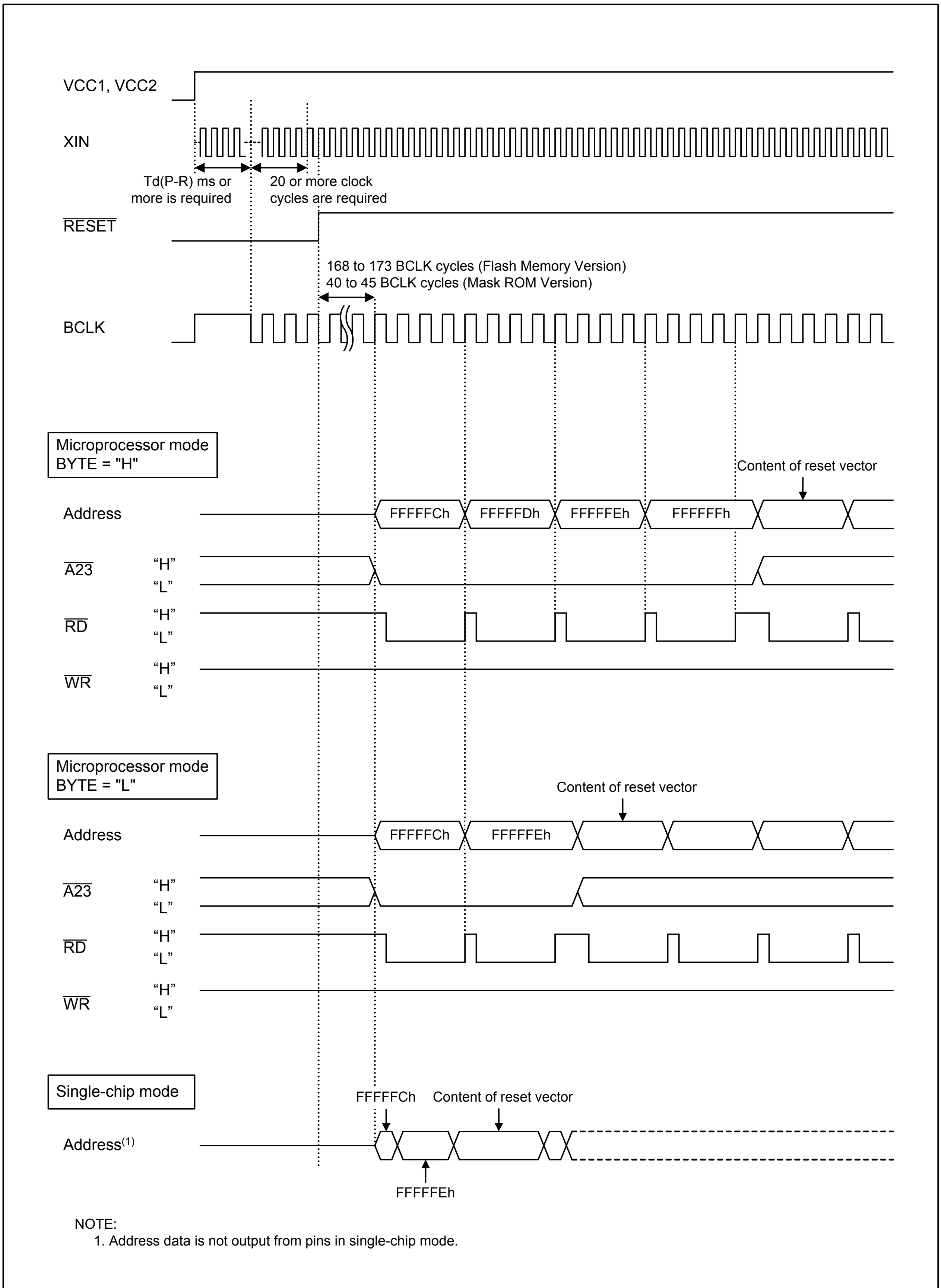


Figure 5.2 Reset Sequence

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. $\overline{\text{EPM}}$ (P5_5) must be "H" in the flash memory version.

5.2 Hardware Reset 2 (Vdet3 detection function)

Pins, CPU, and SFRs are reset by the Vdet3 detection function, when the voltage applied to the VCC1 pin drops to Vdet3 (V) or below. The states of the pins, CPU, and SFRs after reset are the same as the hardware reset 1. Refer to **6. Power Supply Voltage Detection Function** for details on Vdet3 detection function.

5.3 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, the WDC5 bit in the WDC register, and the TCSPR register.
Processor mode remains unchanged since bits PM01 and PM00 are not reset.

5.4 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, the WDC5 bit in the WDC register, and the TCSPR register.
Processor mode remains unchanged since bits PM01 and PM00 are not reset.

5.5 Internal Registers

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

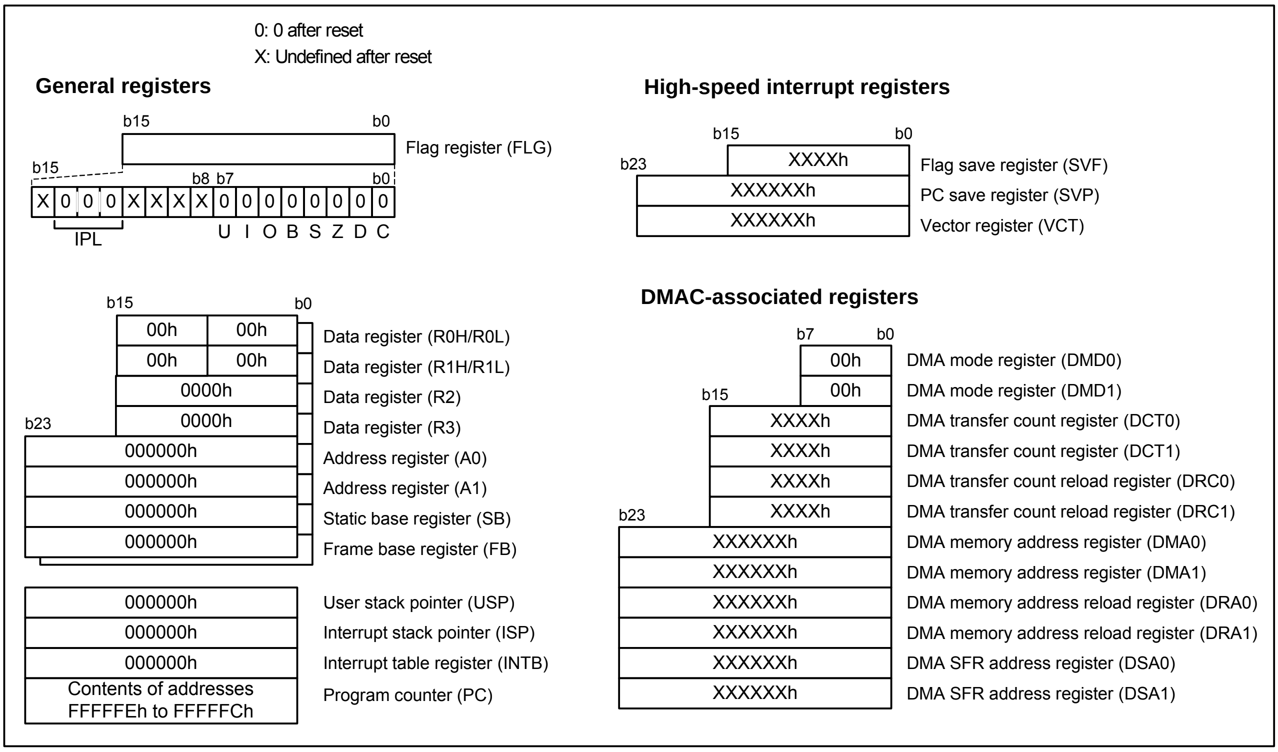
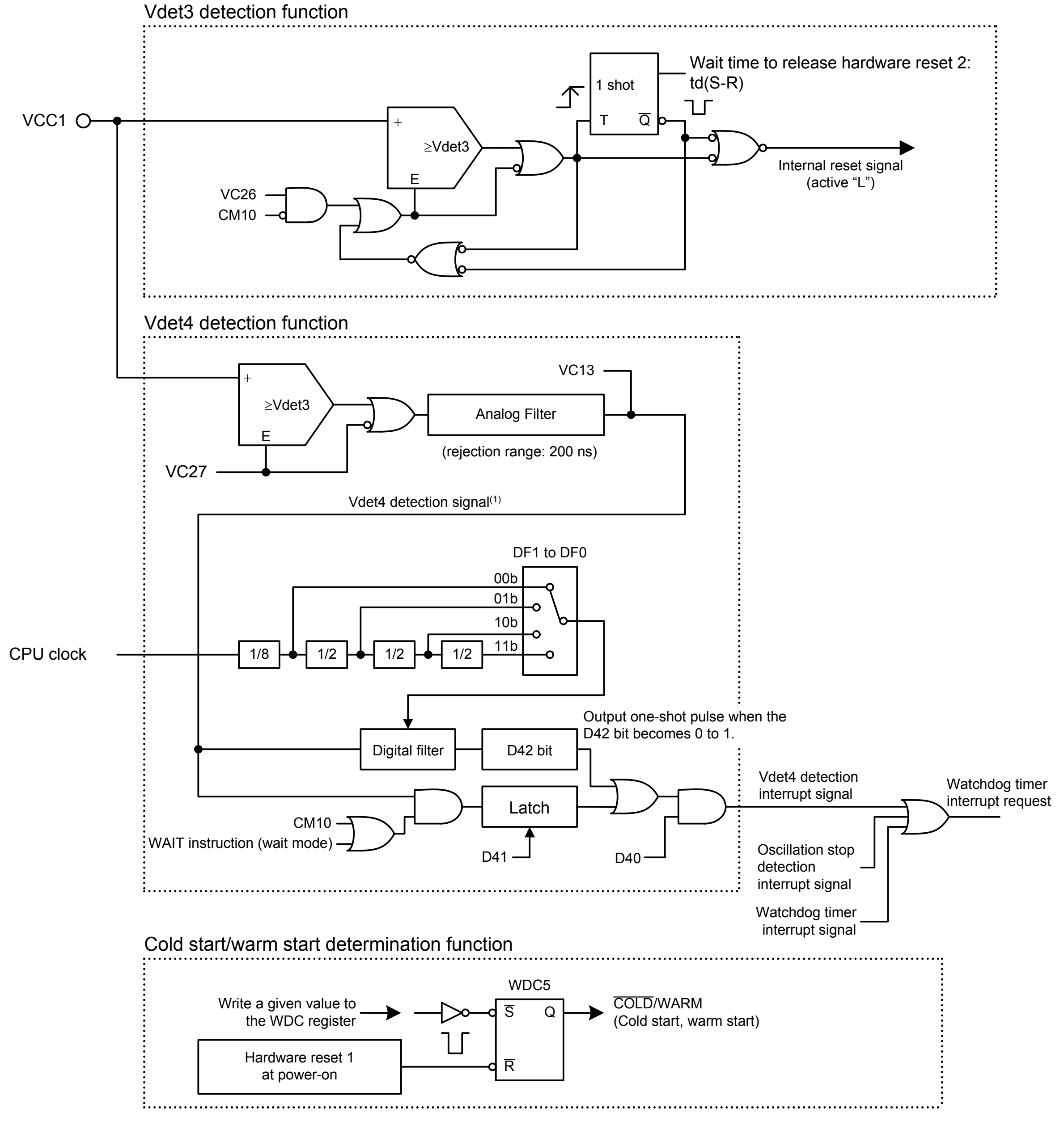


Figure 5.3 CPU Register States after Reset

6. Power Supply Voltage Detection Function

The power supply voltage detection function has the Vdet3 detection function, Vdet4 detection function, and cold start/warm start determination function. The Vdet3 detection function and Vdet4 detection function detect the changes in voltage and trigger the events. The cold start/warm start determination function determines whether the MCU is reset at power-on or reset while running.
The power supply voltage detection function is available only with VCC1 = 4.2V to 5.5V standard.
Figure 6.1 shows a block diagram of the voltage detection circuit. Figures 6.2 to 6.4 show registers associated with the voltage detection function.



CM10: bit in the CM1 register
VC13: bit in the VCR1 register
VC26, VC27: bits in the VCR2 register
DF1 and DF0, D40, D41, D42: bits in the D4INT register
WDC5: bit in the WDC register

NOTE:
1. When the VC27 bit in the VCR2 register is set to 0 (Vdet4 detection function not used), the Vdet4 detection signal becomes "H".

Figure 6.1 Power Supply Voltage Detection Function Block Diagram

Voltage Detection Register 1

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

Symbol: VCR1 — Address: 001Bh — After Reset: 0000 1000b

Bit Symbol	Bit Name	Function	RW
– (b2-b0)	Reserved bits	Set to 0	RW
VC13	Voltage change monitor flag(1)	0: VCC1 < Vdet4 1: VCC1 $\geq$ Vdet4	RO
– (b7-b4)	Reserved bits	Set to 0	RW

NOTE:
1. The VC13 bit is enabled when the VC27 bit in the VCR2 register is set to 1 (Vdet4 detection function used). The VC13 bit becomes 1 when the VC27 bit is set to 0 (Vdet4 detection function not used).

Voltage Detection Register 2(1)

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

Symbol: VCR2 — Address: 0017h — After Reset: 00h

Bit Symbol	Bit Name	Function	RW
– (b5-b0)	Reserved bits	Set to 0	RW
VC26	Vdet3 detection function select bit(2, 4, 5)	0: Vdet3 detection function not used 1: Vdet3 detection function used	RW
VC27	Vdet4 detection function select bit(3, 4)	0: Vdet4 detection function not used 1: Vdet4 detection function used	RW

NOTES:
1. Set the VCR2 register after the PRC3 bit in the PRCR register is set to 1 (write enable).
2. To use the hardware reset 2 (Vdet3 detection function), set the VC26 bit to 1.
3. To use the Vdet4 detection function, set the VC27 bit to 1 and the D40 bit in the D4INT register to 1 (Vdet4 detection interrupt used). The VC13 bit in the VCR1 register and the D42 bit in the D4INT register are enabled when the VC27 bit is set to 1.
4. After the VC26 or VC27 bit is set to 1, the detection circuit waits for td(E-A) to elapse before starting operation.
5. The VC26 bit is disabled when the MCU is in stop mode. (The hardware reset 2 is not performed even if the voltage applied to the VCC1 pin drops below Vdet3.)

Figure 6.2 VCR1 Register, VCR2 Register

Vdet4 Detection Interrupt Register[1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
D4INT	002Fh	XX00 0000b

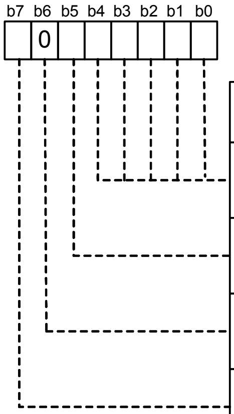
Bit Symbol	Bit Name	Function	RW
D40	Vdet4 detection interrupt enable bit[2]	0: Vdet4 detection interrupt disabled 1: Vdet4 detection interrupt enabled	RW
D41	Wait mode/Stop mode exit control bit[3]	0: Vdet4 detection interrupt is not used to exit wait/stop mode 1: Vdet4 detection interrupt is used to exit wait/stop mode	RW
D42	Voltage change detect flag[4, 5]	0: Not detected 1: Voltage has crossed Vdet4	RW
D43	WDT underflow detect flag[5]	0: Not detected 1: Detected	RW
DF0	Sampling clock select bits	b5 b4 0 0: CPU clock divided-by-8 0 1: CPU clock divided-by-16 1 0: CPU clock divided-by-32 1 1: CPU clock divided-by-64	RW
DF1			RW
– (b7-b6)	Unimplemented. Read as undefined value.		–

NOTES:

1. Set the D4INT register after the PRC3 bit in the PRCR register is set to 1 (write enable).
2. Use the following procedure to set the D40 bit to 1:
   (1) Set the VC27 bit in the VCR2 register to 1
   (2) Wait for td(E-A) before the voltage detection circuit starts operating
   (3) Wait for required sampling time (See **Table "Sampling Period"**)
   (4) Set the D40 bit to 1
3. If the Vdet4 detection interrupt has been used to exit wait mode or stop mode, set the D41 bit to 0 and then set it to 1 to use the Vdet4 detection interrupt again to exit these modes.
4. The D42 bit is enabled when the VC27 bit is set to 1 (Vdet4 detection function used). The D42 bit becomes 0 when the VC27 bit is set to 0 (Vdet4 detection function not used).
5. The D43 bit can be set to 0 by a program. Writing a 1 has no effect.

Figure 6.3 D4INT Register

Watchdog Timer Control Register



Symbol	Address	After Reset
WDC	000Fh	00XX XXXXb

Bit Symbol	Bit Name	Function	RW
– (b4-b0)	High-order bits of watchdog timer		RO
WDC5	Cold start/warm start determine flag(1)	0: Cold start 1: Warm start	RW
– (b6)	Reserved bit	Set to 0	RW
WDC7	Prescaler select bit	0: Divide-by-16 1: Divide-by-128	RW

NOTE:
1. The WDC5 bit is 0 after power-on. It can be set to 1 only by a program. The bit becomes 1 by writing either a 0 or 1. The bit remains a value set before reset, even after reset has been performed.

Figure 6.4 WDC Register

6.1 Vdet3 Detection Function

The hardware reset 2 is performed if the voltage applied to the VCC1 pin drops to Vdet3 (V) or below.
Set the VC26 bit in the VCR2 register to 1 to use this Vdet3 detection function.
When the hardware reset 2 occurs, ports and I/O pins for peripheral functions are reset. The CPU and SFRs are reset when td(S-R) elapses after the voltage applied to the VCC1 pin reaches Vdet3r (V) or above. Then, the MCU executes a program in an address indicated by the reset vector. The states of pins and SFRs after reset are the same as the hardware reset 1.
Use the Vdet3 detection function while operating at or above Vdet3s. If the applied voltage drops below Vdet3s, perform the hardware reset 1 (refer to **5.1.2** Power-on Reset). The Vdet3 detection function cannot be used while the MCU is in stop mode.
Figure 6.5 shows a Vdet3 detection function operation example.

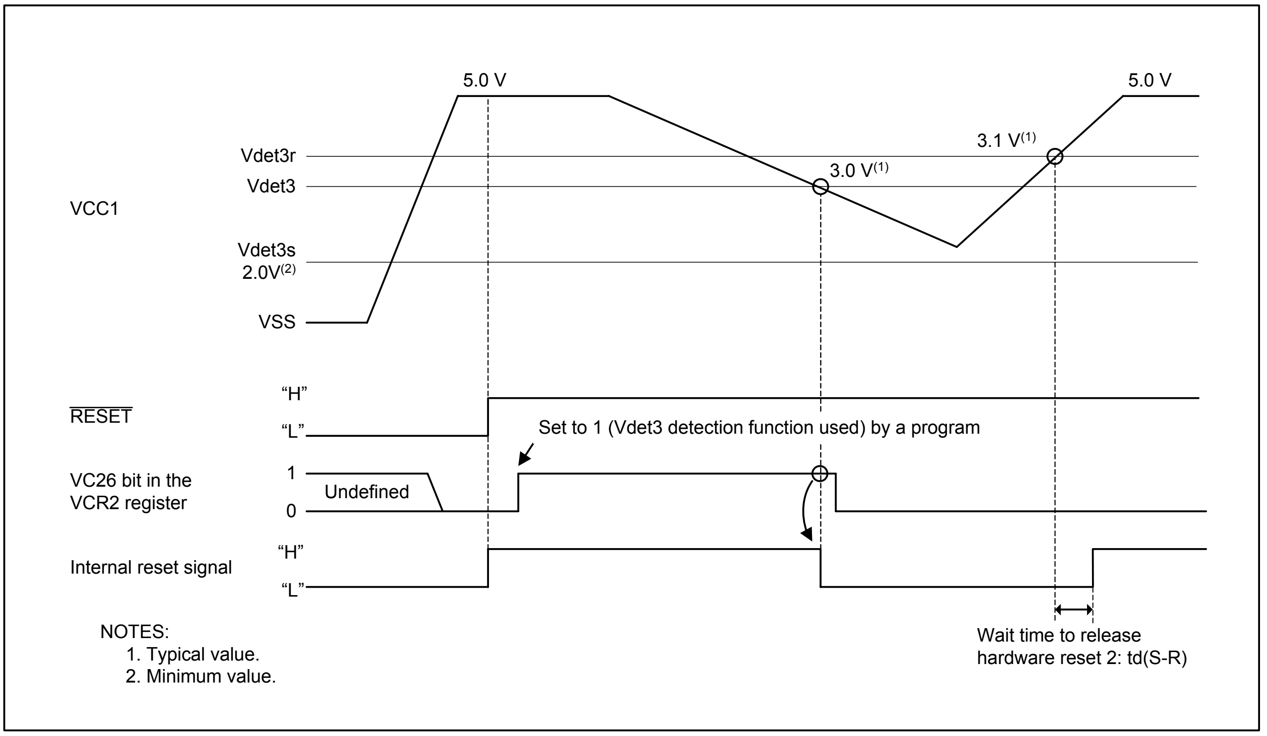


Figure 6.5 Vdet3 Detection Function Operation Example

6.2 Vdet4 Detection Function

Vdet4 detection interrupt is generated if the voltage applied to the VCC1 pin crosses the Vdet4 (V) level, either by dropping below or by rising above Vdet4. Set the VC27 bit in the VCR2 register to 1 (Vdet4 detection function used) and the D40 bit in the D4INT register to 1 (Vdet4 detection interrupt enabled) to use the Vdet4 detection function.
The D42 bit becomes 1 (voltage has crossed Vdet4) as soon as the applied voltage crosses Vdet4. When the D42 bit changes from 0 to 1, a Vdet4 detection interrupt request is generated. The D42 bit does not become 0 automatically when the interrupt is acknowledged. Set it to 0 (not detected) by a program. Whether the voltage has dropped below Vdet4 or risen above Vdet4 can be determined by reading the VC13 bit in the VCR1 register.
Set the D41 bit in the D4INT register to 1 to use the Vdet4 detection interrupt to exit wait mode or stop mode. The MCU exits wait mode or stop mode if the Vdet4 detection signal is generated even if the D42 bit is 1.
The Vdet4 detection interrupt shares the same interrupt vector with watchdog timer interrupt and oscillation stop detection interrupt. When using the Vdet4 detection interrupt simultaneously with these interrupts, determine whether the Vdet4 detection interrupt is generated by reading the D42 bit in the interrupt routine.
Table 6.1 shows conditions to generate Vdet4 detection interrupt request. Figure 6.6 shows a Vdet4 detection function operation example.

Bits DF1 and DF0 in the D4INT register determine the sampling clock which is used to detects if the voltage applied to the VCC1 pin has crossed Vdet4. Table 6.2 shows the sampling periods.

Table 6.1 Conditions to Generate Vdet4 Detection Interrupt Request

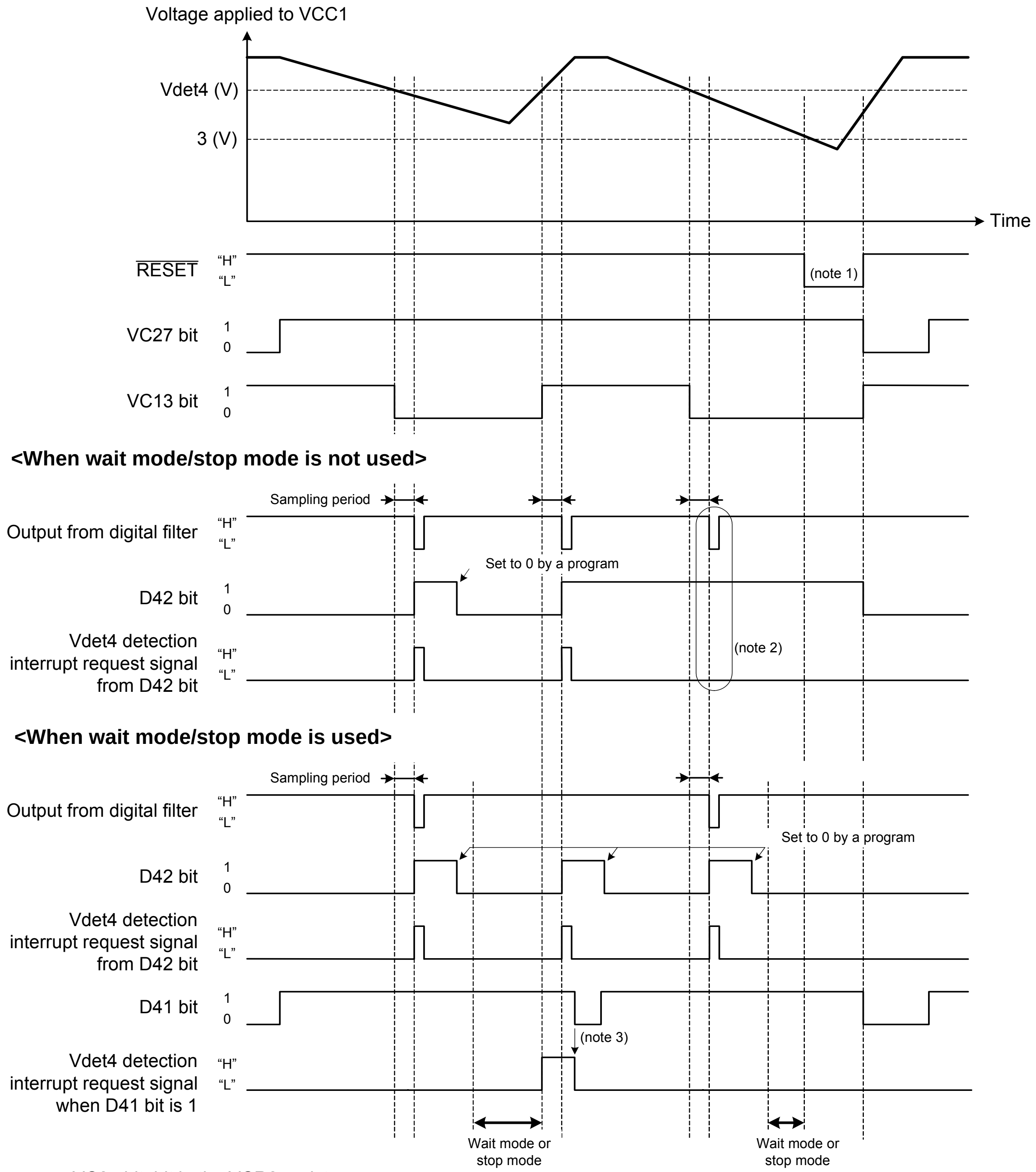
Operating Mode	VC27 Bit	D40 Bit	D41 Bit	D42 Bit(1)	VC13 Bit(2)
CPU operating mode(3)	1	1	0 or 1	0 to 1	0 to 1 1 to 0
Wait mode, Stop mode(4)			1	0 or 1	0 to 1

NOTES:
1. Set to 0 by a program before generating an interrupt.
2. An interrupt request is generated when the sampling period elapses after the value of the VC13 bit is changed. See **Figure 6.6 Vdet4 Detection Function Operation Example** for details.
3. CPU operating mode includes main clock mode, main clock direct mode, PLL mode, low speed mode, low-power consumption mode, on-chip oscillator mode, on-chip oscillator low-power consumption mode. (Refer to **9. Clock Generation Circuits**.)
4. Refer to **6.2.1 Usage Notes on Vdet4 Detection Interrupt**.

Table 6.2 Sampling Periods

CPU Clock (MHz)	Sampling Clock (μs)			
	Divided-by-8	Divided-by-16	Divided-by-32	Divided-by-64
16	3.0	6.0	12.0	24.0
24	2.0	4.0	8.0	16.0

NOTE:
1. Set the CPU clock 24 MHz or lower to use the voltage detection function.



VC27 bit: bit in the VCR2 register
VC13 bit: bit in the VCR1 register
VC41 bit, VC42 bit: bits in the D4INT register

NOTES:
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 output voltage from the main voltage regulator and the main clock oscillation stabilize, apply an "H" to the RESET pin.
2. When the D42 bit is set to 1, the Vdet4 detection interrupt request signal is not generated even if the Vdet4 detection signal is output from the digital filter.
3. If the Vdet4 detection interrupt has been used to exit wait mode or stop mode, set the D41 bit to 0 and then set it back to 1 to use the Vdet4 detection interrupt again to exit wait/stop mode.

Figure 6.6 Vdet4 Detection Function Operation Example

6.2.1 Usage Notes on Vdet4 Detection Interrupt

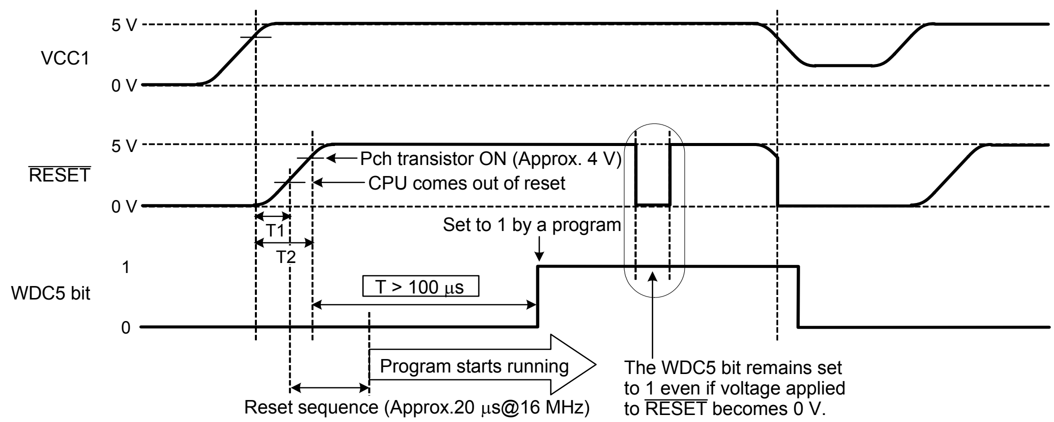
When all the conditions below are met, the Vdet4 detection interrupt is generated and the MCU exits wait mode as soon as the WAIT instruction is executed or exits stop mode as soon as the CM10 bit in the CM1 register is set to 1 (all clocks stopped).

- the VC27 bit in the VCR2 register is set to 1 (Vdet4 detection function used)
- the D40 bit in the D4INT register is set to 1 (Vdet4 detection interrupt enabled)
- the D41 bit in the D4INT register is set to 1 (Vdet4 detection interrupt is used to exit wait/stop mode)
- the voltage applied to the VCC1 pin is Vdet4 or above (the VC13 bit in the VCR1 register is 1)

Execute the WAIT instruction or set the CM10 bit to 1 (all clocks stop) while the VC13 bit is 0 (VCC1 < Vdet4), if the MCU is configured to enter wait/stop mode when voltage applied to the VCC1 pin drops Vdet4 or below and to exit wait/stop mode when the voltage applied rises to Vdet4 or above.
If the Vdet4 detection interrupt has been used to exit wait mode or stop mode, set the D41 bit to 0 and then set it back to 1 to use the Vdet4 detection interrupt again to exit wait/stop mode.

6.3 Cold Start/Warm Start Determination Function

The WDC5 bit in the WDC register determines whether it is a reset process when power-on (cold start) or a reset process when the $\overline{\text{RESET}}$ signal is input during MCU running (warm start). Default value of the WDC5 bit is 0 (cold start) when power-on, and the bit is set to 1 (warm start) by writing given values to the WDC register. The WDC5 bit does not become 0 even if the hardware reset 1, hardware reset 2, software reset, or watchdog timer reset is performed.
Figure 6.7 shows an example of cold start/warm start determination function operation.



NOTE:
1. If the time difference between T1 and T2 is greater, it may take longer to set the WDC5 bit to 1.

Figure 6.7 Cold Start/Warm Start Determination Function Operation

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 **26. 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 CNVSS pin	Input to $\overline{\text{EPM}}$(P5_5)	Memory Type	Processor Mode
L	H or L	Mask ROM version Flash memory version	Single-chip mode
H	H or L	Mask ROM version	Microprocessor mode
H	H	Flash memory version	Microprocessor mode
H	L	Flash memory version	Boot mode

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)

b7	b6	b5	b4	b3	b2	b1	b0
	0						

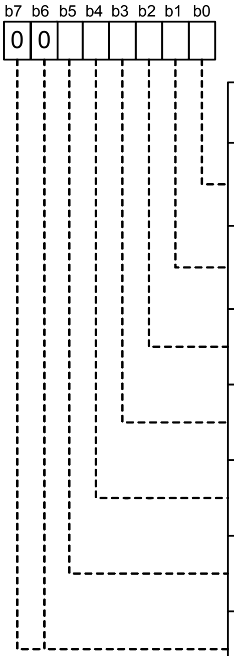
Symbol: PM0 | Address: 0004h | After Reset: 1000 0000b (CNVSS = “L”), 0000 0011b (CNVSS = “H”)

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 $\overline{A23}$ for P4_4 to P4_7) 0 1: Mode 1 (A20 for P4_4, $\overline{CS2}$ to $\overline{CS0}$ for P4_5 to P4_7) 1 0: Mode 2 (A20 and A21 for P4_4 and P4_5, $\overline{CS1}$ and $\overline{CS0}$ for P4_6 and P4_7) 1 1: Mode 3 ($\overline{CS3}$ to $\overline{CS0}$ 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(3)	RW
PM14	ALE pin select bits(2)	b5 b4 0 0: No ALE 0 1: P5_3(4) 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. Set the PM13 bit to 1 before accessing CAN-associated registers.
4. 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

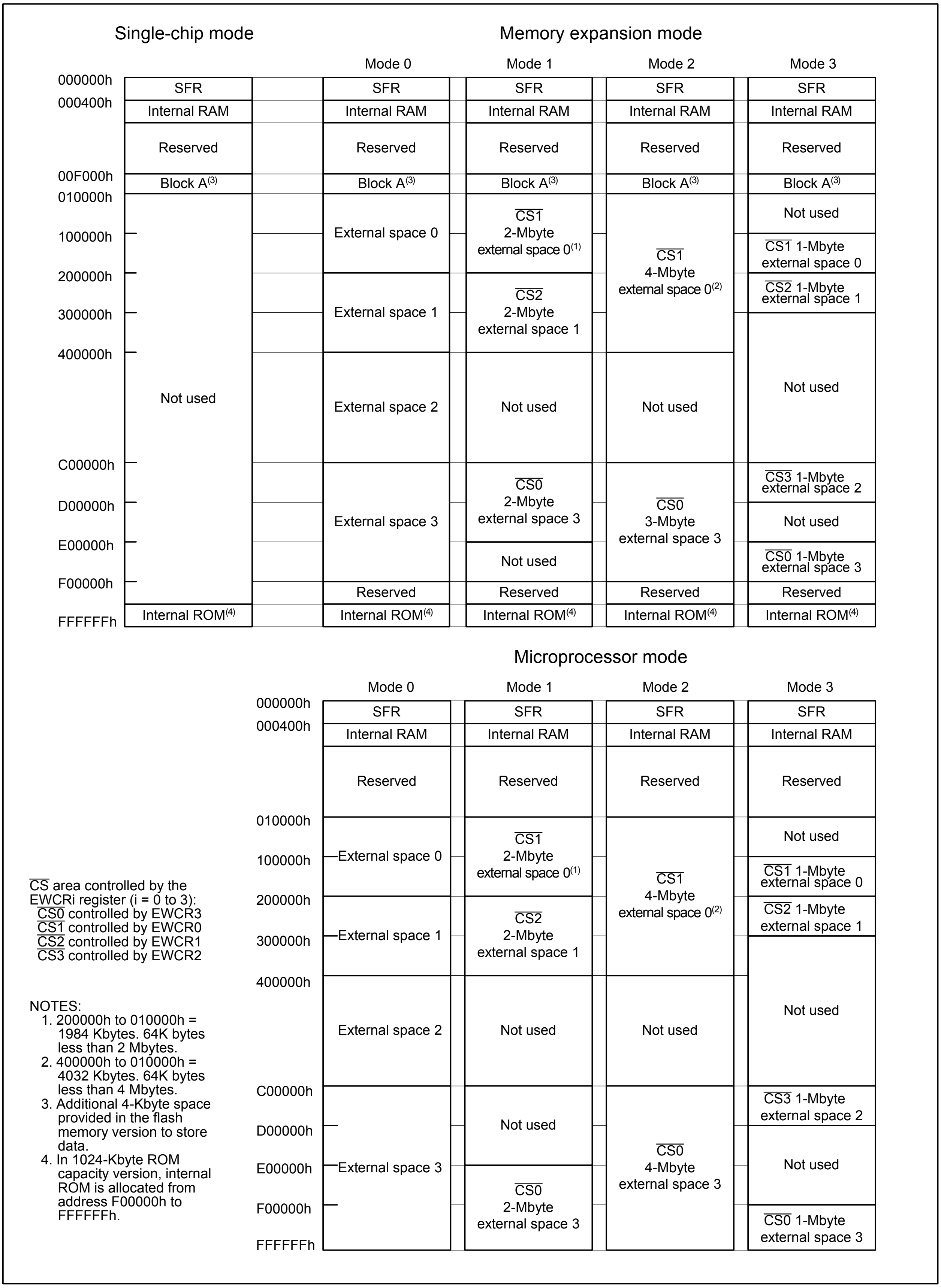


Figure 7.3 Memory Map in Each Processor Mode

8. Bus

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

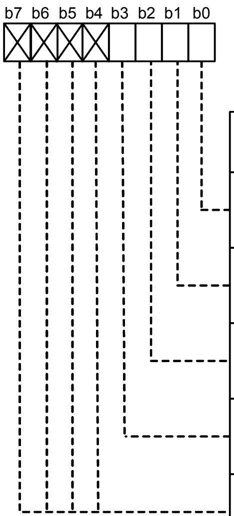
8.1 Bus Settings

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: DS
Address: 000Bh
After Reset: 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 ($\overline{CS}$ 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. The data bus in the external space 3 becomes 16 bits wide when a low-level (“L”) signal is applied to the BYTE pin after reset, 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 $\overline{CS}$ 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

<table>
<tr><th rowspan="2">PM05 and PM04 bits setting</th><th colspan="4">PM11 and PM10 Bits Setting</th></tr>
<tr><th>00b (external space mode 0</th><th>01b (external space mode 1)</th><th>10b (external space mode 2)</th><th>11b (external space mode 3)</th></tr>
<tr><td>00b (multiplexed bus not used)</td><td colspan="4">Separate bus</td></tr>
<tr><td>01b (access the $\overline{CS2}$ area using multiplexed bus)</td><td rowspan="3">Do not set to these values</td><td>$\overline{CS2}$</td><td>Do not set to this value</td><td>$\overline{CS2}$</td></tr>
<tr><td>10b (access the $\overline{CS1}$ area using multiplexed bus)</td><td>$\overline{CS1}$</td><td>$\overline{CS1}$</td><td>$\overline{CS1}$</td></tr>
<tr><td>11b (access the all $\overline{CS}$ areas using multiplexed bus)(1)</td><td>$\overline{CS0}$
$\overline{CS1}$
$\overline{CS2}$</td><td>$\overline{CS0}$
$\overline{CS1}$</td><td>$\overline{CS0}$
$\overline{CS1}$
$\overline{CS2}$
$\overline{CS3}$</td></tr>
</table>

NOTE:
1. In microprocessor mode, do not set bits PM05 and PM04 in the PM0 register to 11b (access all $\overline{CS}$ 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 $\overline{CS2}$ area using multiplexed bus) 10b (Access $\overline{CS1}$ area using multiplexed bus)		11b (Access all $\overline{CS}$ 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		$\overline{CS}$ or address bus (A20 to A22) (Refer to **8.2 Bus Control** for details)[6]					
P4_7		$\overline{CS}$ or address bus ($\overline{A23}$) (Refer to **8.2 Bus Control** for details)[6]					
P5_0 to P5_2		$\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$ outputs or $\overline{RD}$, $\overline{BHE}$, $\overline{WR}$ 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	$\overline{HLDA}$/ALE[3]					
P5_5		$\overline{HOLD}$					
P5_6		ALE[3][5]					
P5_7		$\overline{RDY}$					

NOTES:
1. Do not set bits PM05 and PM04 in the PM0 register to 11b (access all $\overline{CS}$ 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, "$\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$" or "$\overline{RD}$, $\overline{BHE}$, $\overline{WR}$".
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 required to access external devices and the bus timing. 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

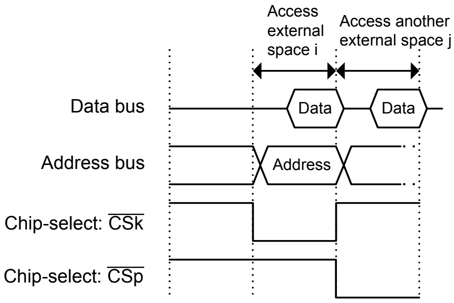
Chip-select outputs share pins with address bus, A20 to A22 and $\overline{\text{A23}}$. Bits PM11 and PM10 in the PM1 register determine the $\overline{\text{CS}}$ areas to be accessed and the number of chip-select outputs. Maximum of four chip-select outputs are provided.

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).

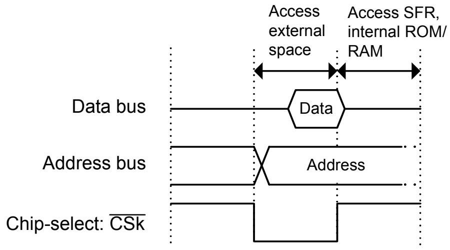
Example 1:
After accessing the external space, both address bus and chip-select output change

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.



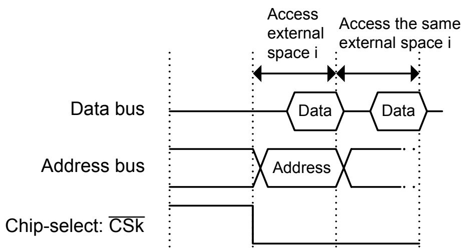
Example 2:
After accessing an external space, the chip-select output changes but the address bus does not.

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.



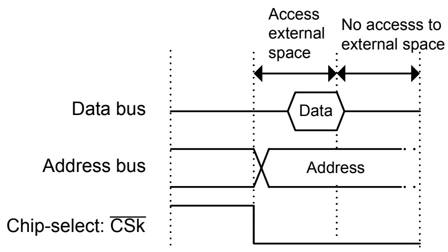
Example 3:
After accessing the external space, the address bus changes but the chip-select output does not.

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.



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.



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.

$\overline{CS1}$ outputs an "L" signal while accessing the external space 0.
$\overline{CS2}$ outputs an "L" signal while accessing the external space 1.
$\overline{CS3}$ outputs an "L" signal while accessing the external space 2.
$\overline{CS0}$ 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 "$\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$" outputs or the "$\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$" 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 ($\overline{RD}$/$\overline{WR}$/$\overline{BHE}$). 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 "$\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$" is automatically selected regardless of the PM02 bit setting. Table 8.4 lists $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ outputs. Table 8.5 list $\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$ outputs.

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

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

Data Bus Width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	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 $\overline{WR}$ output.

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

Data Bus Width	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	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. 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)

b7 b6 b5 b4 b3 b2 b1 b0

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.		–

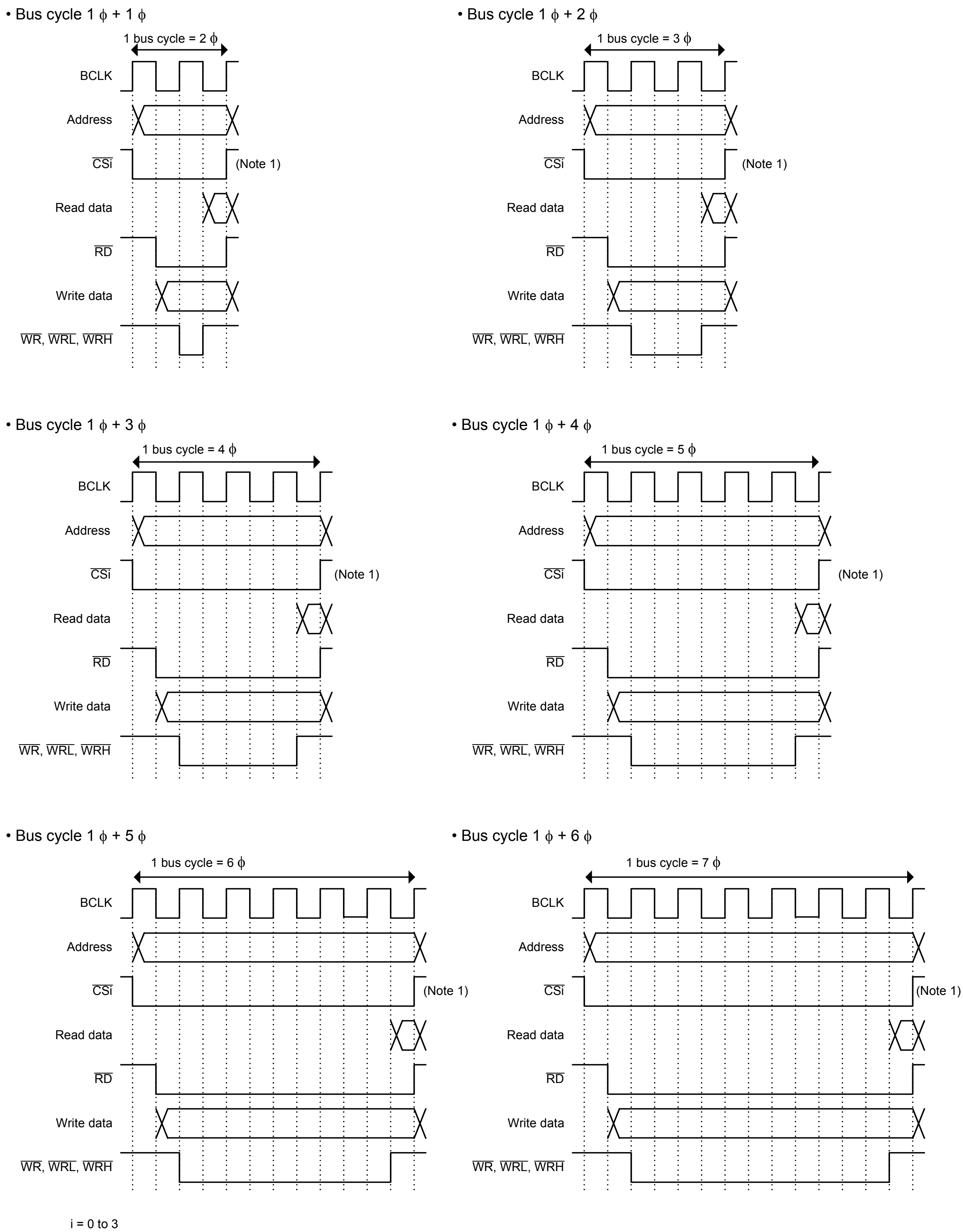
NOTES:
1. The number of BCLK cycles in the period from the beginning of the bus access until the read or write output signal becomes "L".
2. The number of BCLK cycles in the period from the read or write output signal becomes "L" until the signal changes to "H".

Figure 8.3 EWCR0 to EWCR3 Registers

Table 8.6 Software Wait State and Bus Cycle

<table>
<tr><th rowspan="2">Space</th><th rowspan="2">External Bus Status</th><th colspan="2">PM1 Register</th><th>EWCRi Register (i=0 to 3)</th><th rowspan="2">Bus Cycle</th></tr>
<tr><th>PM13 Bit(1)</th><th>PM12 Bit</th><th>Bits EWCRi4 to EWCRi0</th></tr>
<tr><td rowspan="2">SFR area</td><td rowspan="2">–</td><td>0</td><td rowspan="2">–</td><td rowspan="2">–</td><td>2 BCLK cycles</td></tr>
<tr><td>1</td><td>3 BCLK cycles</td></tr>
<tr><td rowspan="2">Internal ROM/ RAM</td><td rowspan="2">–</td><td rowspan="2">–</td><td>0</td><td rowspan="2">–</td><td>1 BCLK cycle</td></tr>
<tr><td>1</td><td>2 BCLK cycles</td></tr>
<tr><td rowspan="19">External memory</td><td rowspan="12">Separate bus</td><td rowspan="12">–</td><td rowspan="12">–</td><td>00001b</td><td>2 BCLK cycles</td></tr>
<tr><td>00010b</td><td>3 BCLK cycles</td></tr>
<tr><td>00011b</td><td>4 BCLK cycles</td></tr>
<tr><td>00100b</td><td>5 BCLK cycles</td></tr>
<tr><td>00101b</td><td>6 BCLK cycles</td></tr>
<tr><td>00110b</td><td>7 BCLK cycles</td></tr>
<tr><td>01010b</td><td>4 BCLK cycles</td></tr>
<tr><td>01011b</td><td>5 BCLK cycles</td></tr>
<tr><td>01100b</td><td>6 BCLK cycles</td></tr>
<tr><td>10011b</td><td>6 BCLK cycles</td></tr>
<tr><td>10100b</td><td>7 BCLK cycles</td></tr>
<tr><td>10110b</td><td>9 BCLK cycles</td></tr>
<tr><td rowspan="7">Multiplexed bus</td><td rowspan="7">–</td><td rowspan="7">–</td><td>01010b</td><td>4 BCLK cycles</td></tr>
<tr><td>01011b</td><td>5 BCLK cycles</td></tr>
<tr><td>01101b</td><td>7 BCLK cycles</td></tr>
<tr><td>10011b</td><td>6 BCLK cycles</td></tr>
<tr><td>10100b</td><td>7 BCLK cycles</td></tr>
<tr><td>10101b</td><td>8 BCLK cycles</td></tr>
<tr><td>10110b</td><td>9 BCLK cycles</td></tr>
</table>

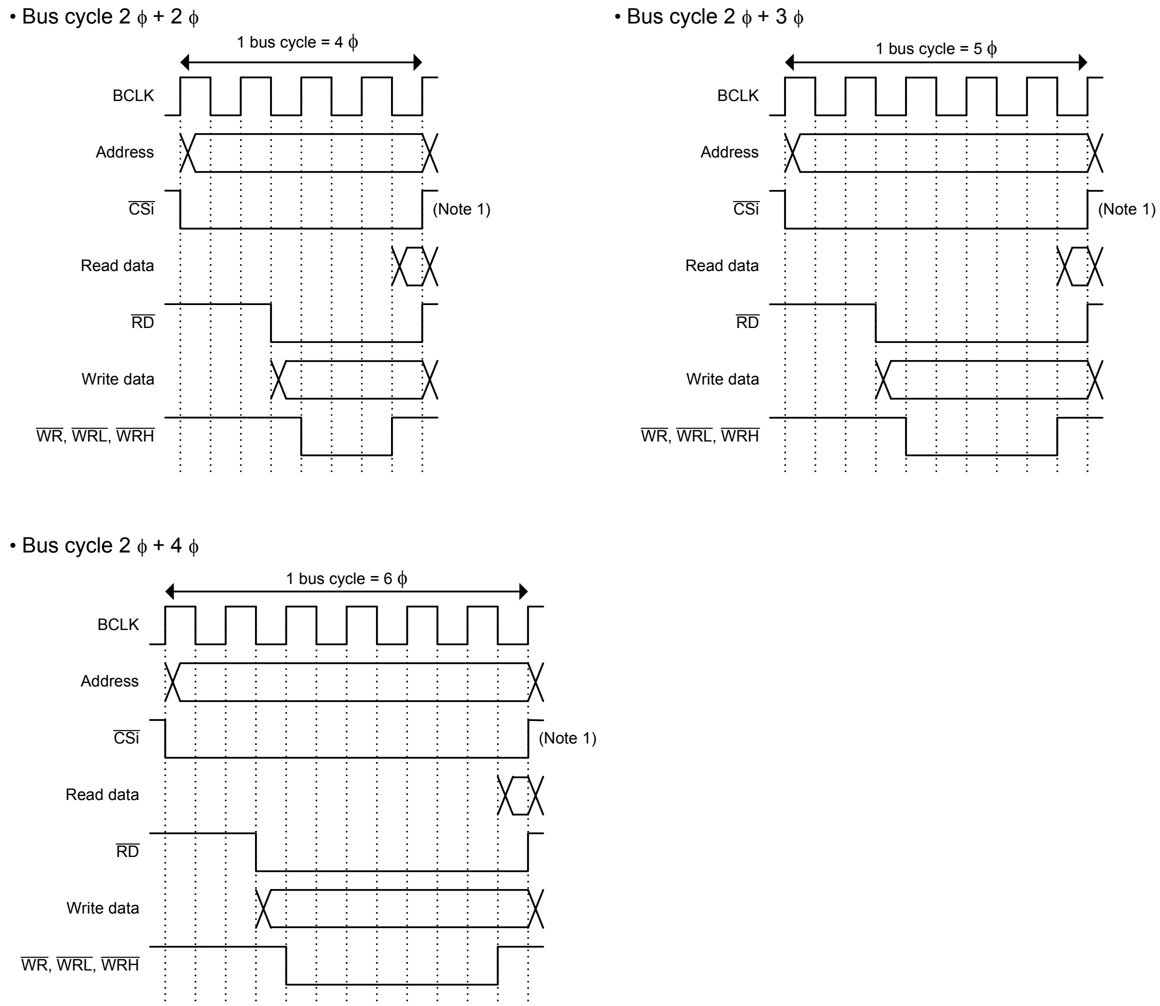
NOTE:
1. Set the PM13 bit to 1 before accessing CAN-associated registers.



NOTE:

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

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

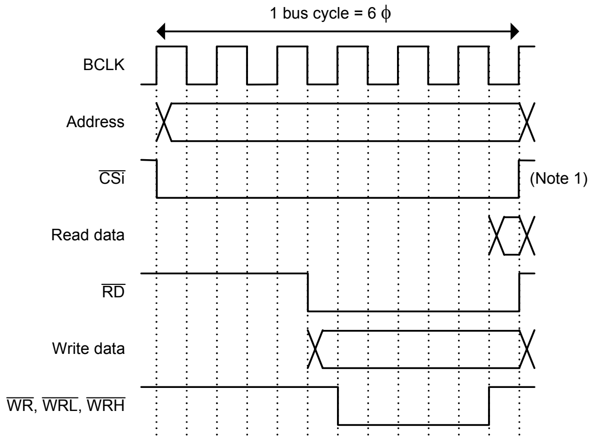


i = 0 to 3

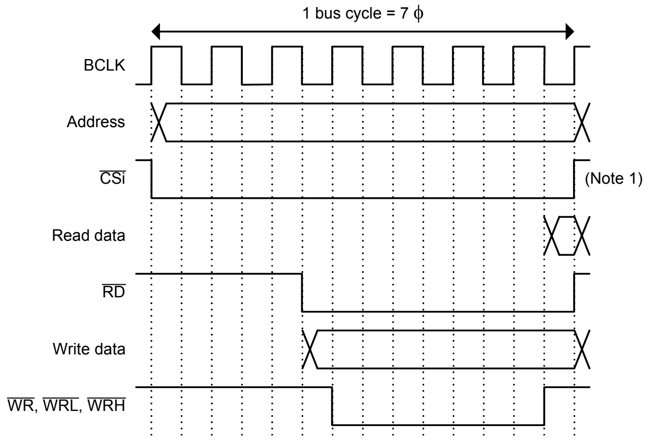
NOTE:
1. When the MCU accesses the same $\overline{CS}$ area consecutively, the $\overline{CSi}$ pin keeps outputting "L".

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

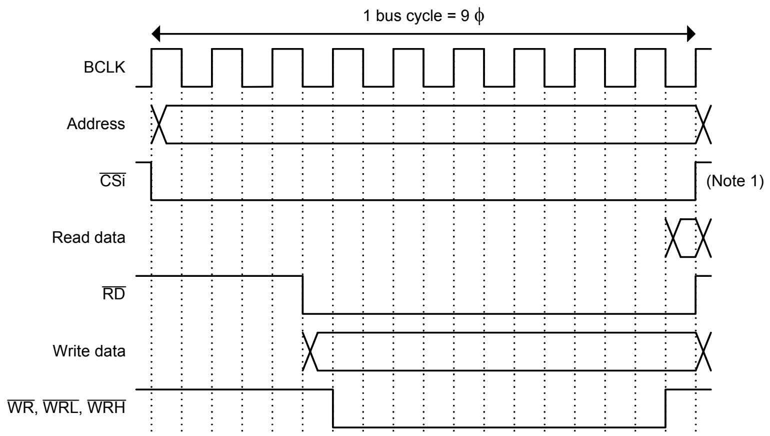
• Bus cycle 3 ϕ + 3 ϕ



• Bus cycle 3 ϕ + 4 ϕ



• Bus cycle 3 ϕ + 6 ϕ

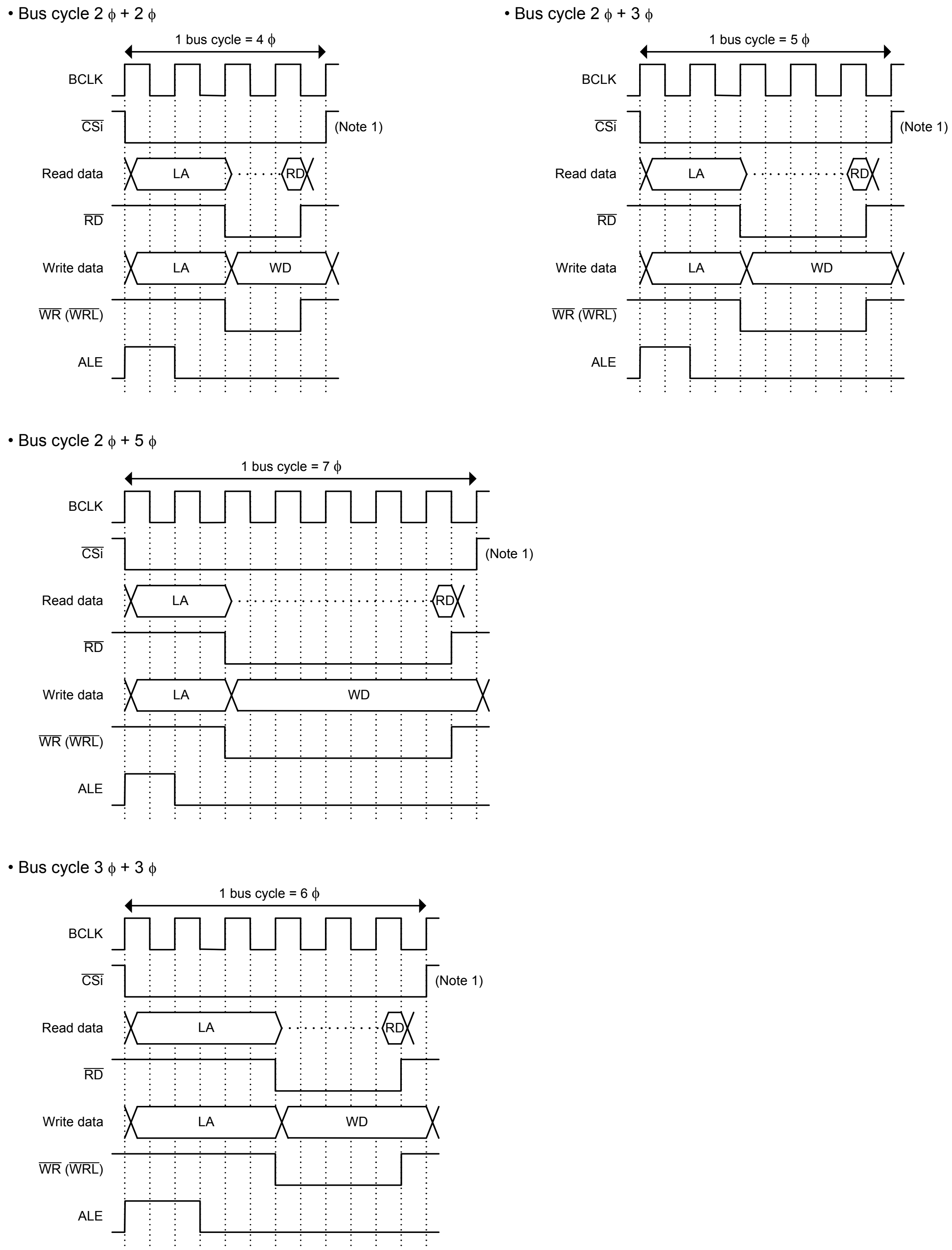


i = 0 to 3

NOTE:

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

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



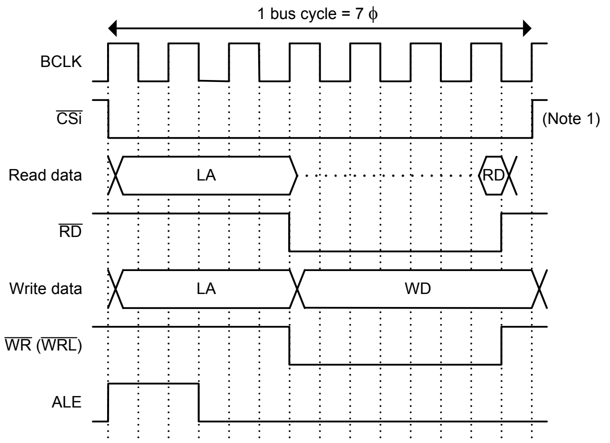
LA: Latch address RD: Read data WD: Write data

i=0 to 3

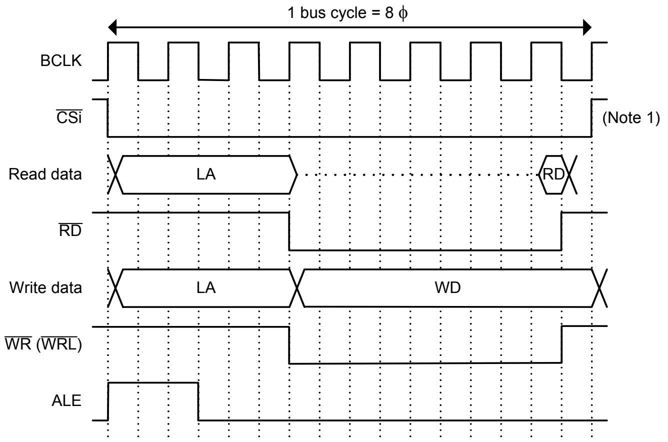
NOTE:
1. When the MCU accesses the same $\overline{CS}$ area consecutively, the $\overline{CSi}$ pin keeps outputting "L".

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

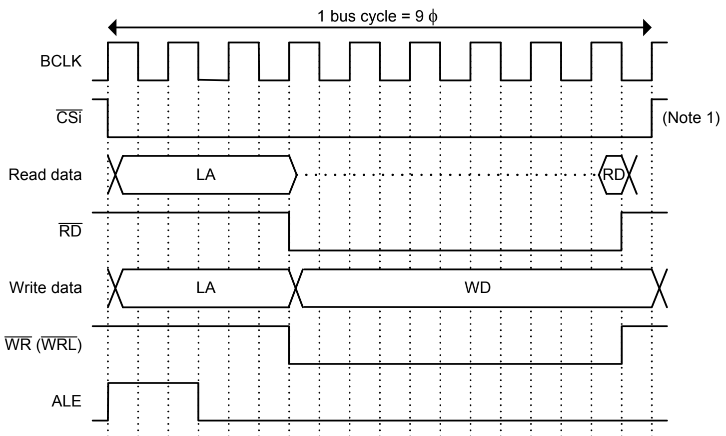
• Bus cycle 3 ϕ + 4 ϕ



• 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:

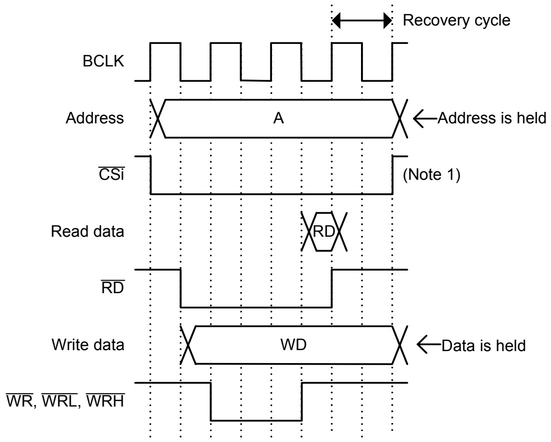
1. When the MCU accesses the same $\overline{CS}$ area consecutively, the $\overline{CSi}$ pin keeps outputting "L".

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

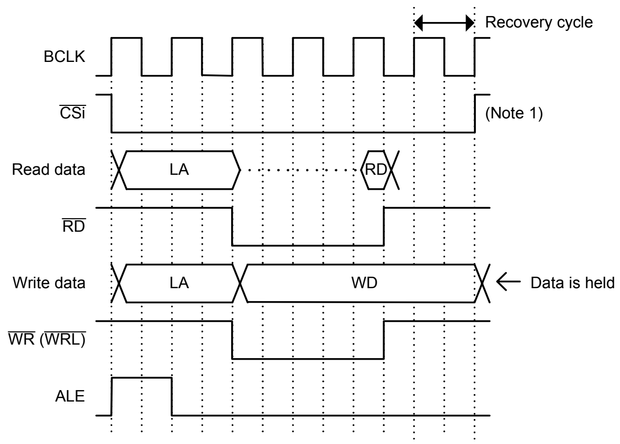
8.2.4.1 Bus Cycle with Recovery Cycle Inserted

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

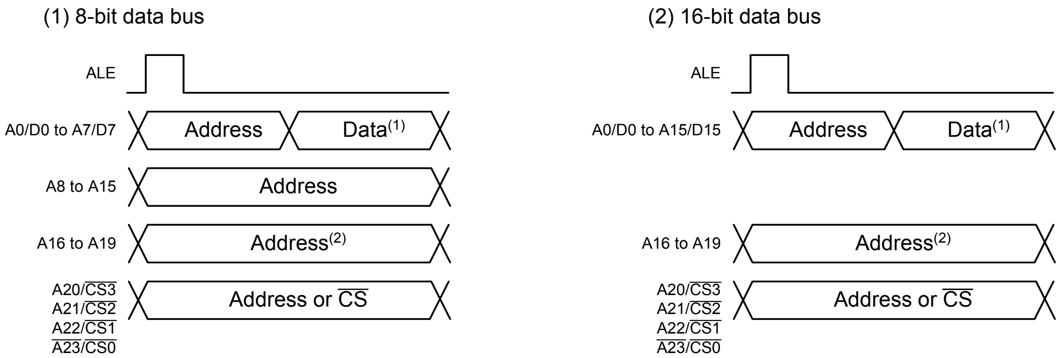
i = 0 to 3

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 $\overline{\text{CS}}$ areas, A16 to A19 become I/O ports.

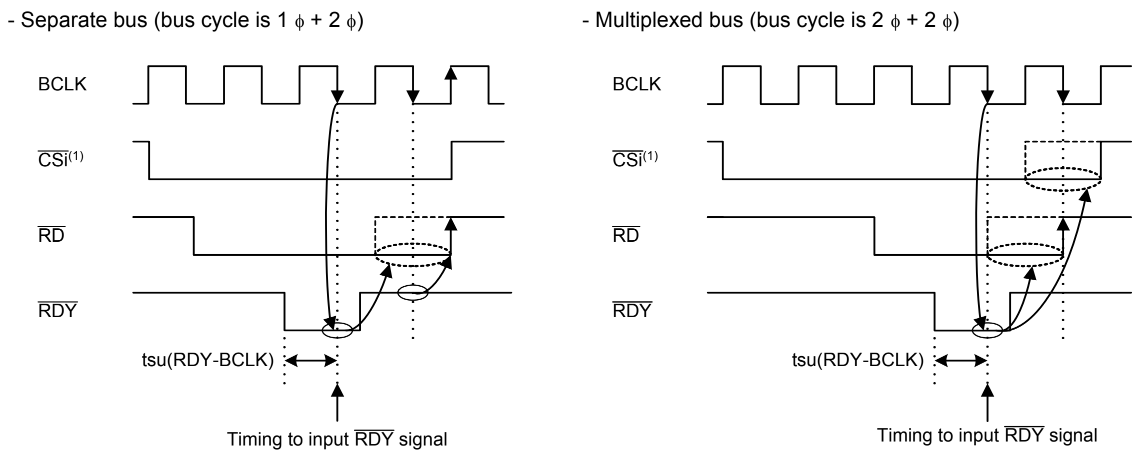
Figure 8.10 ALE Output and Address/Data Bus

8.2.6 $\overline{\text{RDY}}$ Input

The $\overline{\text{RDY}}$ signal facilitates access to external devices requiring longer access time. When $\overline{\text{RDY}}$ 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 $\overline{\text{RDY}}$ 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 $\overline{\text{RDY}}$ input. Figure 8.11 shows an example of the $\overline{\text{RD}}$ signal that is extended by the $\overline{\text{RDY}}$ signal.

Table 8.7 MCU States while "L" is Input to the $\overline{\text{RDY}}$ Pin

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



i = 0 to 3

: Wait states inserted by RDY input

tsu(RDY-BCLK): RDY input setup time

NOTE:
1. Chip-select output (CSi) may be extended depending on the CPU state such as the instruction queue buffer.

Figure 8.11 RD Output Signal Extended by RDY Input

8.2.7 HOLD Input

The 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 HOLD pin, the MCU enters a hold state after the bus access in progress is completed. While the HOLD pin is held “L”, the MCU remains in a hold state and the HLDA pin outputs an “L” signal. Table 8.8 lists the MCU states in hold state.
Bus is used in the following priority order: 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)
RD, WR, A0 to A22, A23, D0 to D15, CS0 to CS3, BHE	High-impedance
HLDA	Outputs “L”
ALE	Outputs “L”
Programmable I/O ports	Maintains the same state as when “L” is input to 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, $\overline{A23}$	Hold the last accessed address in the external space
D0 to D15	High-impedance
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$	Outputs “H”
$\overline{BHE}$	Holds the output level at the time when the MCU accessed the external space or SFR area for the last time
$\overline{CS}$	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.

9. Clock Generation Circuits

9.1 Types of the Clock Generation Circuit

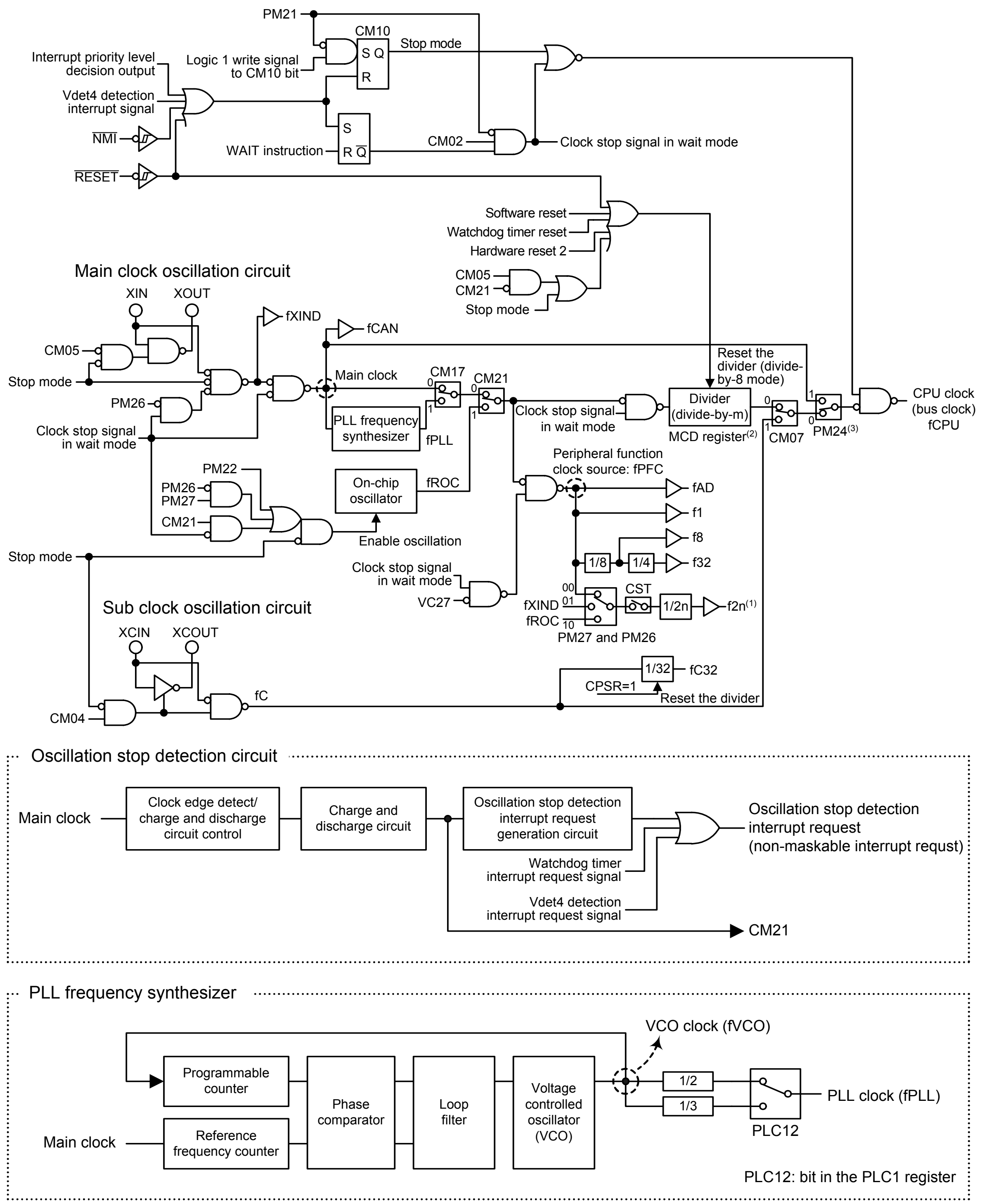
The MCU has four on-chip clock generation circuits to generate system clock signals.

- 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 32 MHz	32.768 kHz	Approx. 1 MHz	Up 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	30 MHz or 20 MHz: Input 10 MHz to the main clock 32 MHz or 21.3 MHz Input 8 MHz to the main clock

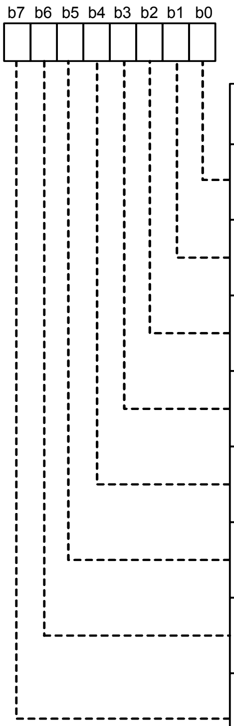


VC27: bit in the VCR2 register
CM02, CM04, CM05, and CM07: bits in the CM0 register
CM10 and CM17: bits in the CM1 register
CM21: bit in the CM2 regsiter
PM21, PM22, PM24, PM26, and PM27: bits in the PM2 register
CST: bit in the TCSPR register
CPSR: bit in the CPSRF register

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).
3. To use the XIN clock as the CAN clock, set the PM24 bit to 1 when accessing the CAN module.

Figure 9.1 Clock Generation Circuit

System Clock Control Register 0[1]



Symbol: CM0 | Address: 0006h | After Reset: 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 in memory expansion mode or microprocessor mode. Port 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]

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

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]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
MCD	000Ch	XXX0 1000b

Bit Symbol	Bit Name	Function	RW
MCD0	Main clock division select bits[2, 3]	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 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.
   In on-chip oscillator mode, bits MCD4 to MCD0 do not become 01000b even if the CM05 bit in the CM0 register is set to 1 (main clock stops).
3. When the PM24 bit in the PM2 register is set to 0 (clock selected by the CM07 bit), access the CAN-associated registers after bits MCD4 to MCD0 are set to 10010b.

Figure 9.4 MCD Register

Oscillation Stop Detection Register(1)

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

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 set to 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, 5)

b7 b6 b5 b4 b3 b2 b1 b0: [ ] [1] [0] [1] [X] [ ] [ ] [ ]

Symbol	Address	After Reset
PLC0	0026h	0001 X010b

Bit Symbol	Bit Name	Function	RW
PLC00	Programmable counter select bits(3)	The VCO clock is the main clock multiplied by the following variables. b2 b1 b0 0 1 1: Multiply-by-6 1 0 0: Multiply-by-8 Do not set to values other than the above	RW
PLC01			RW
PLC02			RW
– (b3)	Reserved bit	Read as undefined value	–
– (b4)	Reserved bit	Set to 1	RW
– (b5)	Reserved bit	Set to 0	RW
– (b6)	Reserved bit	Set to 1	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. Set registers PLC0 and PLC1 simultaneously in 16-bit units .

PLL Control Register 1 (1, 2, 3, 4)

b7 b6 b5 b4 b3 b2 b1 b0: [0] [0] [0] [X] [0] [ ] [1] [0]

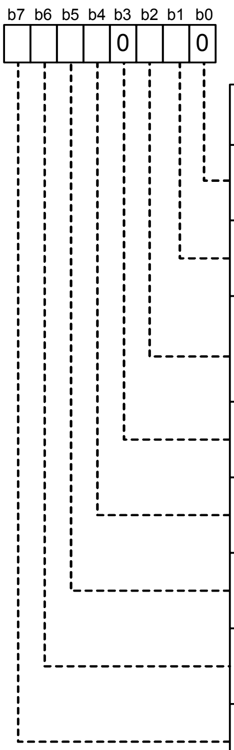
Syambol	Address	After Reset
PLC1	0027h	000X 0000b

Bit Symbol	Bit Name	Function	RW
– (b0)	Reserved bit	Set to 0	RW
– (b1)	Reserved bit	Set to 1	RW
PLC12	PLL clock division select bit	0: Divide-by-2 1: Divide-by-3	RW
– (b3)	Reserved bit	Set to 0	RW
– (b4)	Reserved bit	Read as undefined value	–
– (b7-b5)	Reserved bits	Set to 0	RW

NOTES:
1. Set the PLC1 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 the PLC1 register has no effect.
3. Set the PLC1 register while the PLC07 bit is 0 (PLL stopped).The PLC1 register can be written only once.
4. Set registers PLC0 and PLC1 simultaneously in 16-bit units.

Figure 9.6 PLC0 Register, PLC1 Register

Processor Mode Register 2[1]



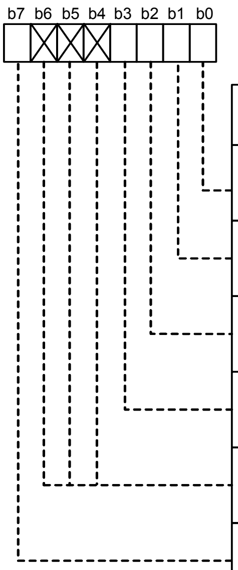
Symbol	Address	After Reset
PM2	0013h	00h

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
– (b3)	Reserved bit	Set to 0	RW
PM24	CPU clock select bit 3	0: Clock selected by the CM07 bit 1: Main clock[5]	RW
PM25	CAN clock select bit	0: f1 1: fCAN	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 of 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
5. When the PM25 bit is set to 1 (CAN clock is fCAN), set the PM24 bit to 1 before accessing the CAN-associated registers.

Figure 9.7 PM2 Register

Count Source Prescaler Register

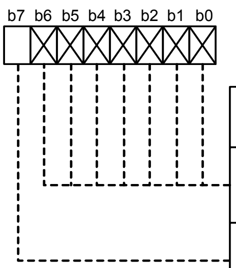


Symbol	Address	After Reset[2]
TCSPR	035Fh	0XXX 0000b

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



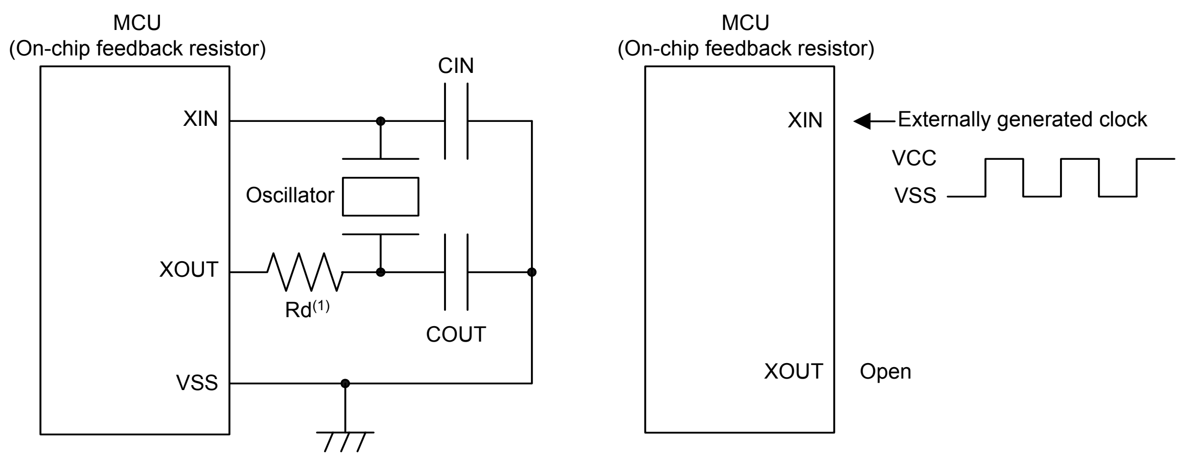
Symbol	Address	After Reset
CPSRF	0341h	0XXX XXXXb

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

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 9.9 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.



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 9.9 Main Clock Circuit Connection

9.1.2 Sub Clock

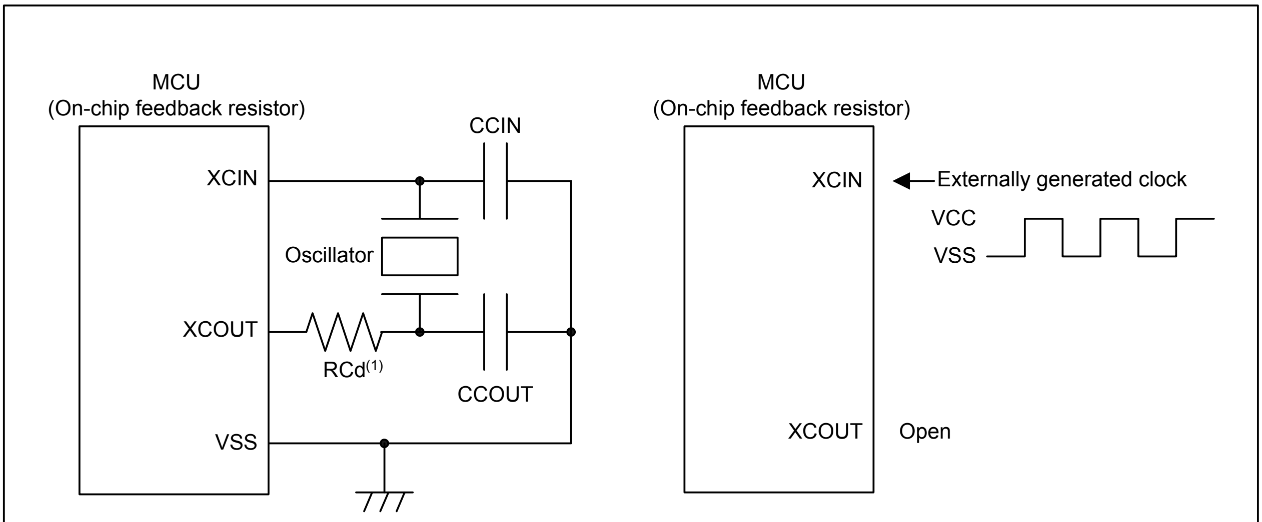
Sub clock oscillation circuit generates the sub clock. The sub clock is used as the clock source for the CPU clock and for timer A and timer B. fC, which has the same frequency as the sub clock can be output from the CLKOUT pin.

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 9.10 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.



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 9.10 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 Vdet4 detection 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 9.11 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.

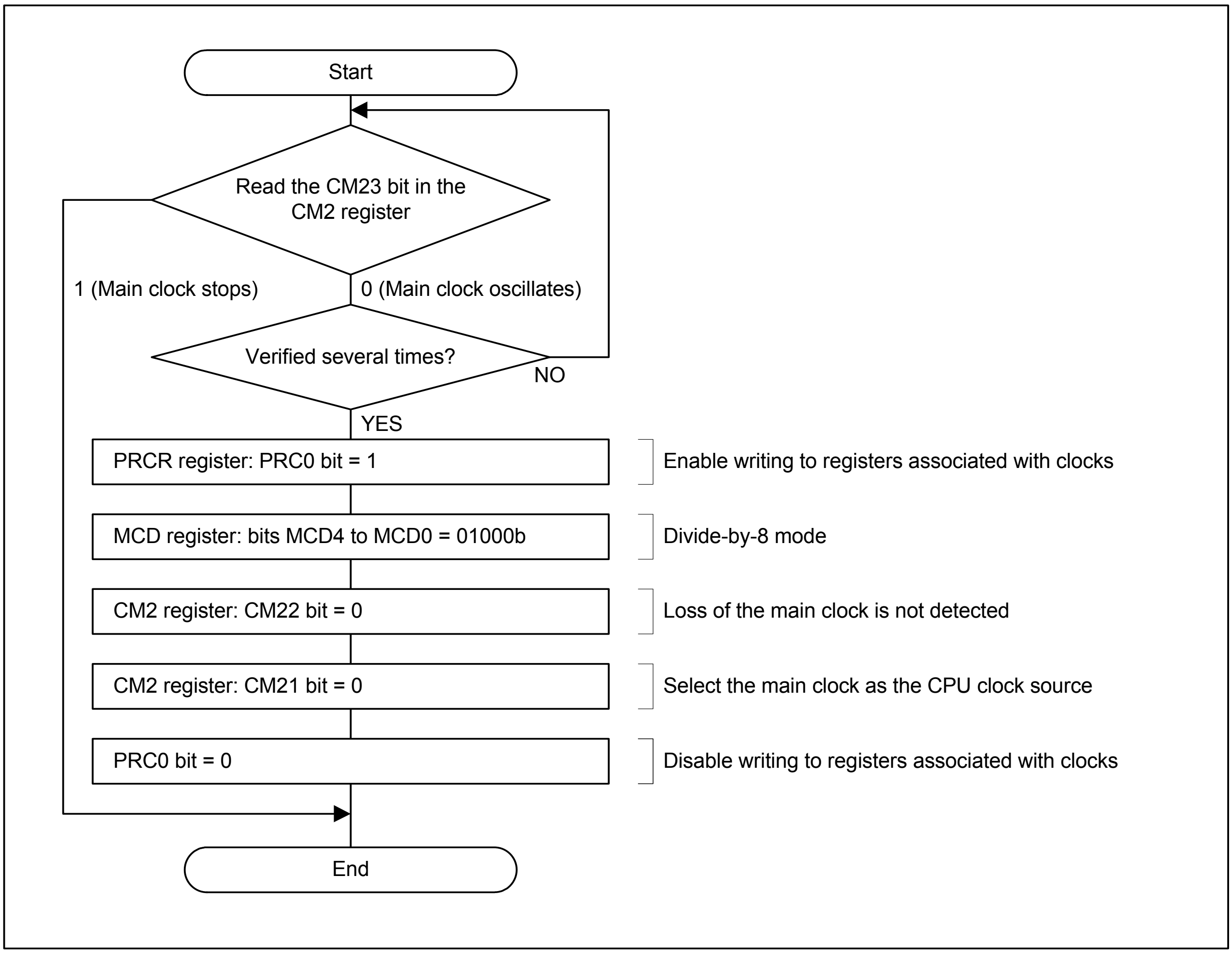


Figure 9.11 Procedure to Switch from On-chip Oscillator Clock to Main Clock

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.

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.

The PLL clock is the VCO clock divided by either 2 or 3. When the PLL clock is used as the clock source for the CPU clock or peripheral function clocks, set each bit as shown in Table 9.3. Figure 9.12 shows the procedure to use the PLL clock as the CPU clock source.

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 the PLC07 bit to 0 (PLL stops).

Table 9.3 Bit Settings to Use PLL Clock as CPU Clock Source

Multiplication factor	PLC0 Register			PLC1 Register	PLL Clock
	PLC02 bit	PLC01 bit	PLC00 bit	PLC12 bit	
2	0	1	1	1	fPLL = 2 × fXIN
3				0	fPLL = 3 × fXIN
8/3	1	0	0	1	fPLL = 8/3 × fXIN
4				0	fPLL = 4 × fXIN

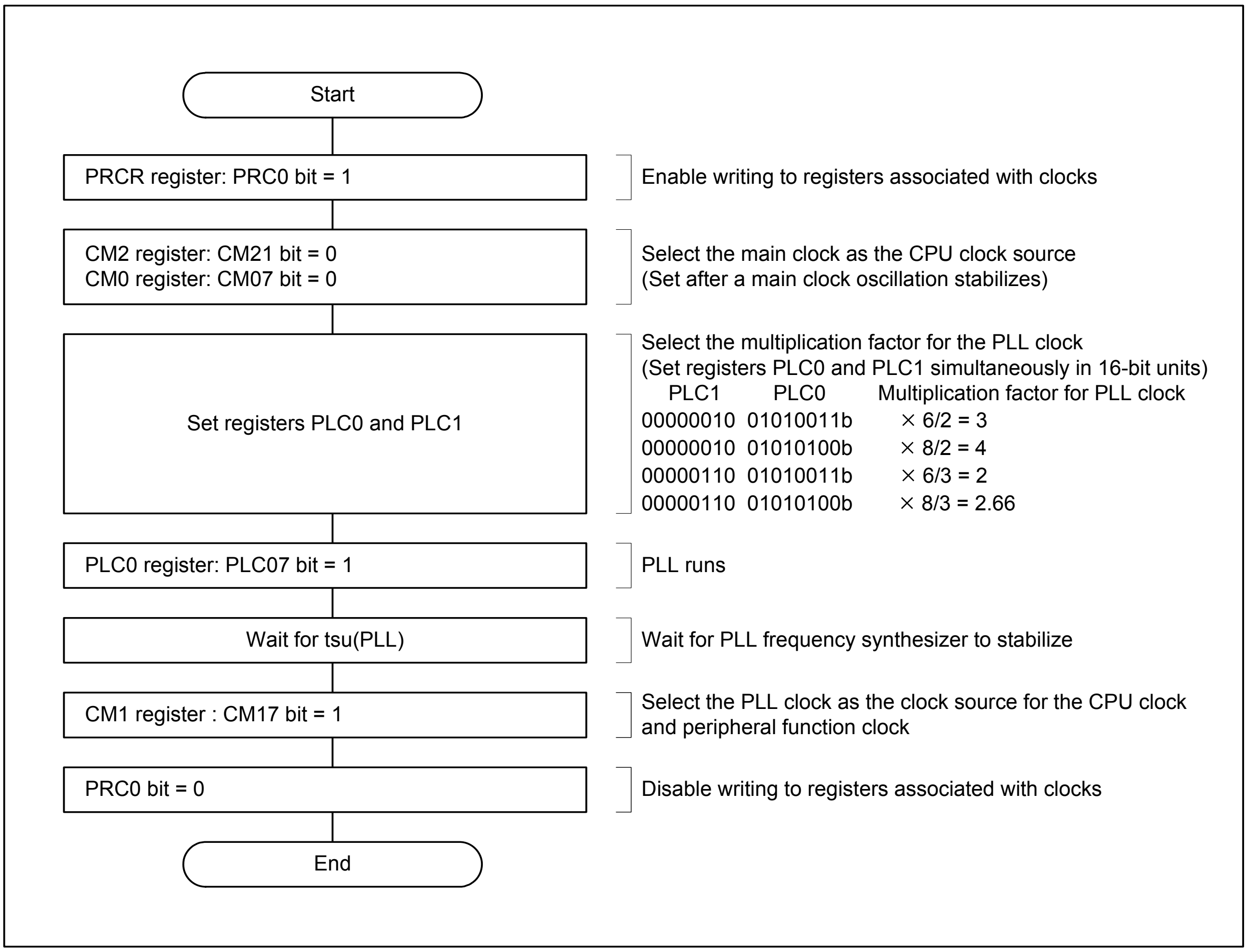


Figure 9.12 Procedure to Use PLL Clock as CPU Clock Source

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.

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. f1 is also used to operate the intelligent I/O and CAN modules.

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.

9.3.4 fCAN

fCAN has the same frequency as the main clock. It is the clock for the CAN module only.

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) Bits CM01 and CM00	P5_3/CLKOUT Pin Function
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).

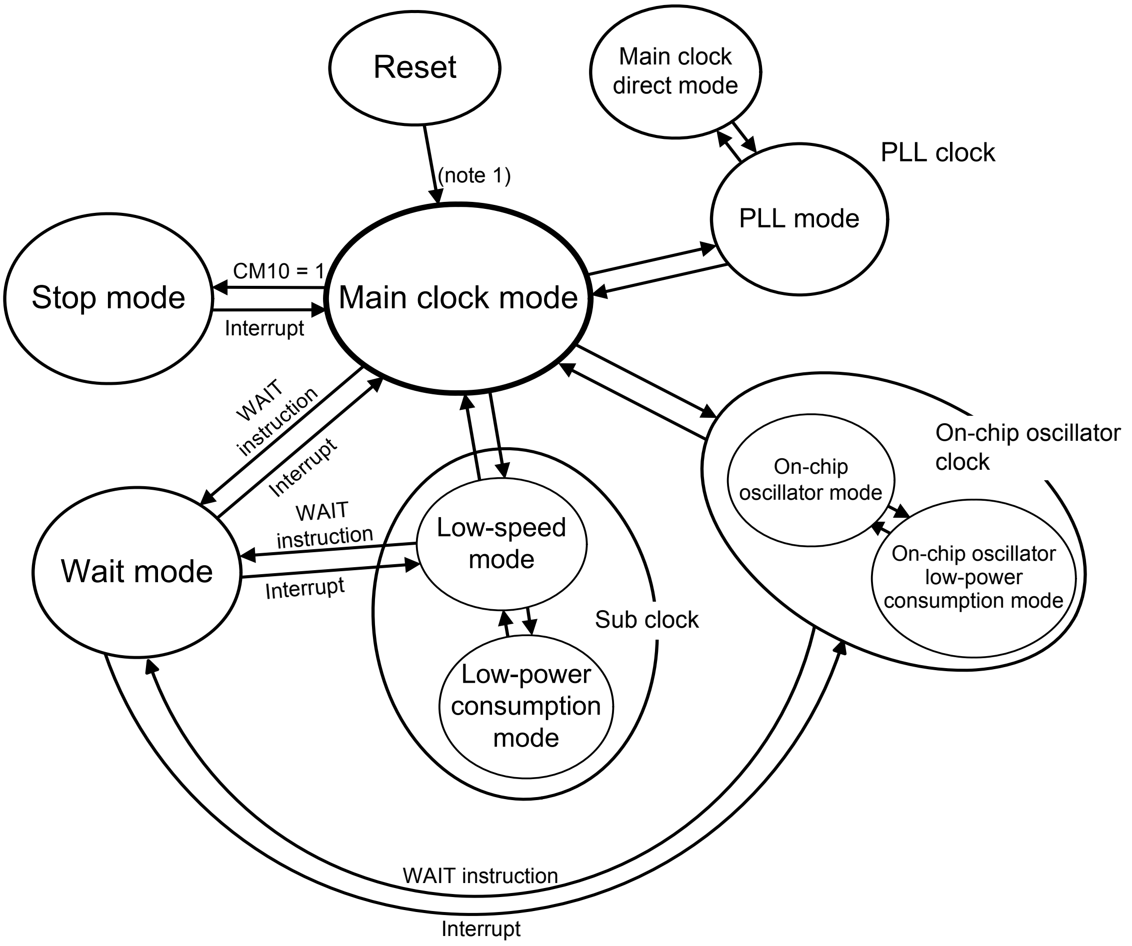
Table 9.5 CLKOUT Pin Function in Memory Expansion Mode and Microprocessor Mode

CM0 Register(1) Bits CM01 and CM00	PM1 Register(2) Bits PM15 and PM14	PM0 Register(2) PM07 bit	CLKOUT/BCLK/ALE Pin Function
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, on-chip oscillator low-power consumption mode, and main clock direct mode.
Figure 9.13 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 9.13 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.

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.

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 Main Clock Direct Mode

The main clock is used as the source for the CPU clock in main clock direct mode. The PLL clock is used for fPFC.

When fCAN is used to operate the CAN modules, enter main clock direct mode before accessing the CAN-associated registers.

Table 9.6 Operation Mode Setting

CPU Clock Source	Operating Mode	Oscillation Control				Selector		
		CM0 Register		PLC0 Register	CM2 Register	CM1 Register	CM0 Register	PM2 Register
		CM05	CM04	PLC07	CM21(1)	CM17	CM07	PM24
Main clock	Main clock mode	0	0 or 1	0 or 1	0	0	0	0
	Main clock direct mode(2)	0	0 or 1	0 or 1	0	0	0	1
PLL clock	PLL mode	0	0 or 1	1	0	1	0	0
Sub clock	Low-speed mode	0	1	0 or 1	0	0	1	0
	Low power consumption mode	1	1	0	0	0	1	0
On-chip oscillator clock	On-chip oscillator mode	0	0 or 1	0 or 1	1	0	0	0
	On-chip oscillator low-power consumption mode	1	0 or 1	0	1	0	0	0

NOTES:
1. The CM21 bit in the CM2 register has both the oscillation control and selector functions.
2. Refer to **23.2 CAN Clock and CPU Clock** for details.

9.5.2 Wait Mode

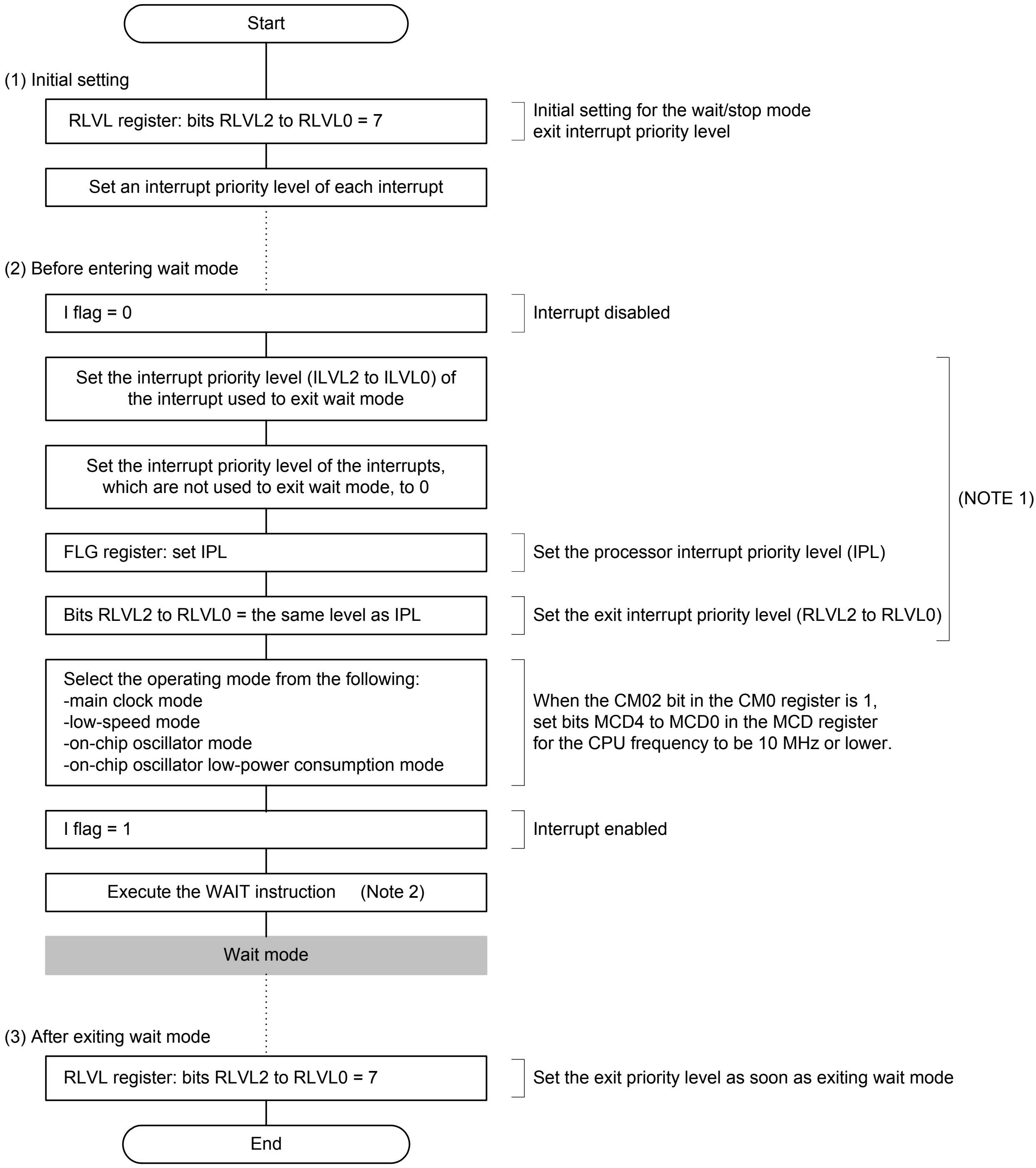
In wait mode, the CPU and watchdog timer stop operating. If the PM22 bit in the PM2 register is set to 1 (on-chip oscillator clock as watchdog timer count source), the watchdog timer continues operating. Since the main clock, sub clock, and on-chip oscillator clock continue running, peripheral functions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

To enter wait mode with the CM02 bit in the CM0 register set to 1, set bits MCD4 to MCD0 in the MCD register for the CPU clock frequency to be 10 MHz or lower after dividing the main clock.
Figure 9.14 shows a procedure to enter wait mode.



NOTES:
1. Set each level to meet the formula shown as below.
(ILVL2 to ILVL0) > IPL = (RLVL2 to RLVL0)
2. Insert at least 4 NOP's after WAIT instruction.

Figure 9.14 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

<table>
<tr><th colspan="2">Pin</th><th>Memory Expansion Mode
Microprocessor Mode</th><th>Single-Chip Mode</th></tr>
<tr><td colspan="2">Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BHE}}$</td><td>Maintain the state immediately before entering wait mode</td><td rowspan="4"></td></tr>
<tr><td colspan="2">$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$</td><td>“H”</td></tr>
<tr><td colspan="2">$\overline{\text{HLDA}}$, BCLK</td><td>“H”</td></tr>
<tr><td colspan="2">ALE</td><td>“L”</td></tr>
<tr><td colspan="2">Ports</td><td colspan="2">Maintain the state immediately before entering wait mode</td></tr>
<tr><td rowspan="2">CLKOUT</td><td>When fC is selected</td><td colspan="2">Continue to output the clock</td></tr>
<tr><td>When f8, f32 are selected</td><td colspan="2">• 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 clock stops in wait mode): The clock is stopped and holds the level immediately before entering wait mode</td></tr>
</table>

9.5.2.4 Exiting Wait Mode

Wait mode is exited by the hardware reset 1, hardware reset 2, $\overline{\text{NMI}}$ interrupt, Vdet4 detection 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
Vdet4 detection 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
CAN interrupt	Available	Available when fCAN is used
Intelligent I/O Interrupt	Available	Not available

9.5.3 Stop Mode

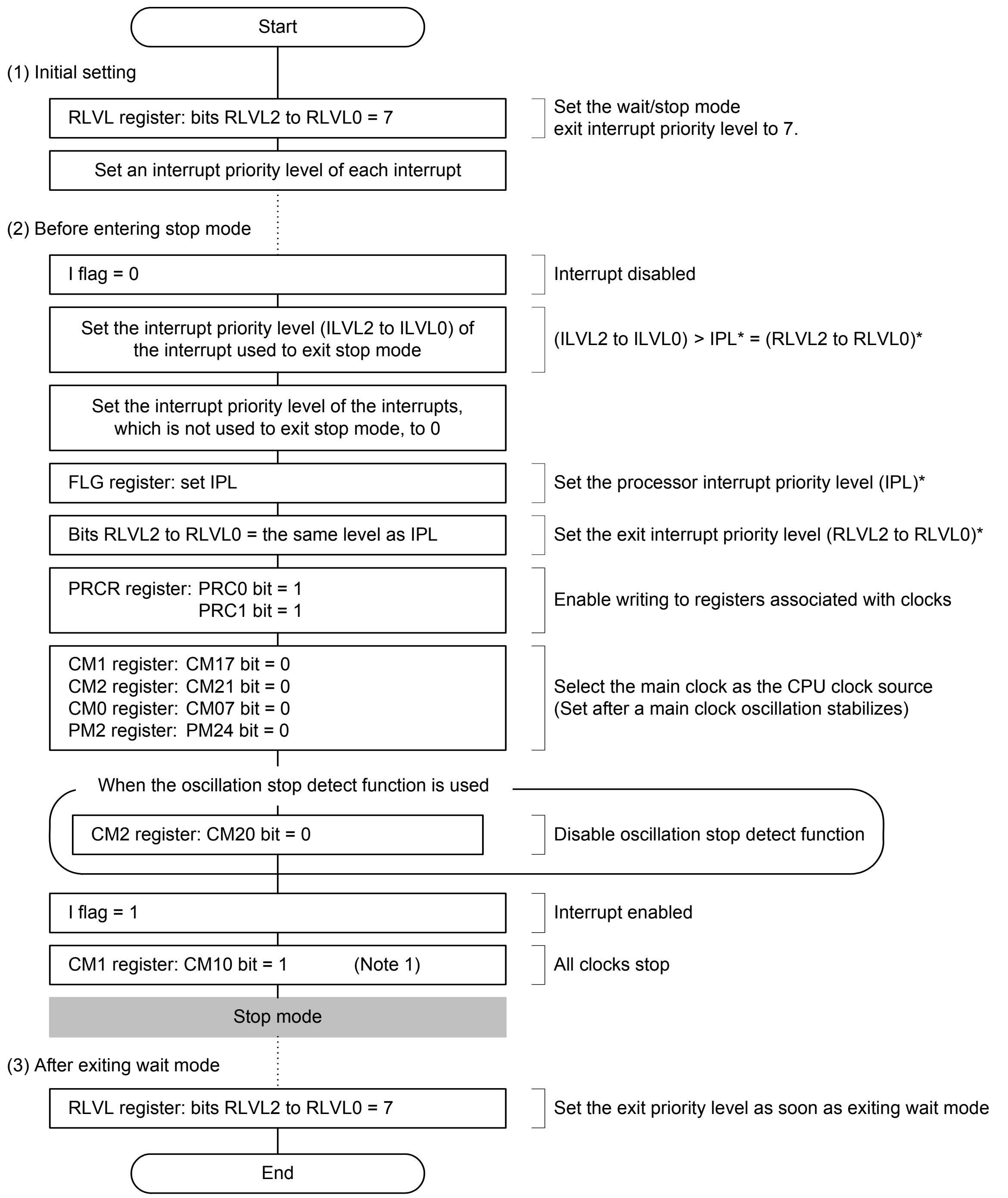
In stop mode, all clocks are stopped. Since the CPU clock and peripheral function clocks are stopped, the CPU and the peripheral functions which are operated by these clocks stop their operation. The least power is required to operate the MCU in stop mode. Enter stop mode from main clock mode.

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.
Figure 9.15 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
                .
                .
                .
```



NOTE:
1. 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 9.15 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, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Maintain the state immediately before entering stop mode	
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		“H”	
$\overline{HLDA}$, BCLK		“H”	
ALE		“H”	
Ports		Maintain the state immediately before entering stop mode	Maintain the state immediately before entering stop mode
CLKOUT	When fC is selected	“H”	“H”
	When f8, f32 are selected	The clock is stopped and holds the level immediately before entering stop mode	The clock is stopped and holds the level immediately before entering stop mode
XIN		Placed in a high-impedance state	Placed in a high-impedance state
XOUT		“H”	“H”
XCIN, XCOUT		Placed in a high-impedance state	Placed in a high-impedance state

9.5.3.3 Exiting Stop Mode

Stop mode is exited by the hardware reset 1, $\overline{NMI}$ interrupt, Vdet4 detection interrupt, or peripheral function interrupts. The following are the peripheral function interrupts that can be used to exit stop mode.

- Key input interrupt
- $\overline{INT}$ 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 1, $\overline{NMI}$ interrupt, or Vdet4 detection 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 1 after the voltage has satisfied the recommended operating conditions.

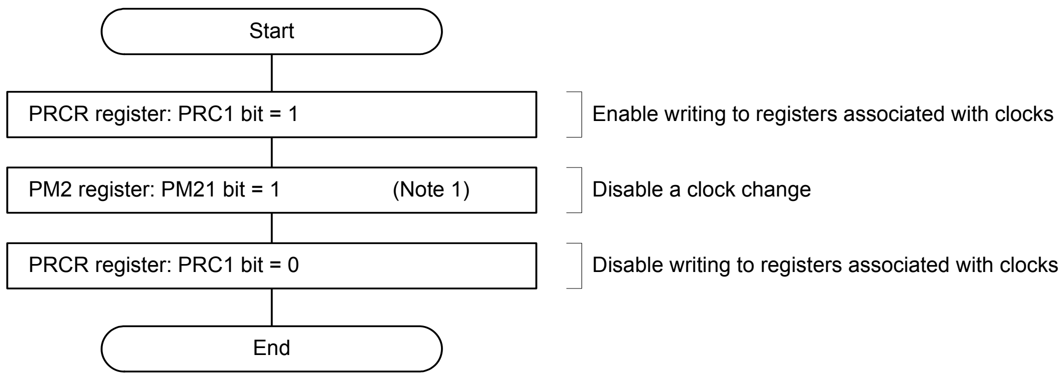
9.6 System Clock Protect Function

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 registers PLC0 and PLC1

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

Figure 9.16 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).



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

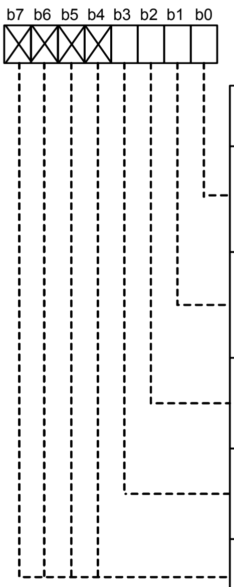
Figure 9.16 Procedure to Use System Clock Protect Function

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, PLC0, and PLC1 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 VCR2 and D4INT is enabled 0: Write disable 1: Write enable	RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

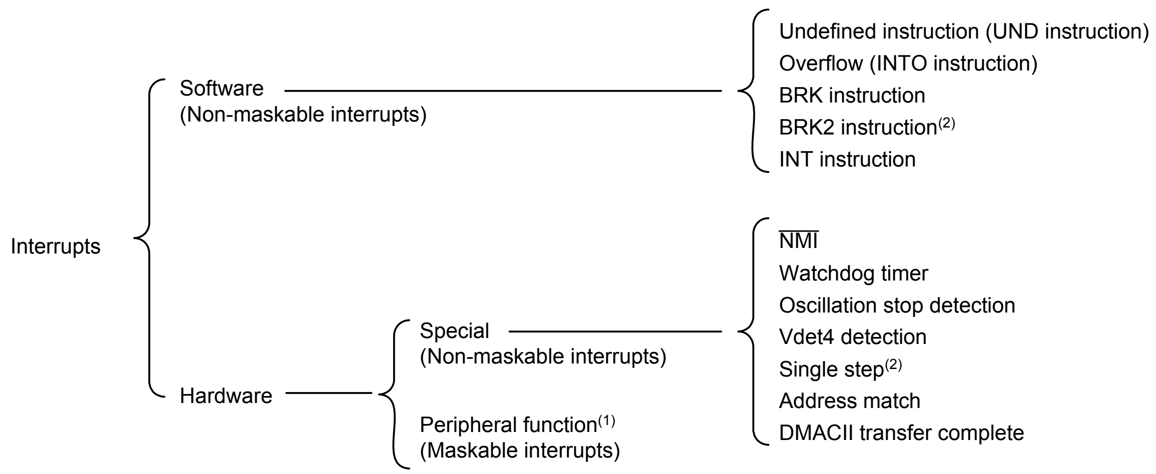
NOTES:
1. Bits PRC0, PRC1, and PRC3 do not become 0 automatically even after a write to the SFR area. Set bits PRC0, PRC1, and PRC3 to 0 by a program.
2. The PRC2 bit becomes 0 by a write to the SFR area after the PRC2 bit is set to 1.

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 54 and 57 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 54 and 57, 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 Vdet4 Detection Interrupt

The Vdet4 detection interrupt occurs when the voltage applied to VCC1 rises above or drops below Vdet4. Refer to **6.2** Vdet4 Detection 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.

11.3.2 DMACII End-of-Transfer Complete Interrupt

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

11.3.3 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 54 and 57 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.

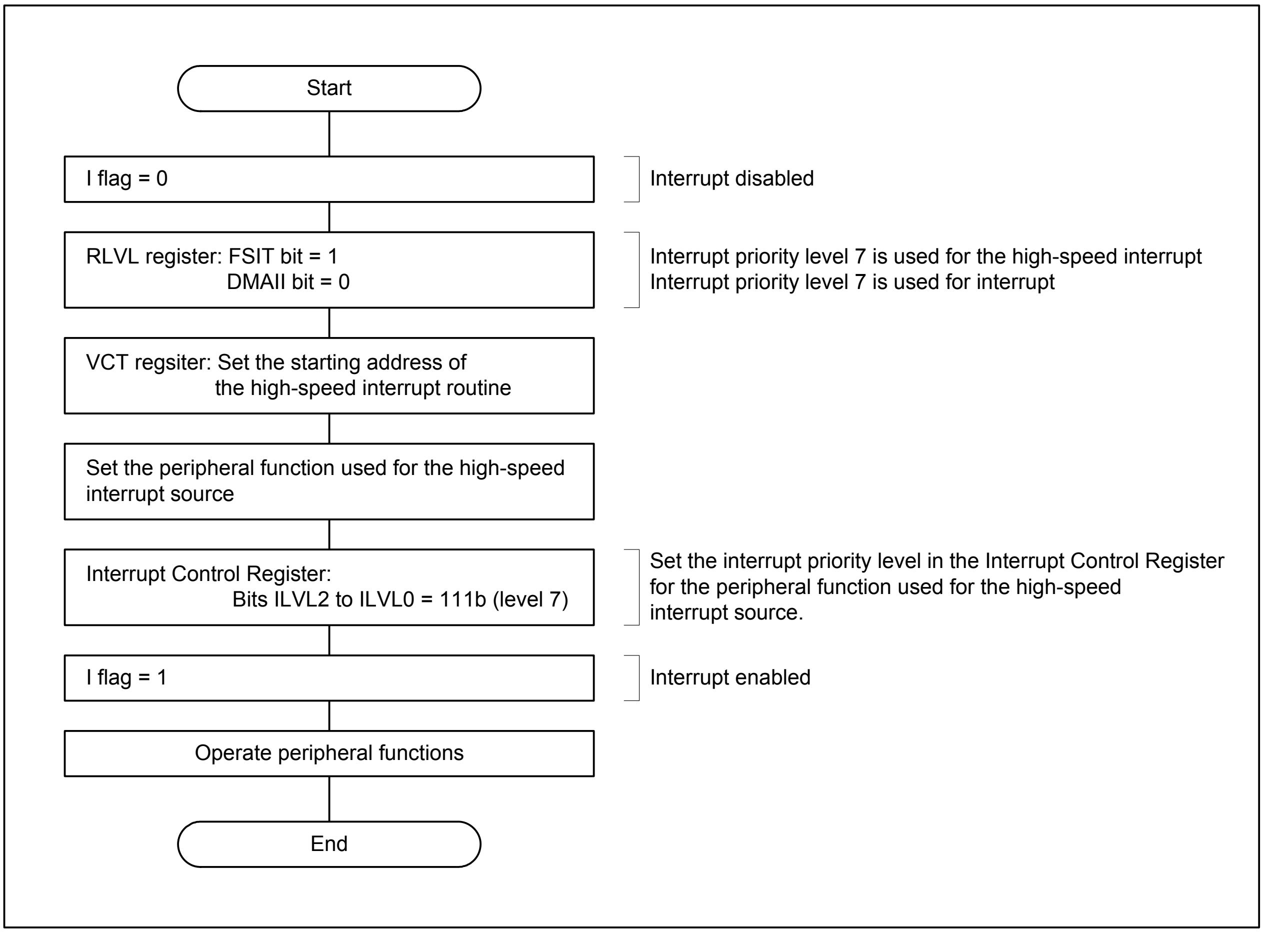


Figure 11.2 Procedure to Use High-Speed Interrupt

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.

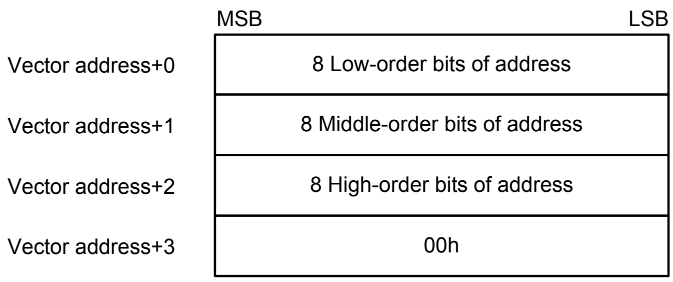


Figure 11.3 Interrupt Vector

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 **26.2.2 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 Vdet4 detection interrupt.	Voltage detection 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	
$\overline{INT5}$	+104 to +107 (0068h to 006Bh)	26	Interrupts
$\overline{INT4}$	+108 to +111 (006Ch to 006Fh)	27	
$\overline{INT3}$	+112 to +115 (0070h to 0073h)	28	
$\overline{INT2}$	+116 to +119 (0074h to 0077h)	29	
$\overline{INT1}$	+120 to +123 (0078h to 007Bh)	30	
$\overline{INT0}$	+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. These are the addresses offset 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
Intelligent I/O interrupt 0, CAN10(5), UART5 reception	+176 to +179 (00B0h to 00B3h)	44	Intelligent I/O, CAN, UART5, UART6, $\overline{\text{INT}}$
Intelligent I/O interrupt 1, CAN11(5), UART5 transmission	+180 to +183 (00B4h to 00B7h)	45	
Intelligent I/O interrupt 2	+184 to +187 (00B8h to 00BBh)	46	
Intelligent I/O interrupt 3	+188 to +191 (00BCh to 00BFh)	47	
Intelligent I/O interrupt 4	+192 to +195 (00C0h to 00C3h)	48	
Intelligent I/O interrupt 5, CAN12(5), CAN1 wake-up	+196 to +199 (00C4h to 00C7h)	49	
Intelligent I/O interrupt 6	+200 to +203 (00C8h to 00CBh)	50	
Intelligent I/O interrupt 7	+204 to +207 (00CCh to 00CFh)	51	
Intelligent I/O interrupt 8	+208 to +211 (00D0h to 00D3h)	52	
Intelligent I/O interrupt 9, CAN00(5), UART6 reception, $\overline{\text{INT6}}$	+212 to +215 (00D4h to 00D7h)	53	
Intelligent I/O interrupt 10, CAN01(5), UART6 transmission, $\overline{\text{INT7}}$	+216 to +219 (00D8h to 00DBh)	54	
Reserved space	+220 to +227 (00DCh to 00E3h)	55, 56	–
Intelligent I/O interrupt 11, CAN02(5), $\overline{\text{INT8}}$	+228 to +231 (00E4h to 00E7h)	57	Intelligent I/O, CAN, $\overline{\text{INT}}$
Reserved space	+232 to +255 (00E8h to 00FFh)	58 to 63	–
INT instruction(2)	+0 to +3 (0000h to 0003h) to +252 to +255 (00FCh to 00FFh)	0 to 63	Interrupts

NOTES:
1. These are the addresses offset 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.
5. Any CAN interrupt source cannot be used in M32C/87B. Only CAN00, CAN01, and CAN02 interrupt sources can be used in M32C/87A.

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.

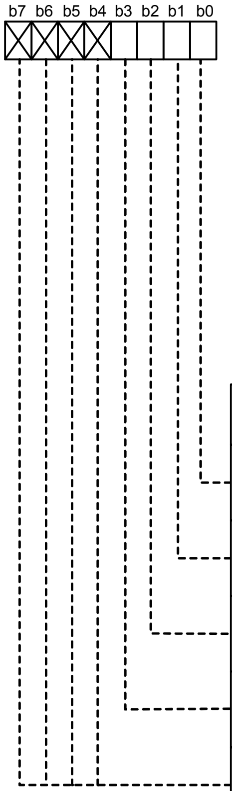
Table 11.4 Interrupt Priority Levels

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



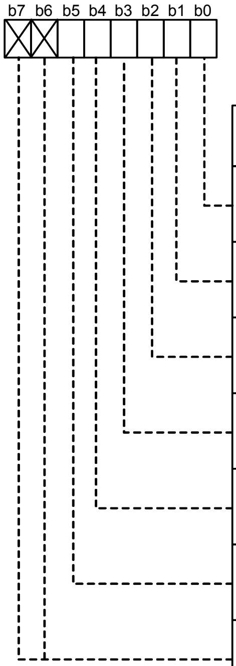
Symbol	Address	After Reset
TA0IC to TA4IC	006Ch, 008Ch, 006Eh, 008Eh, 0070h	XXXX X000b
TB0IC to TB5IC	0094h, 0076h, 0096h, 0078h, 0098h, 0069h	XXXX X000b
S0TIC to S4TIC	0090h, 0092h, 0089h, 008Bh, 008Dh	XXXX X000b
S0RIC to S4RIC	0072h, 0074h, 006Bh, 006Dh, 006Fh	XXXX X000b
BCN0IC to BCN4IC	0071h, 0091h, 008Fh, 0071h(1), 0091h(2)	XXXX X000b
DM0IC to DM3IC	0068h, 0088h, 006Ah, 008Ah	XXXX X000b
AD0IC	0073h	XXXX X000b
KUPIC	0093h	XXXX X000b
IIO0IC to IIO5IC	0075h, 0095h, 0077h, 0097h, 0079h, 0099h	XXXX X000b
IIO6IC to IIO11IC	007Bh, 009Bh, 007Dh, 009Dh, 007Fh, 0081h	XXXX X000b
CAN0IC to CAN2IC	009Dh, 007Fh, 0081h(3)	XXXX X000b
CAN3IC to CAN5IC	0075h, 0095h, 0099h(3)	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(4)	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 CAN-associated registers cannot be used in M32C/87B. Only registers CAN0IC to CAN2IC can be used in M32C/87A for the CAN-associated registers.
   The CAN0IC register controls the CAN00 interrupt.
   The CAN1IC register controls the CAN01 interrupt.
   The CAN2IC register controls the CAN02 interrupt.
   The CAN3IC register controls the CAN10 interrupt.
   The CAN4IC register controls the CAN11 interrupt.
   The CAN5IC register controls the CAN12 interrupt and CAN1 wake-up interrupt.
   The IIO9IC register shares the address with the CAN0IC register.
   The IIO10IC register shares the address with the CAN1IC register.
   The IIO11IC register shares the address with the CAN2IC register.
   The IIO0IC register shares the address with the CAN3IC register.
   The IIO1IC register shares the address with the CAN4IC register.
   The IIO5IC register shares the address with the CAN5IC register.
4. 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 $\overline{INT3}$ to $\overline{INT5}$ 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)

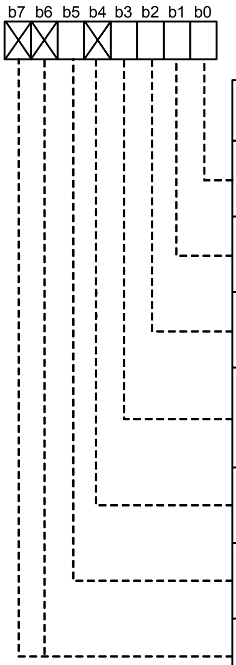
11.6.2.1 Bits ILVL2 to ILVL0

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).

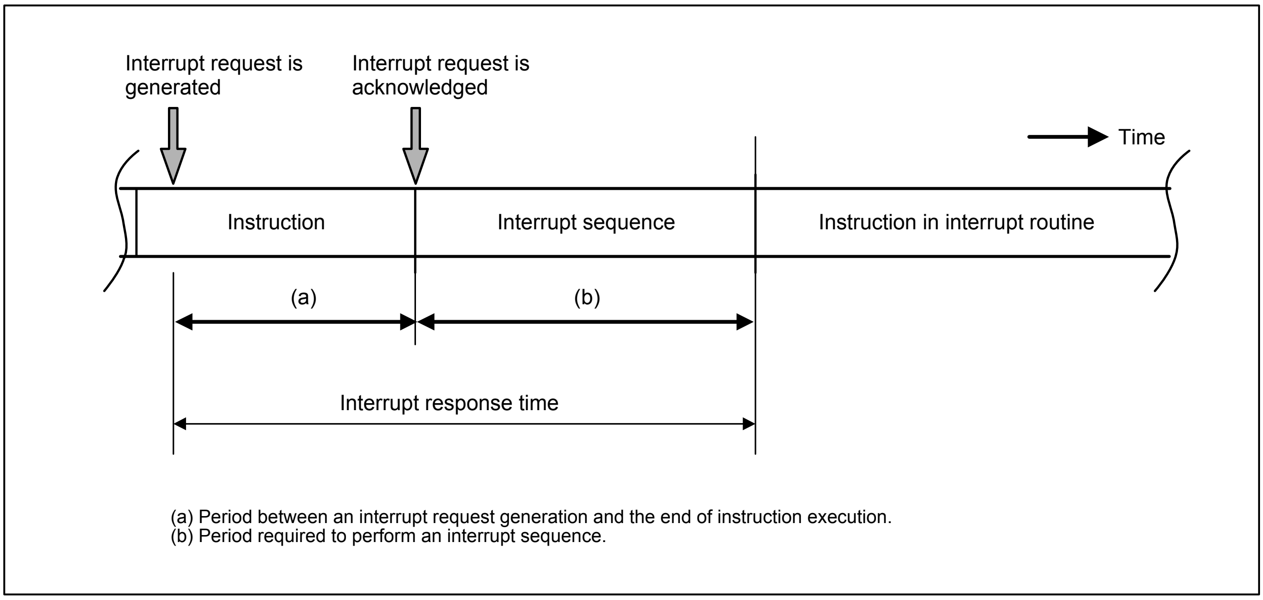


Figure 11.7 Interrupt Response Time

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, $\overline{NMI}$, oscillation stop detection, Vdet4 detection, 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.

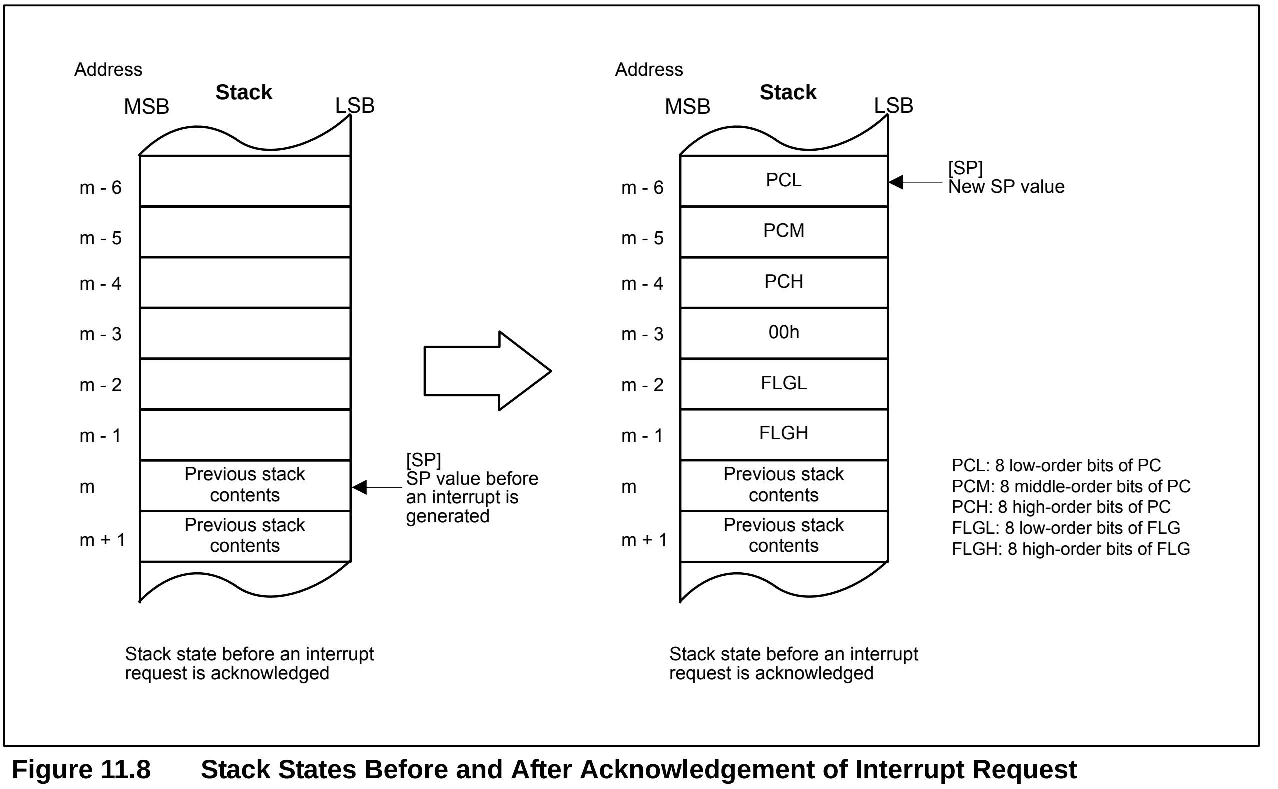


Figure 11.8 Stack States Before and After Acknowledgement of Interrupt Request

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.

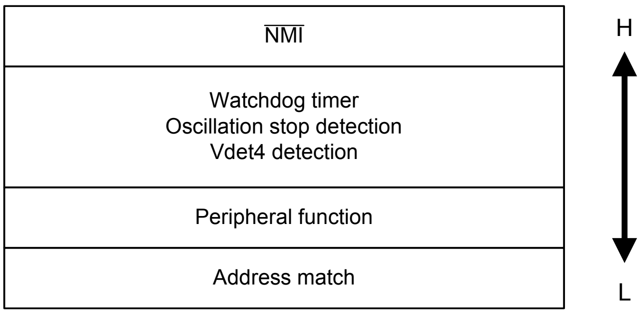


Figure 11.9 Interrupt Priority of Hardware Interrupts

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.

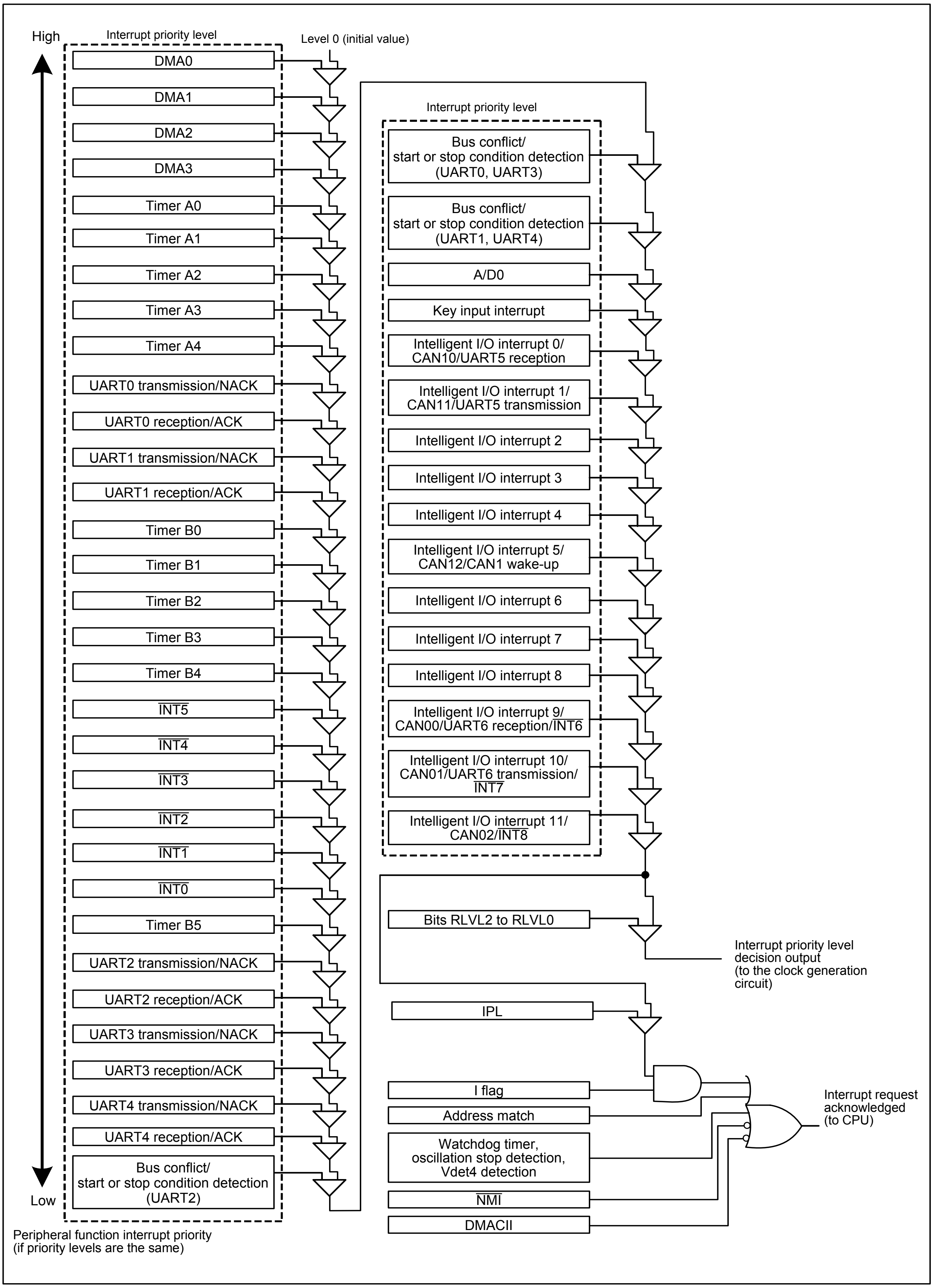


Figure 11.10 Interrupt Priority Level Decision Circuit

11.7 INT Interrupt

External input to pins INT0 to INT8 generates the INT0 to INT8 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. The INT6 to INT8 interrupts are available only in the 144-pin package with edge-sensitive triggering.
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.
To use INT6 to INT8 interrupts with edge sensitive, select a rising edge or falling edge by the IFSRj bit (j = 10 to 12) in the IFSRA register. Interrupts can be enabled or disabled using the INTiE bit in the IIOkIE register (k = 9 to 11) and bits ILVL2 to ILVL0 in the IIOkIC register.
Refer to **11.11 Intelligent I/O Interrupts, CAN Interrupts, UART5 and UART6 Transmit/Receive Interrupts, and INT6 to INT8 Interrupts** for details.
Figure 11.11 shows INTi interrupt setting procedures (i = 0 to 5). Figure 11.12 shows INTi interrupt setting procedures (i = 6 to 8). Figure 11.13 shows the IFSR register and Figure 11.14 shows IFSRA register.

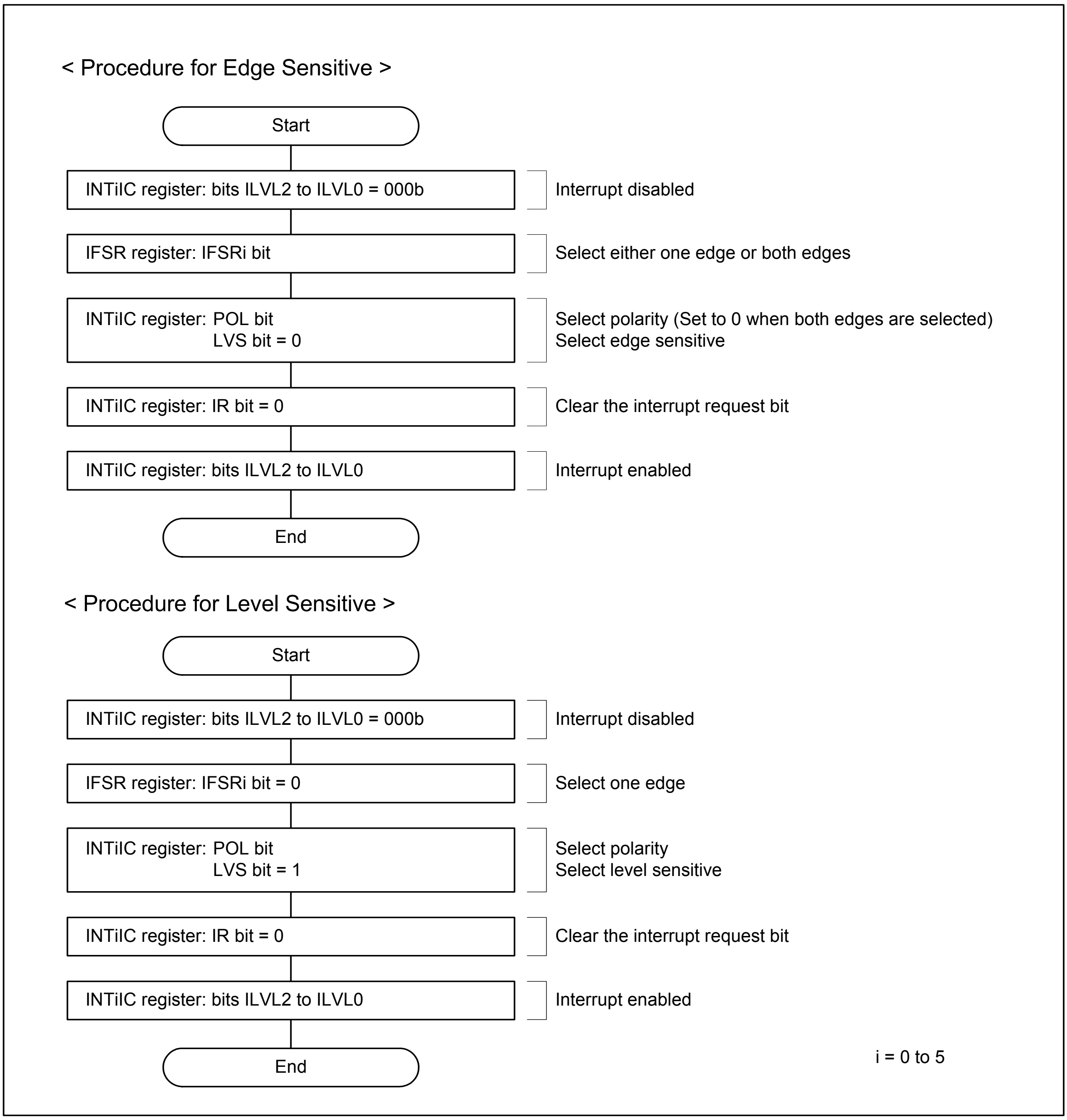


Figure 11.11 $\overline{\text{INTi}}$ Interrupt Setting Procedures (i = 0 to 5)

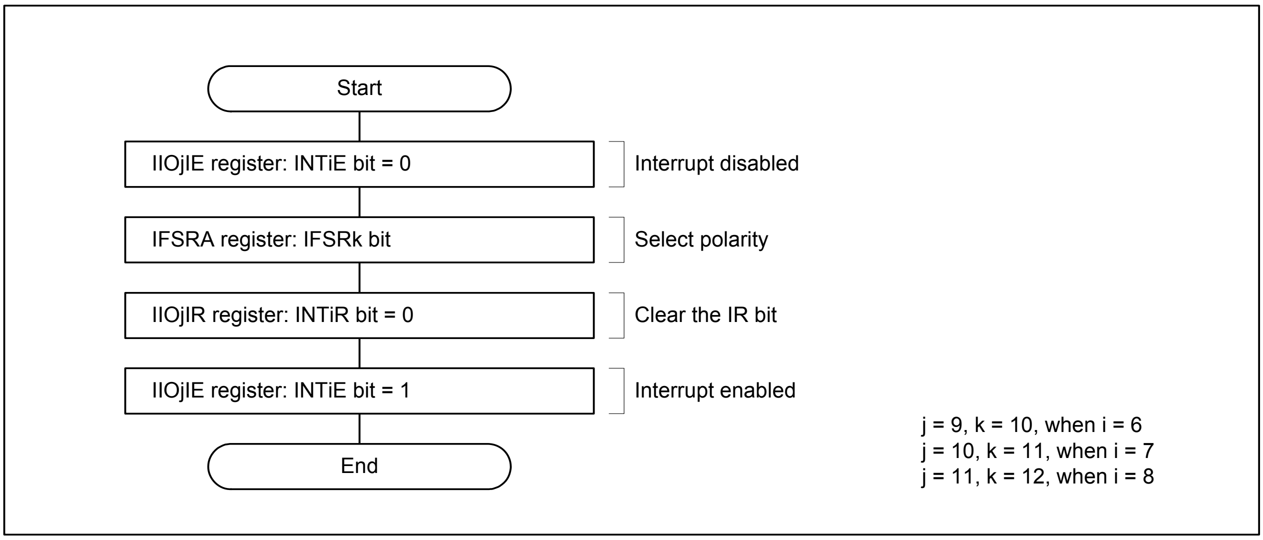


Figure 11.12 $\overline{\text{INTi}}$ Interrupt Setting Procedures (i = 6 to 8)

External Interrupt Source Select Register

b7 b6 b5 b4 b3 b2 b1 b0

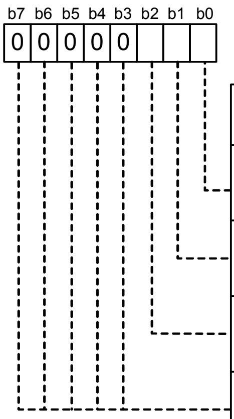
Symbol	Address	After Reset
IFSR	031Fh	00h

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 11.13 IFSR Register

External Interrupt Source Select Register1(1)



Symbol	Address	After Reset
IFSRA	031Eh	00h

Bit Symbol	Bit Name	Function	RW
IFSR10	$\overline{INT6}$ interrupt polarity select bit	0: One edge (falling edge) 1: One edge (rising edge)	RW
IFSR11	$\overline{INT7}$ interrupt polarity select bit	0: One edge (falling edge) 1: One edge (rising edge)	RW
IFSR12	$\overline{INT8}$ interrupt polarity select bit	0: One edge (falling edge) 1: One edge (rising edge)	RW
– (b7-b3)	Reserved bits	Set to 0	RW

NOTE:
1. The IFSRA register is available in the 144-pin package only.

Figure 11.14 IFSRA 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 11.15 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.

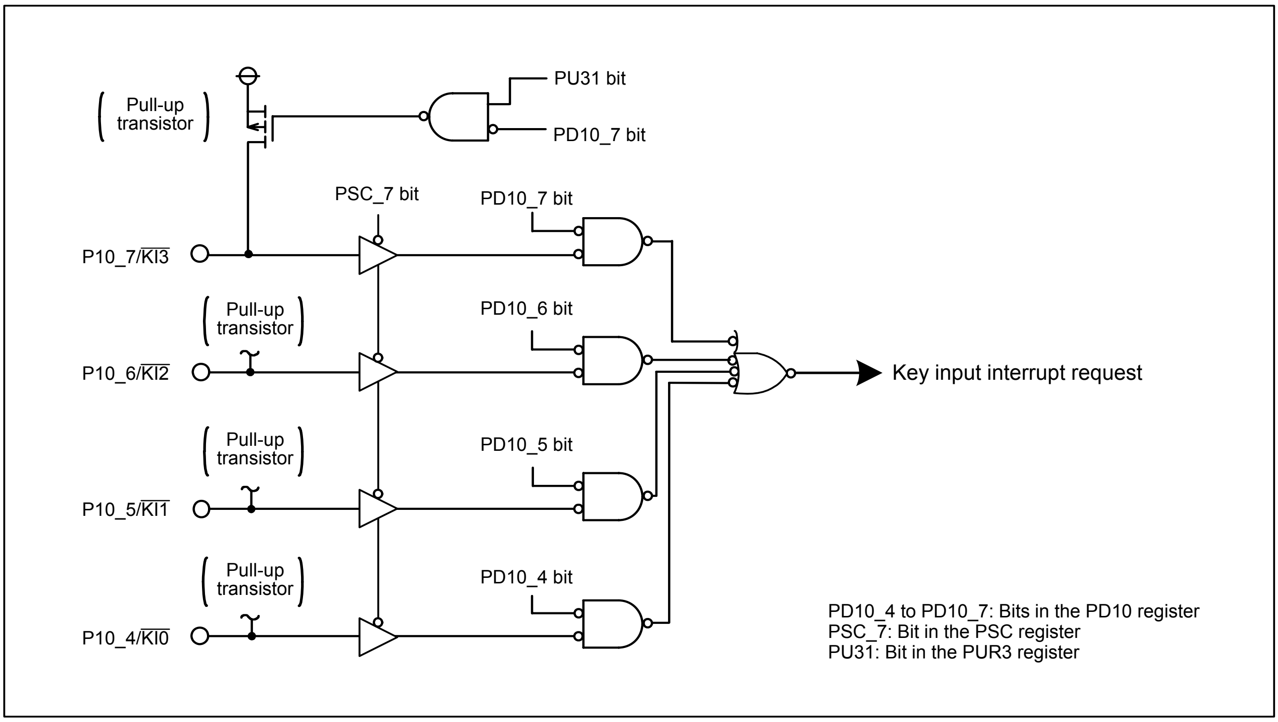


Figure 11.15 Key Input Interrupt Block Diagram

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 11.16 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)

b23 b16 b15 b8 b7 b0

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

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
AIER	0009h	00h

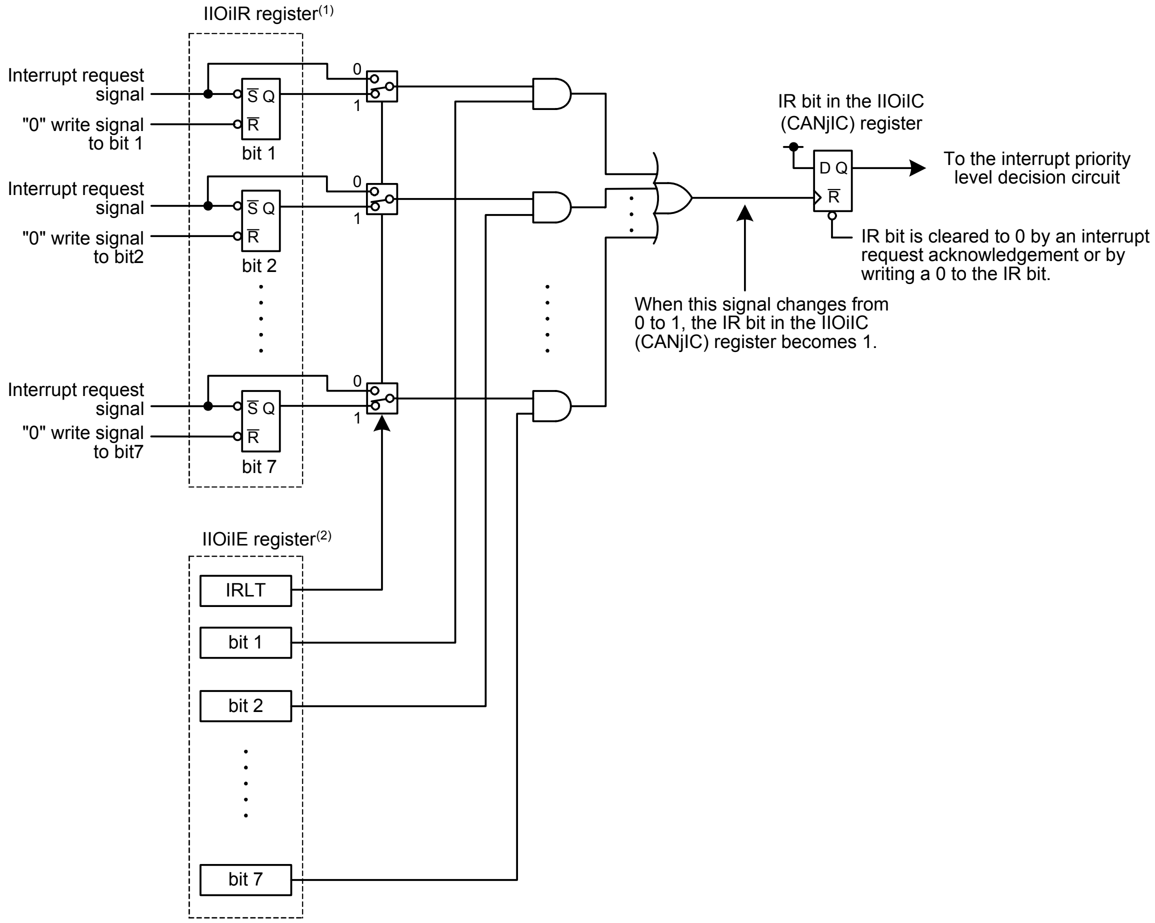
Bit Symbol	Bit Name	Function	RW
AIER0	Address match interrupt 0 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER2	Address match interrupt 2 enable bit	0: interrupt disabled 1: interrupt enabled	RW
AIER3	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 11.16 RMAD0 to RMAD7 Registers, AIER Register

11.11 Intelligent I/O Interrupts, CAN Interrupts, UART5 and UART6 Transmit/ Receive Interrupts, and $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ Interrupts

The intelligent I/O interrupts are shared by CAN interrupt, $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ interrupts, UART5 and UART6 transmit/ receive interrupt. A logical sum of interrupt request signals from individual peripheral functions is used to generate an interrupt.

Figure 11.17 shows a block diagram of the intelligent I/O interrupts. Figure 11.18 shows the IIOiIR (i = 0 to 11) register. Figure 11.19 shows the IIOiIE register.



i = 0 to 11, j = 0 to 5

NOTES:
1. Bits 1 to 7 in the IIOiIR register do not automatically become 0 when the interrupt request is acknowledged. Set to 0 by a program.
2. Do not change the interrupt enable bit (bits 1 to 7 in the IIOiIE register) and IRLT bit in the IIOiIE register simultaneously.

Figure 11.17 Intelligent I/O Interrupt Block Diagram

Interrupt Request Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
IIO0IR to IIO11IR	See below	0000 000Xb

Bit Symbol	Function		RW
– (b0)	Unimplemented. Write 0. Read as undefined value.		–
(Note 1)	Interrupt request flag 1	0: Interrupt not requested 1: Interrupt requested(2)	RW
(Note 1)	Interrupt request flag 2	0: Interrupt not requested 1: Interrupt requested(2)	RW
(Note 1)	Interrupt request flag 3	0: Interrupt not requested 1: Interrupt requested(2)	RW
(Note 1)	Interrupt request flag 4	0: Interrupt not requested 1: Interrupt requested(2)	RW
(Note 1)	Interrupt request flag 5	0: Interrupt not requested 1: Interrupt requested(2)	RW
(Note 1)	Interrupt request flag 6	0: Interrupt not requested 1: Interrupt requested(2)	RW
(Note 1)	Interrupt request flag 7	0: Interrupt not requested 1: Interrupt requested(2)	RW

NOTES:
1. See table below for bit symbols.
2. These bits can be set to only 0. Do not write a 1 to these bits.

Bit Symbols for the Interrupt Request Register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IR	00A0h	CAN10R	U5RR	SIO0RR	G0RIR	–	TM13R/PO13R	–	–
IIO1IR	00A1h	CAN11R	U5TR	SIO0TR	G0TOR	–	TM14R/PO14R	–	–
IIO2IR	00A2h	–	–	SIO1RR	G1RIR	–	TM12R/PO12R	–	–
IIO3IR	00A3h	–	–	SIO1TR	G1TOR	PO27R	TM10R/PO10R	–	–
IIO4IR	00A4h	SRT0R	SRT1R	–	BT1R	–	TM17R/PO17R	–	–
IIO5IR	00A5h	CAN12R	CAN1WUR	–	SIO2RR	–	PO21R	–	–
IIO6IR	00A6h	–	–	–	SIO2TR	–	PO20R	–	–
IIO7IR	00A7h	IE0R	–	–	–	–	PO22R	–	–
IIO8IR	00A8h	IE1R	IE2R	–	BT2R	–	PO23R	TM11R/PO11R	–
IIO9IR	00A9h	CAN00R	INT6R	U6RR	–	–	PO24R	TM15R/PO15R	–
IIO10IR	00AAh	CAN01R	INT7R	U6TR	–	–	PO25R	TM16R/PO16R	–
IIO11IR	00ABh	CAN02R	INT8R	–	–	–	PO26R	–	–

BTqR: Intelligent I/O group q base timer interrupt request
TM1jR: Intelligent I/O group 1 time measurement function j interrupt request
POqjR: Intelligent I/O group q waveform generation function j interrupt request
SIOkRR: Intelligent I/O group k receive interrupt request
SIOkTR: Intelligent I/O group k transmit interrupt request
GmTOR: Intelligent I/O group m HDLC data processing function interrupt request (TO: Transmit Output)
GmRIR: Intelligent I/O group m HDLC data processing function interrupt request (RI: Receive Input)
SRTmR: Intelligent I/O group m special communication function interrupt request
IEkR: Intelligent I/O group 2 IEBus communication function interrupt request
CAN0kR: CAN0 communication function interrupt request
CAN1kR: CAN1 communication function interrupt request
CAN1WUR: CAN1 wake-up interrupt request
INTnR: INTn interrupt request
UpTR: UARTp transmit interrupt request
UpRR: UARTp receive interrupt request
–: Reserved bit. Set to 0

j = 0 to 7
k = 0 to 2
m = 0, 1
n = 6 to 8
p = 5, 6
q = 1, 2

Figure 11.18 IIO0IR to IIO11IR Registers

Interrupt Enable Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
IIO0IE to IIO11E	See below	00h

Bit	Bit Symbol	Bit Name	Function	RW
b0	IRLT	Interrupt request select bit(2)	0: Uses an interrupt request for DMA, DMA II 1: Uses an interrupt request for interrupt	RW
b1	(Note 1)	Interrupt enabled bit 1	0: Disables an interrupt set by bit 1 in the IIOiIR register 1: Enables an interrupt set by bit 1 in the IIOiIR register	RW
b2	(Note 1)	Interrupt enabled bit 2	0: Disables an interrupt set by bit 2 in the IIOiIR register 1: Enables an interrupt set by bit 2 in the IIOiIR register	RW
b3	(Note 1)	Interrupt enabled bit 3	0: Disables an interrupt set by bit 3 in the IIOiIR register 1: Enables an interrupt set by bit 3 in the IIOiIR register	RW
b4	(Note 1)	Interrupt enabled bit 4	0: Disables an interrupt set by bit 4 in the IIOiIR register 1: Enables an interrupt set by bit 4 in the IIOiIR register	RW
b5	(Note 1)	Interrupt enabled bit 5	0: Disables an interrupt set by bit 5 in the IIOiIR register 1: Enables an interrupt set by bit 5 in the IIOiIR register	RW
b6	(Note 1)	Interrupt enabled bit 6	0: Disables an interrupt set by bit 6 in the IIOiIR register 1: Enables an interrupt set by bit 6 in the IIOiIR register	RW
b7	(Note 1)	Interrupt enabled bit 7	0: Disables an interrupt set by bit 7 in the IIOiIR register 1: Enables an interrupt set by bit 7 in the IIOiIR register	RW

NOTES:
1. See table below for bit symbols.
2. To use an interrupt request for interrupt, set the interrupt enabled bit r (r = 1 to 7) to 1 after setting the IRLT bit to 1.

Bit Symbols for the Interrupt Enable Register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IE	00B0h	CAN10E	U5RE	SIO0RE	G0RIE	–	TM13E/PO13E	–	IRLT
IIO1IE	00B1h	CAN11E	U5TE	SIO0TE	G0TOE	–	TM14E/PO14E	–	IRLT
IIO2IE	00B2h	–	–	SIO1RE	G1RIE	–	TM12E/PO12E	–	IRLT
IIO3IE	00B3h	–	–	SIO1TE	G1TOE	PO27E	TM10E/PO10E	–	IRLT
IIO4IE	00B4h	SRT0E	SRT1E	–	BT1E	–	TM17E/PO17E	–	IRLT
IIO5IE	00B5h	CAN12E	CAN1WUE	–	SIO2RE	–	PO21E	–	IRLT
IIO6IE	00B6h	–	–	–	SIO2TE	–	PO20E	–	IRLT
IIO7IE	00B7h	IE0E	–	–	–	–	PO22E	–	IRLT
IIO8IE	00B8h	IE1E	IE2E	–	BT2E	–	PO23E	TM11E/PO11E	IRLT
IIO9IE	00B9h	CAN00E	INT6E	U6RE	–	–	PO24E	TM15E/PO15E	IRLT
IIO10IE	00BAh	CAN01E	INT7E	U6TE	–	–	PO25E	TM16E/PO16E	IRLT
IIO11IE	00BBh	CAN02E	INT8E	–	–	–	PO26E	–	IRLT

BTqE: Intelligent I/O group q base timer interrupt enabled
TM1jE: Intelligent I/O group 1 time measurement function j interrupt enabled
POqjE: Intelligent I/O group q waveform generation function j interrupt enabled
SIOkRE: Intelligent I/O group k receive interrupt enabled
SIOkTE: Intelligent I/O group k transmit interrupt enabled
GmTOE: Intelligent I/O group m HDLC data processing function interrupt enabled (TO: Transmit Output)
GmRIE: Intelligent I/O group m HDLC data processing function interrupt enabled (RI: Receive Input)
SRTmE: Intelligent I/O group m special communication function interrupt enabled
IEkE: Intelligent I/O group 2 IEBus communication function interrupt enabled
CAN0kE: CAN0 communication function interrupt enabled
CAN1kE: CAN1 communication function interrupt enabled
CAN1WUE: CAN1 wake-up interrupt enabled
INTnE: INTn interrupt enabled
UpTE: UARTp transmit interrupt enabled
UpRE: UARTp receive interrupt enabled
–: Reserved bit. Set to 0

i = 1 to 11
j = 0 to 7
k = 0 to 2
m = 0, 1
n = 6 to 8
p = 5, 6
q = 1, 2

Figure 11.19 IO0IE to IIO11IE Registers

To configure for intelligent I/O interrupts, use IIOiIE register (i = 0 to 11), IIOiIR register, and IIOiIC (CANjIC (j = 0 to 5)) register.

11.11.1 IIOiIE Register

- IRLT bit
  Set to 1 to use interrupt requests from individual peripheral functions for interrupts. Set to 0 to use them for DMA or DMACII trigger sources.
- Interrupt enable bit
  Set the interrupt enable bit corresponding to the interrupt to be used, to 1 (interrupt enabled) after setting the IRLT bit.

11.11.2 IIOiIR Register

- Interrupt request flag
  The interrupt request flag becomes 1 (interrupt requested) when an interrupt request is generated. This flag does not automatically become 0 when the interrupt request is acknowledged. Use AND or BCLR instruction to set it to 0 (interrupt not requested) in the interrupt routine. If any of these flags remains 1, the IR bit in the IIOiIC (CANjIC) register does not become 1 when an interrupt request is generated in the same register. (Interrupt does not occur.)
  If an interrupt request is generated while writing a 0 to the corresponding interrupt flag, the flag may not be cleared to 0. In this case, keep writing a 0 until 0 is read.

11.11.3 IIOiIC (CANjIC) Register

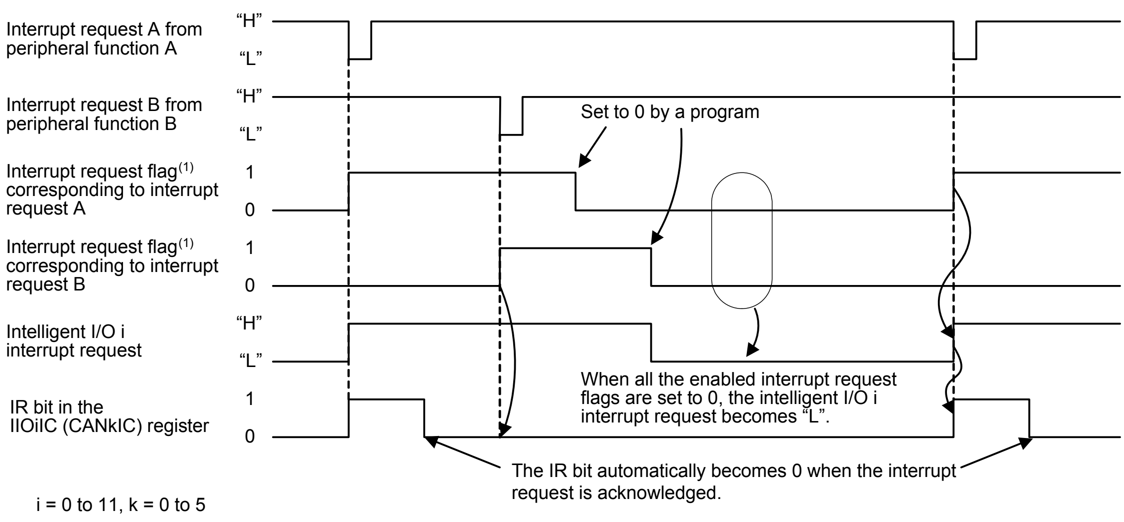
- IR bit
  The IR bit in the IIOiIC register becomes 1 (interrupt requested), if all the enabled request flags in the corresponding IIOiIR register are set to 0, and an interrupt request corresponding to one of these flags is generated. The IR bit automatically becomes 0 when the interrupt is acknowledged.

Table 11.7 lists registers used for CAN interrupts, UART5 and UART6 transmit/receive interrupts, and $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ interrupts. Figure 11.20 shows an interrupt request bit timing with multiple interrupt sources. Figure 11.21 shows an interrupt routine example.

Table 11.7 Registers Used for CAN interrupts, UART5 and UART6 transmit/receive interrupts, and INT6 to INT8 interrupts

Interrupts shared with Intelligent I/O Interrupt			Registers to be Used[2]		
CAN Interrupt[1]	UART Transmit/receive	$\overline{\text{INT}}$ Interrupt			
CAN00	UART6 receive	$\overline{\text{INT6}}$	IIO9IE	IIO9IR	IIO9IC (CAN0IC)
CAN01	UART6 transmit	$\overline{\text{INT7}}$	IIO10IE	IIO10IR	IIO10IC (CAN1IC)
CAN02	–	$\overline{\text{INT8}}$	IIO11IE	IIO11IR	IIO11IC (CAN2IC)
CAN10	UART5 receive	–	IIO0IE	IIO0IR	IIO0IC (CAN3IC)
CAN11	UART5 transmit	–	IIO1IE	IIO1IR	IIO1IC (CAN4IC)
CAN12 CAN1 Wake-up	–	–	IIO5IE	IIO5IR	IIO5IC (CAN5IC)

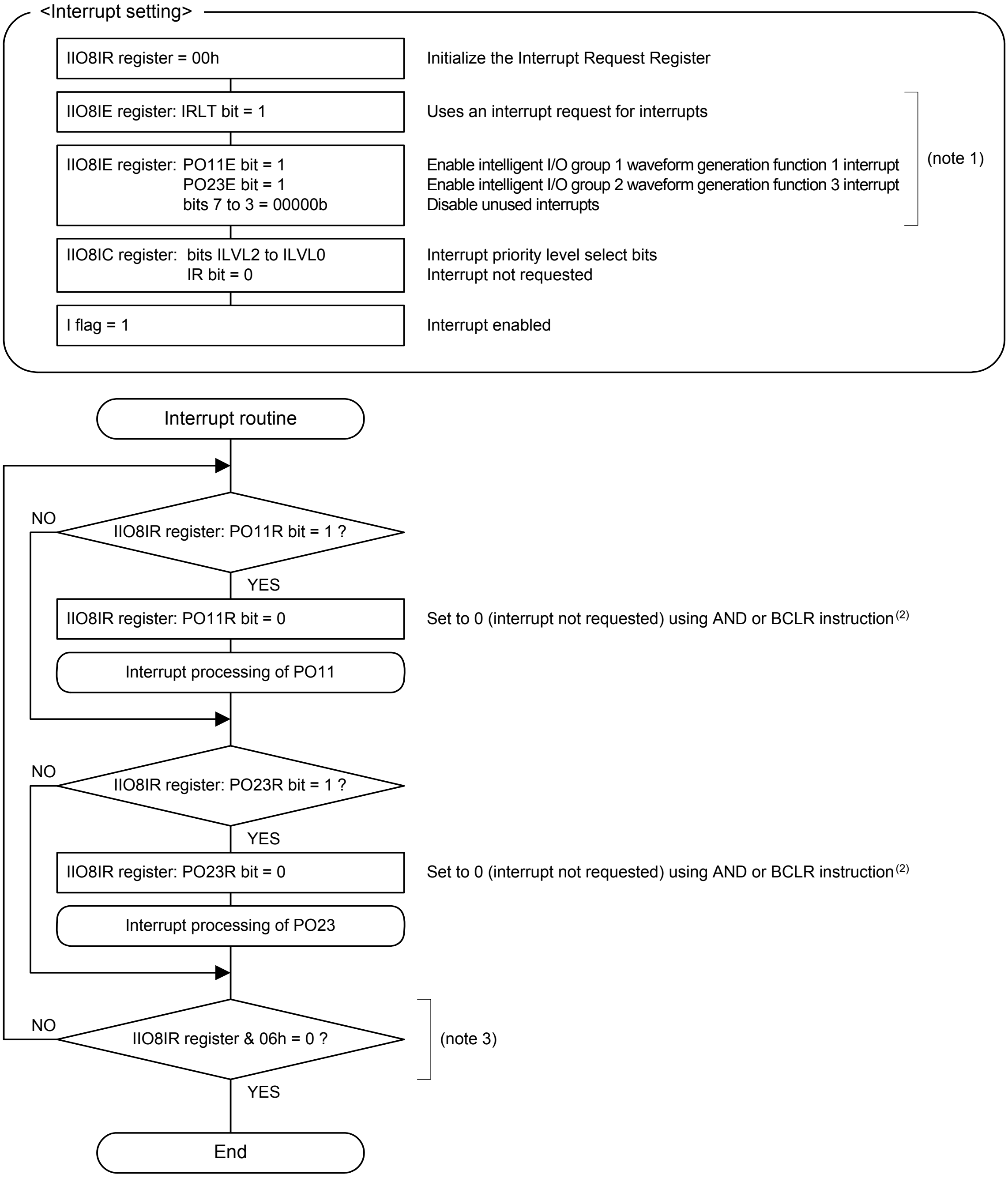
NOTES:
1. Only CAN00 to CAN02 interrupts can be used in M32C/87A. No CAN interrupt is provided in M32C/87B.
2. The IIO9IC register and the CAN0IC register share the same address. So do the IIO10IC register and CAN1IC register, the IIO11IC register and the CAN2IC register, the IIO0IC register and the CAN3IC register, the IIO1IC register and the CAN4IC register, and the IIO5IC register and the CAN5IC register.



NOTE:
1. These interrupt request flags are assigned to the same IIOiIR register and both flags are enabled in the IIOiIE register.

Figure 11.20 Interrupt Request Bit Timing with Multiple Interrupt Sources

Example: Intelligent I/O Group 1 Waveform Generation Function 1 Interrupt, Group 2 Waveform Generation Function 3 Interrupt are used.



NOTES:

1. Do not change the interrupt enable bit (bits 1 to 7 in the IIOiIE register (i = 0 to 11)) and the IRLT bit in the IIOiIE register simultaneously. Set the IRLT bit to 1 first, and then set the interrupt enable bit to 1.
2. If an interrupt request is generated while writing a 0 to the corresponding interrupt request flag, the flag may not be cleared to 0. In this case, keep writing a 0 until 0 is read.
3. Ensure that all the enabled interrupt request flags are set to 0. If any of these flags remains 1, the IR bit in the IIOiIC (CANkIC (k = 0 to 5)) register does not become 1 when an interrupt request is generated in the same register. (Interrupt does not occur.)

Figure 11.21 Interrupt Routine Example

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 Vdet4 detection interrupt. When using the watchdog timer interrupt simultaneously with these interrupts, determine whether the watchdog timer interrupt is generated by reading the D43 bit in the D4INT register in the interrupt routine.

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.

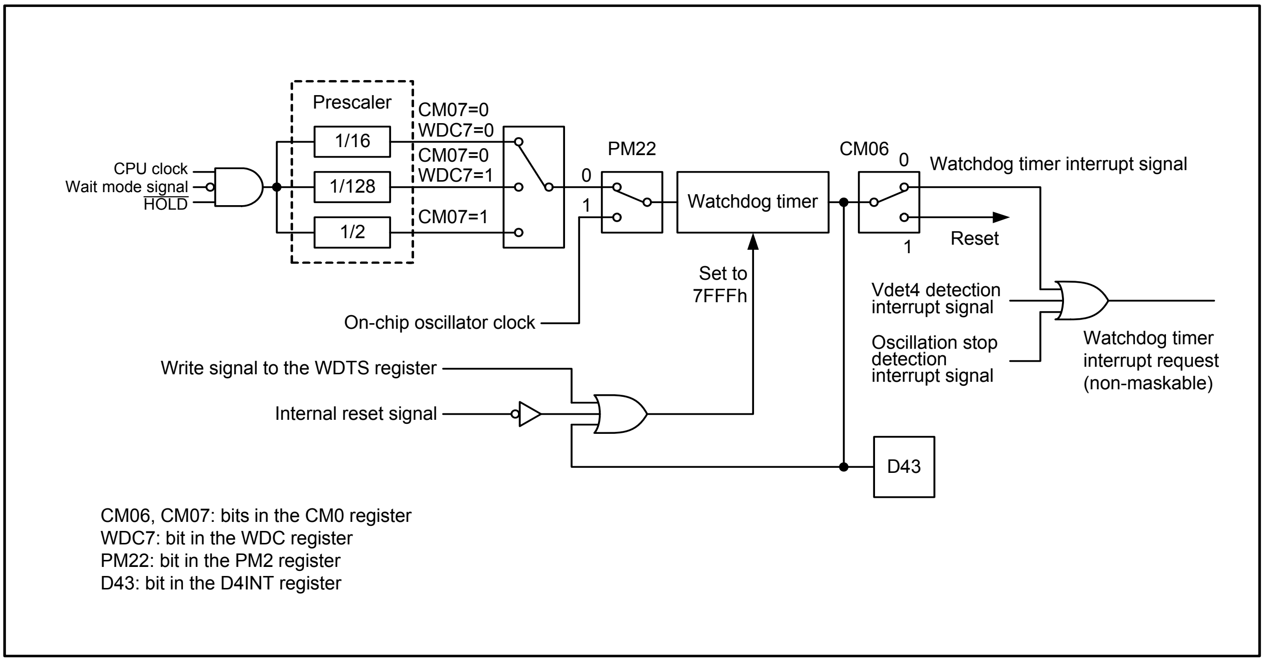
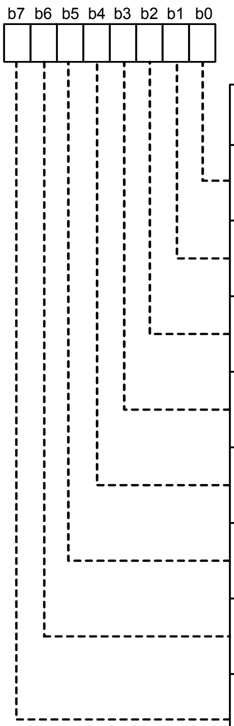


Figure 12.1 Watchdog Timer Block Diagram

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 in memory expansion mode or microprocessor mode. Port 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

b7 b6 b5 b4 b3 b2 b1 b0

0 (b6)

Symbol	Address	After Reset
WDC	000Fh	00XX XXXXb

Bit Symbol	Bit Name	Function	RW
– (b4-b0)	High-order bits of watchdog timer		RO
WDC5	Cold start/warm start determine flag(1)	0: Cold start 1: Warm start	RW
– (b6)	Reserved bit	Set to 0	RW
WDC7	Prescaler select bit	0: Divide-by-16 1: Divide-by-128	RW

NOTES:
1. The WDC5 bit is 0 after power-on. It can be set to 1 only by a program. The bit becomes 1 by writing either a 0 or 1. The bit maintains a value set before reset, even after reset has been performed.

Watchdog Timer Start Register

b7 b0

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/87 Group (M32C/87, M32C/87A, M32C/87B) 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.

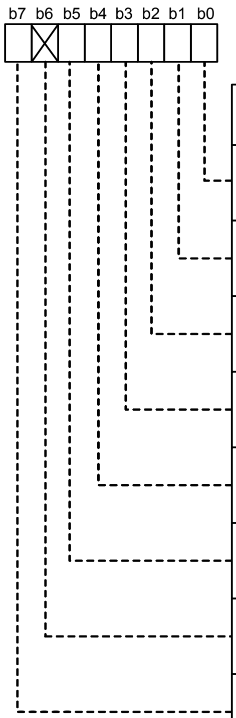
Table 13.1 DMAC Specifications

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 $\overline{\text{INT3}}$ • INT6 to INT8 interrupt requests • Timer A0 to A4 interrupt requests • Timer B0 to B5 interrupt requests • UART0 to UART6 transmit/receive interrupt requests • A/D0 interrupt request • Intelligent I/O interrupt request • CAN interrupt request(1) • 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

NOTE:
1. Only CAN00, CAN01, and CAN02 interrupt requests can be used for M32C/87A. Any CAN interrupt request cannot be used for M32C/87B.

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)"	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 INT0	Falling edge of INT1	Falling edge of INT2	Falling edge of INT3[1]	(Note 2)
0	0	0	1	0	Both edges of INT0	Both edges of INT1	Both edges of INT2	Both edges of 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				
1	1	0	0	1	Intelligent I/O interrupt 0 request[4]	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request[7]	
1	1	0	1	0	Intelligent I/O interrupt 1 request[5]	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request[8]	
1	1	0	1	1	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request[7]	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request[9]	
1	1	1	0	0	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request[8]	Intelligent I/O interrupt 5 request[6]	Intelligent I/O interrupt 0 request[4]	
1	1	1	0	1	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request[9]	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request[5]	
1	1	1	1	0	Intelligent I/O interrupt 5 request[6]	Intelligent I/O interrupt 0 request[4]	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request	
1	1	1	1	1	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request[5]	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request	

NOTES:
1. When the 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 INT3 pin.
2. The falling edge or both edges of input signal to the INTi pin can be a DMA request source. It is not affected by the 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^2C mode) and the IICM2 bit to 0 (NACK/ACK interrupt).
4. The same setting is used for a CAN10 interrupt request and a UART5 receive interrupt request.
5. The same setting is used for a CAN11 interrupt request and a UART5 transmit interrupt request.
6. The same setting is used for a CAN12 interrupt request.
7. The same setting is used for a CAN00 interrupt request, an INT6 interrupt request, and a UART6 receive interrupt request.
8. The same setting is used for a CAN01 interrupt request, an INT7 interrupt request, and a UART6 transmit interrupt request.
9. The same setting is used for a CAN02 interrupt request and INT8 interrupt request.

DMAi Memory Address Register (i = 0 to 3)

b23 b16 b15 b8 b7 b0

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)

b23 b16 b15 b8 b7 b0

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)

b23 b16 b15 b8 b7 b0

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)

b15 b8 b7 b0

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)

b15 b8 b7 b0

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]

b7 b6 b5 b4 b3 b2 b1 b0

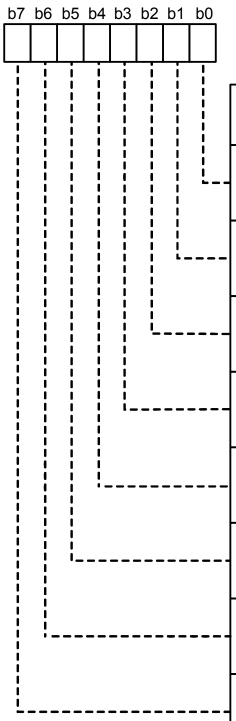
Symbol	Address	After Reset
DMD0	(CPU internal register)	00h

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]

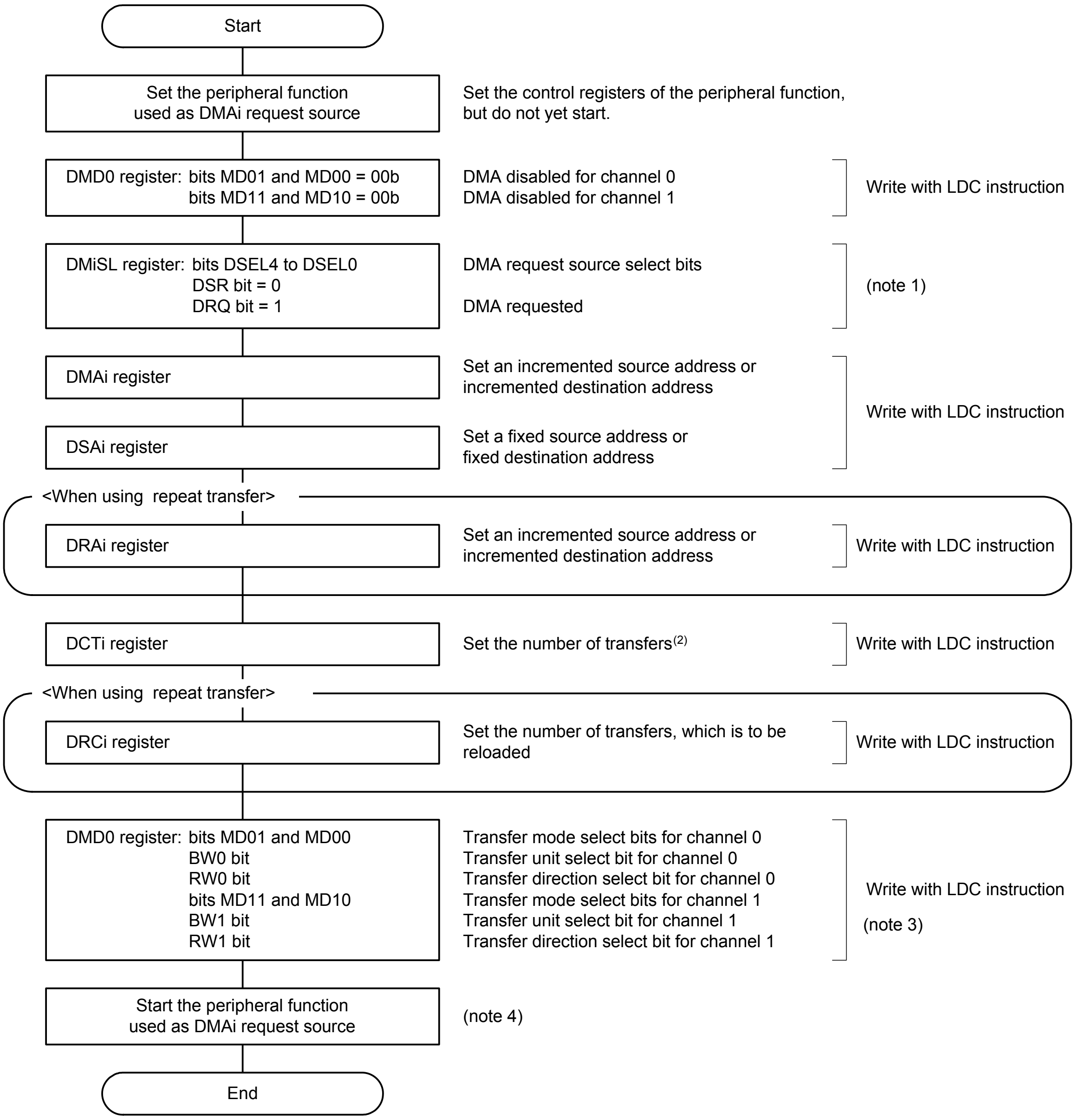


Symbol	Address	After Reset
DMD1	(CPU internal register)	00h

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

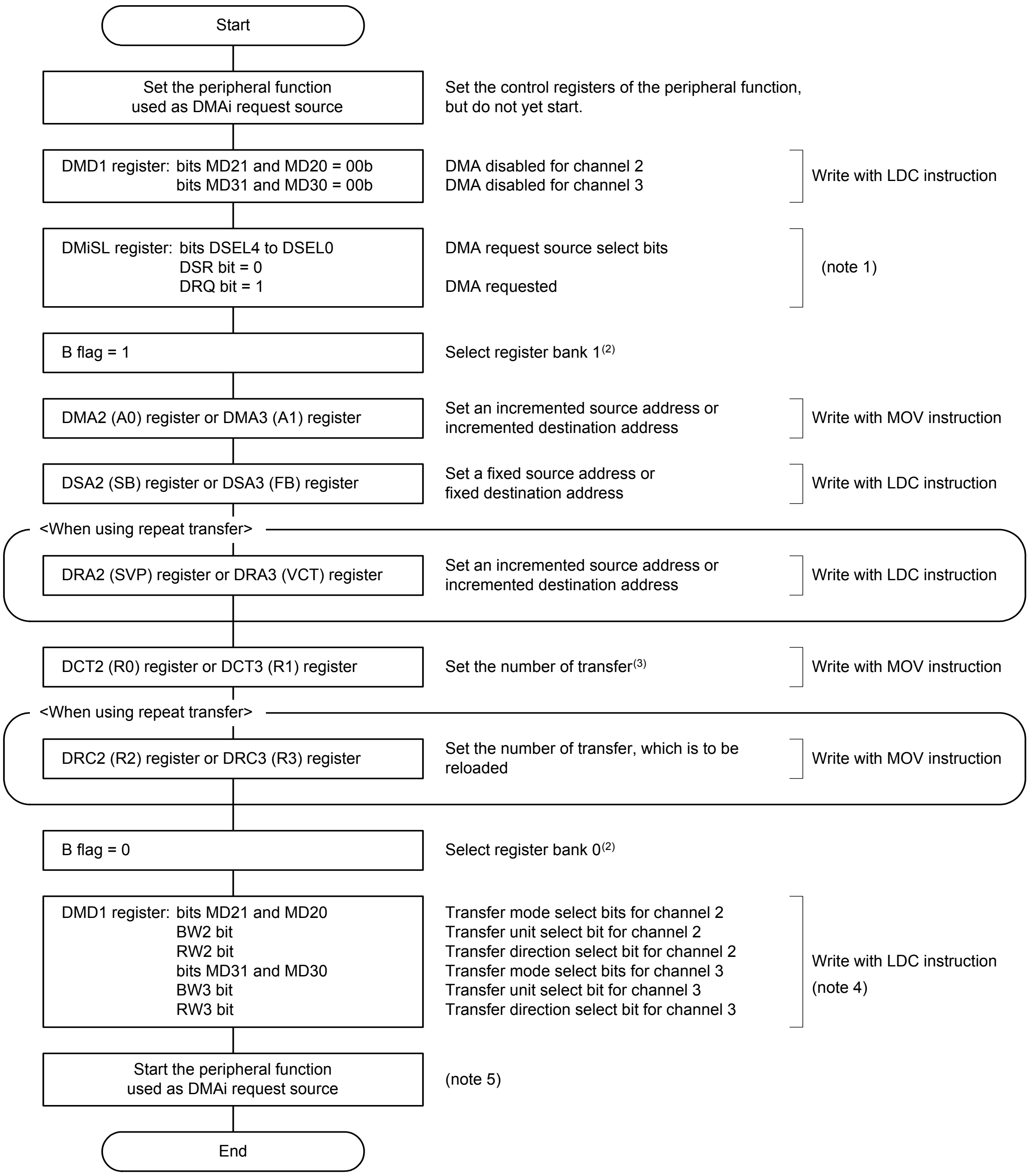


i = 0, 1

NOTES:

1. When setting the DMiSL register, write a 1 to the DRQ bit.
2. When the $\overline{INT}$ 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



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 $\overline{\text{RDY}}$ 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 $\overline{\text{RDY}}$ Signal

In memory expansion mode and microprocessor mode, the $\overline{\text{RDY}}$ 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 $\overline{\text{RDY}}$ 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

<table>
<tr><th colspan="3">Internal Space</th><th>External Space</th></tr>
<tr><td>Internal ROM or internal RAM</td><td>Internal ROM or internal RAM</td><td>SFR area</td><td rowspan="2">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</td></tr>
<tr><td>with no wait state
j=1
k=1</td><td>with wait state
j=2
k=2</td><td>j=2
k=2</td></tr>
</table>

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.

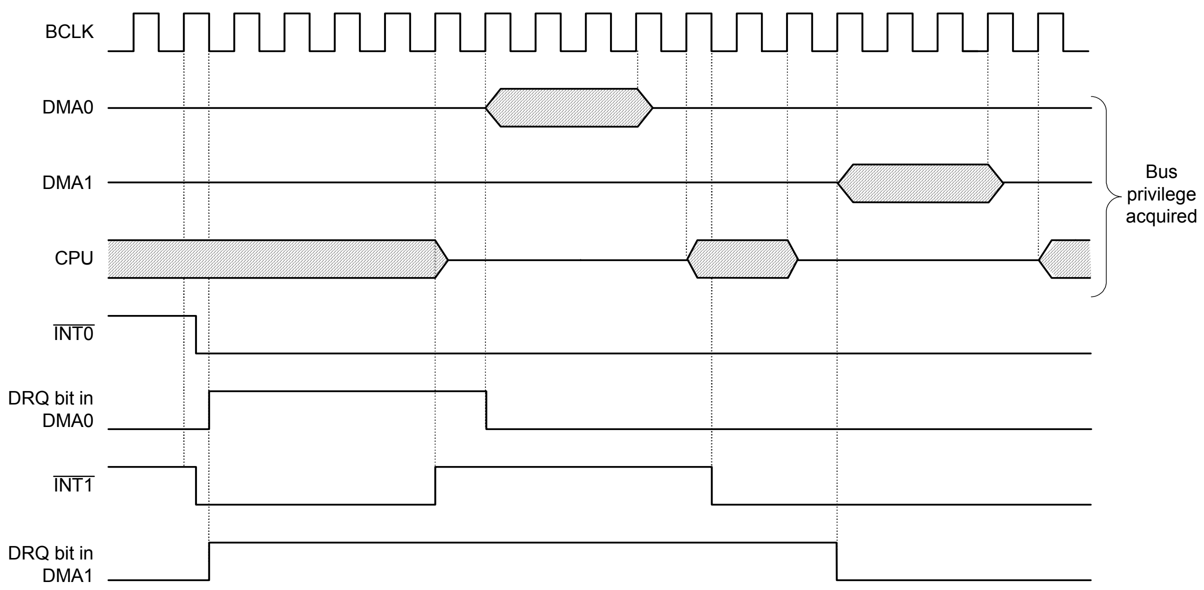


Figure 13.9 DMA Transfers Triggered by $\overline{INT}$ Interrupt Requests

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

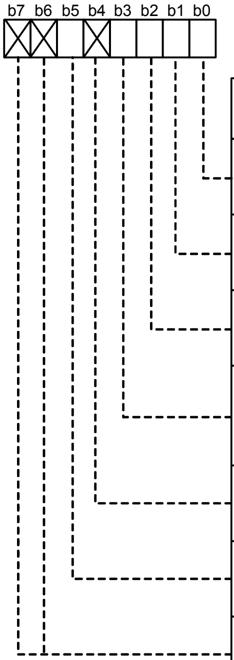
Set up the following registers and tables to activate DMACII.

- 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
- IRLT bit in the IIOiIE register (i = 0 to 11) if using the intelligent I/O interrupt, CAN interrupt, $\overline{\text{INTj}}$ interrupt (j = 6 to 8), UARTk (k = 5, 6) transmit, or UARTk receive interrupt. Refer to **11. Interrupts** for details on the IIOiIE register.

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 Starting Address	16 bits
(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.

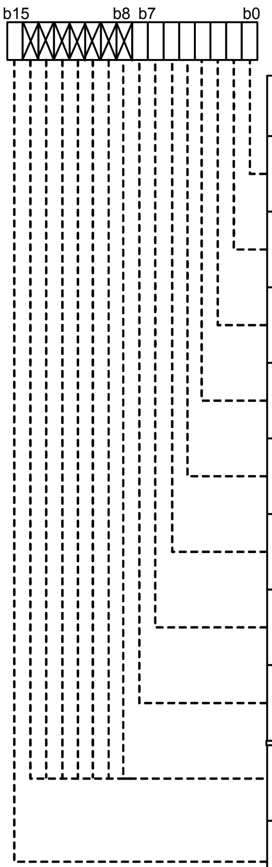
- Transfer mode (MOD)
  MOD is two-byte data and required to set transfer mode. Figure 14.3 shows a configuration for transfer mode.
- Transfer counter (COUNT)
  COUNT is two-byte data and required to set the number of transfer.
- Transfer source address (SADR)
  SADR is two-byte data and required to 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.

14.1.5 IRLT Bit in the IIOiIE Register (i = 0 to 11)

When the intelligent I/O interrupt, CAN interrupt, $\overline{\text{INTj}}$ interrupt (j = 6 to 8), UARTk (k = 5, 6) transmit interrupt, or UARTk receive interrupt is used to activate DMACII, set the IRLT bit in the corresponding IIOiIE register (i = 0 to 11) to 0 (interrupt request is used for DMAC, DMACII).

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.

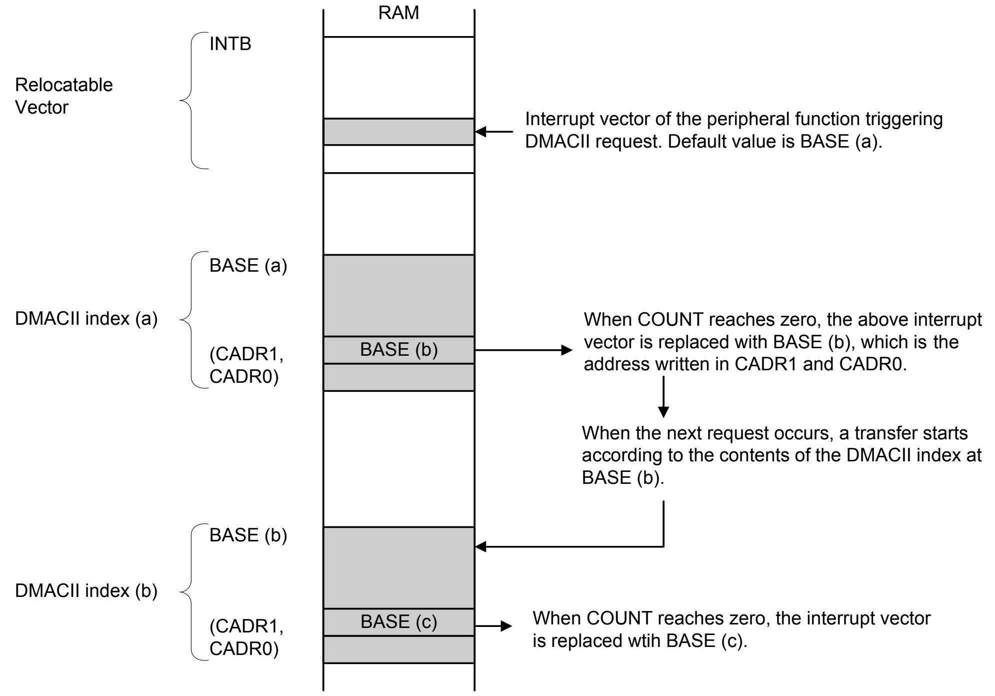


Figure 14.4 Relocatable Vector and DMACII Index When using the Chain Transfer

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

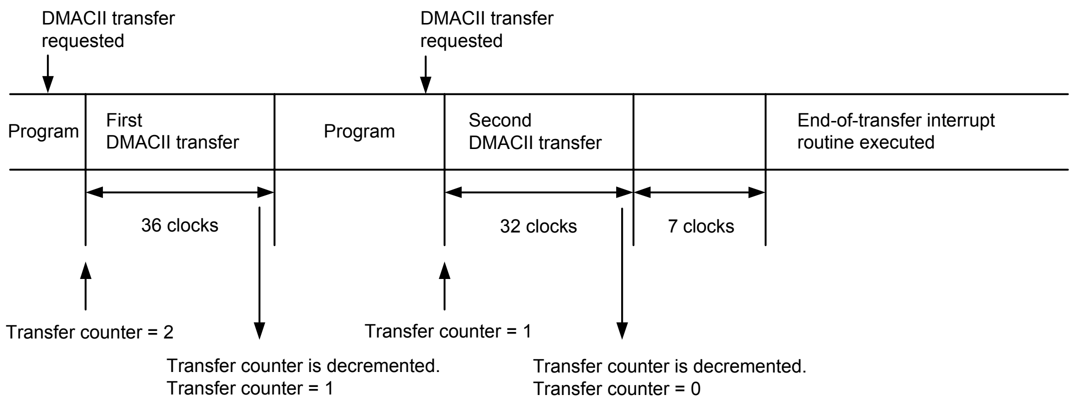


Figure 14.5 Transfer Time

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/87 Group (M32C/87, M32C/87A, M32C/87B) 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.

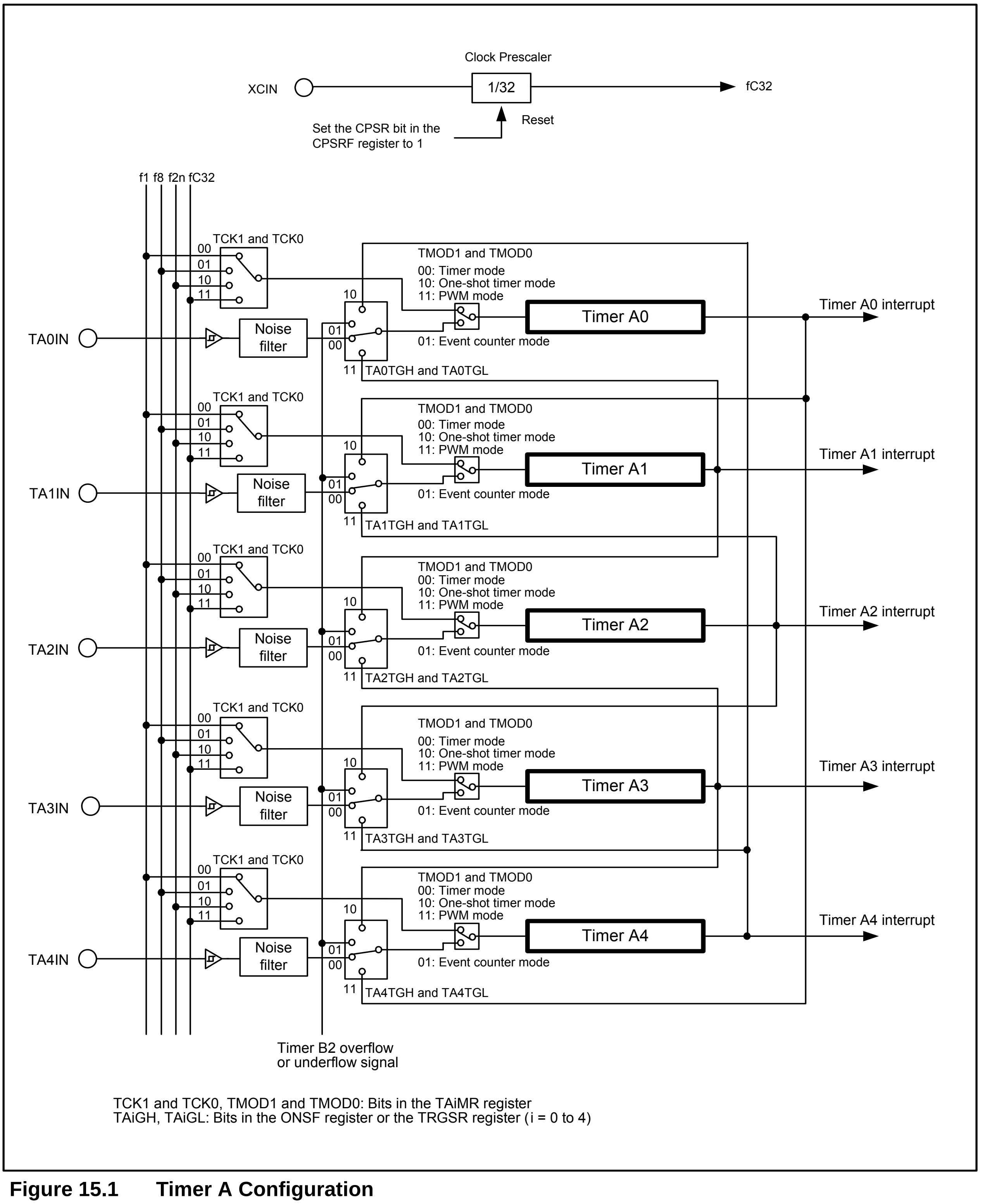


Figure 15.1 Timer A Configuration

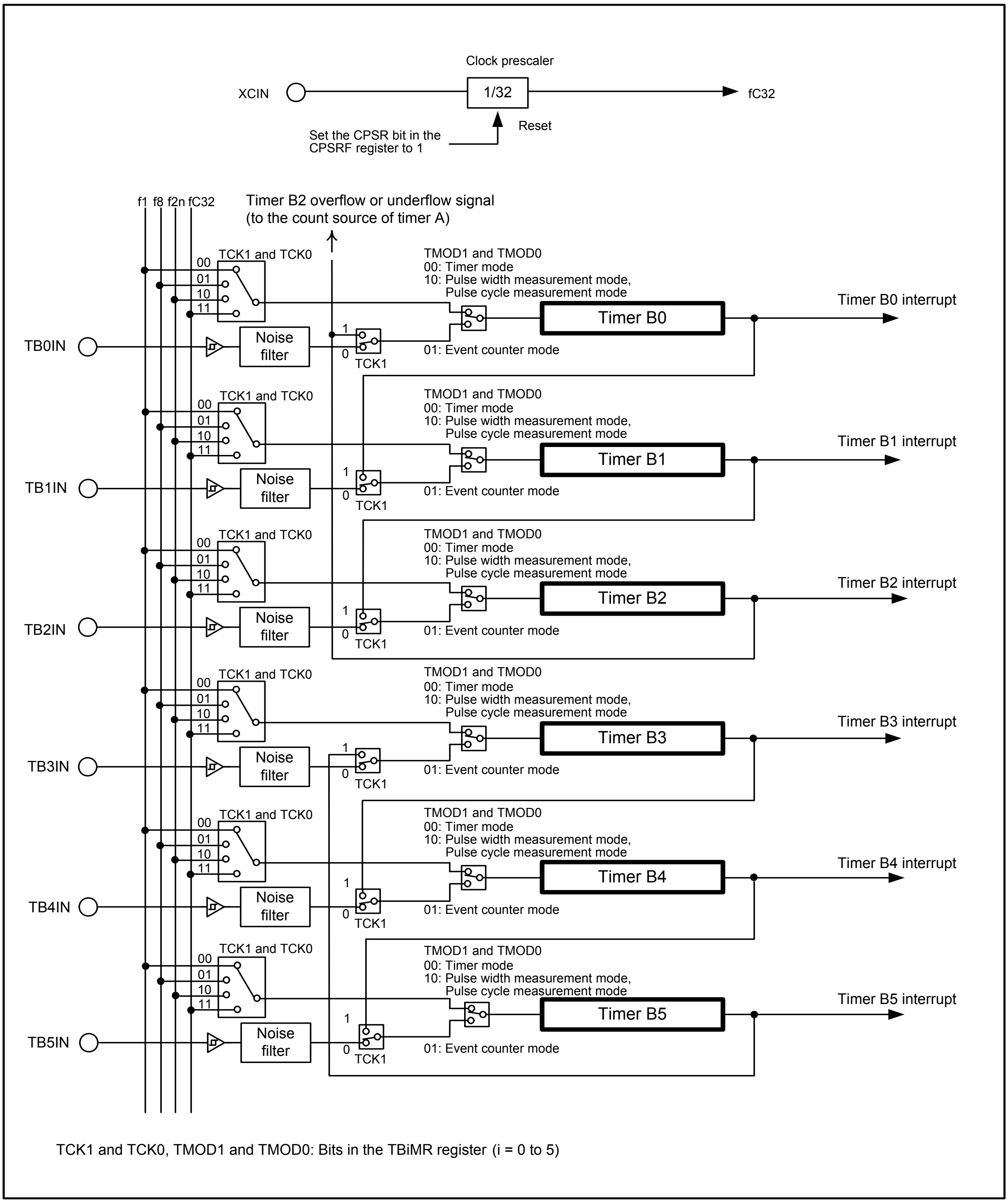


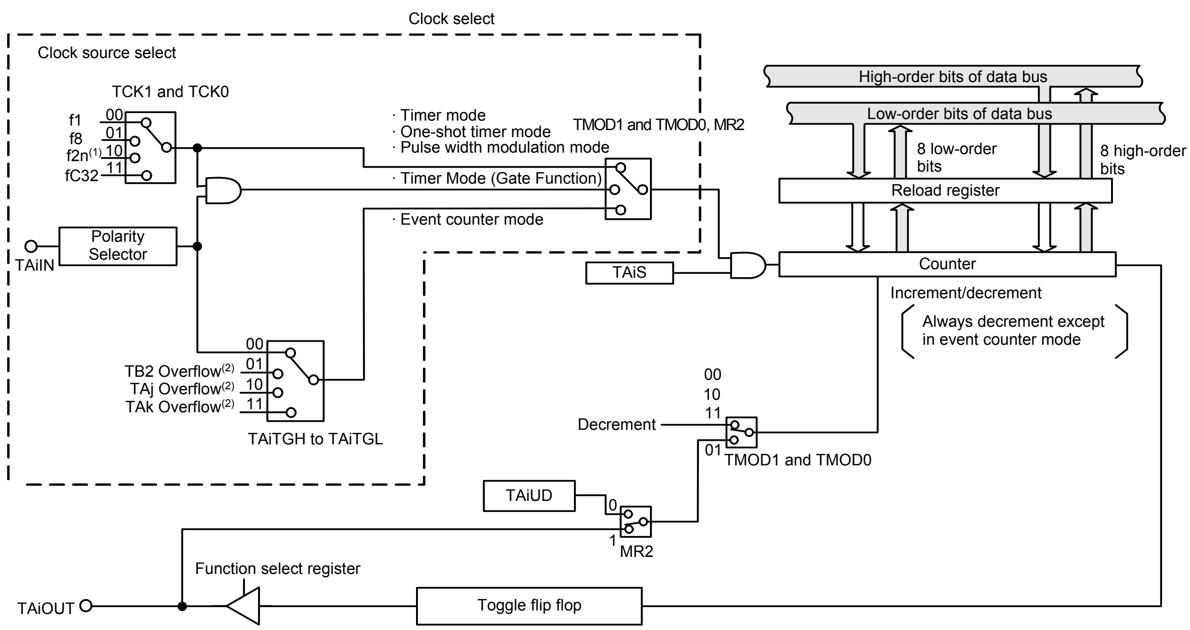
Figure 15.2 Timer B Configuration

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

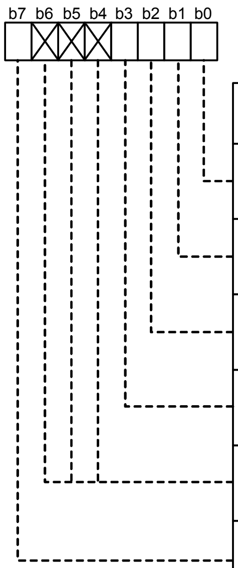
NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Overflow signal or underflow signal.

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)

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

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 registser 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)

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

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)

b7	b6	b5	b4	b3	b2	b1	b0
					0	1	1

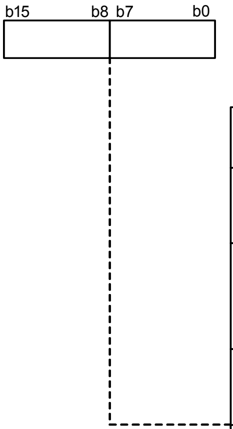
Symbol	Address	After Reset
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	00h

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]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
UDF	0344h	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

b7 b6 b5 b4 b3 b2 b1 b0

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

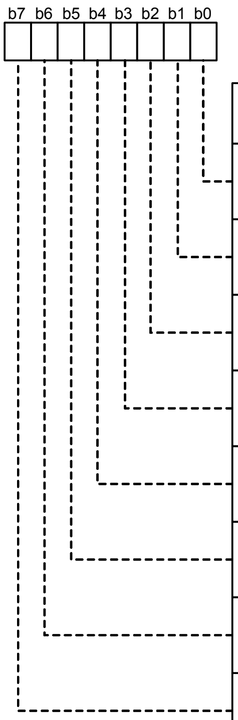
b7 b6 b5 b4 b3 b2 b1 b0

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.

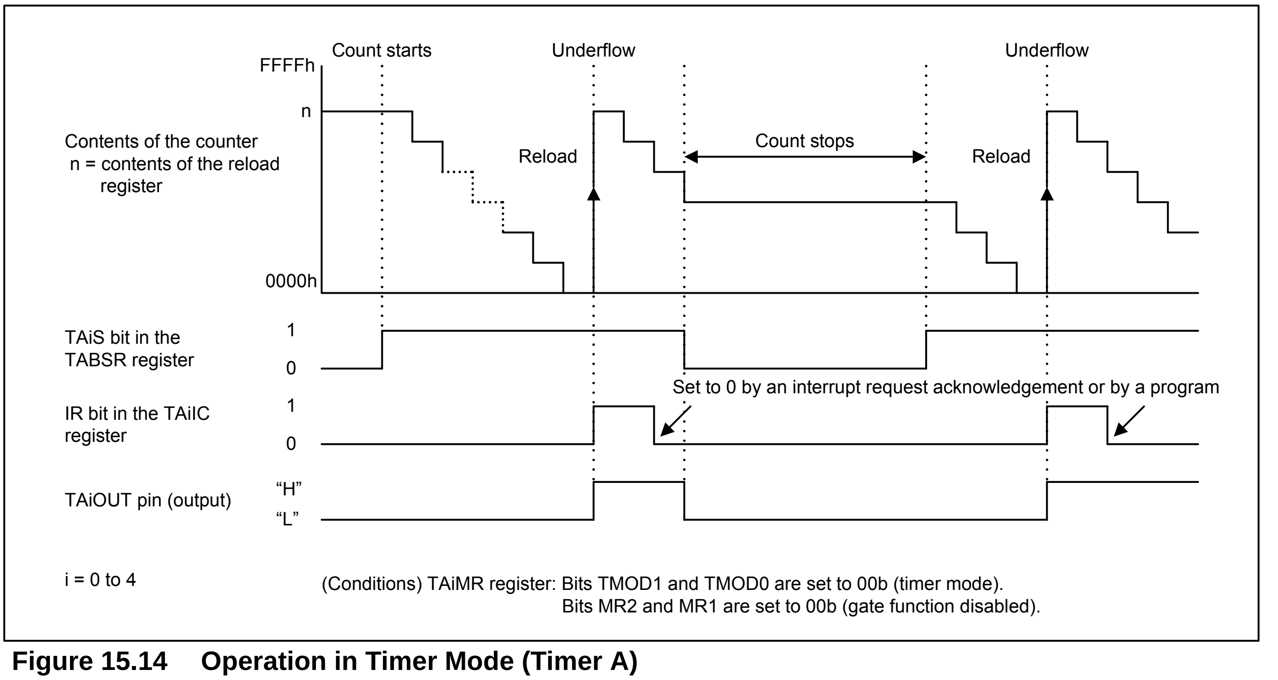


Figure 15.14 Operation in Timer Mode (Timer A)

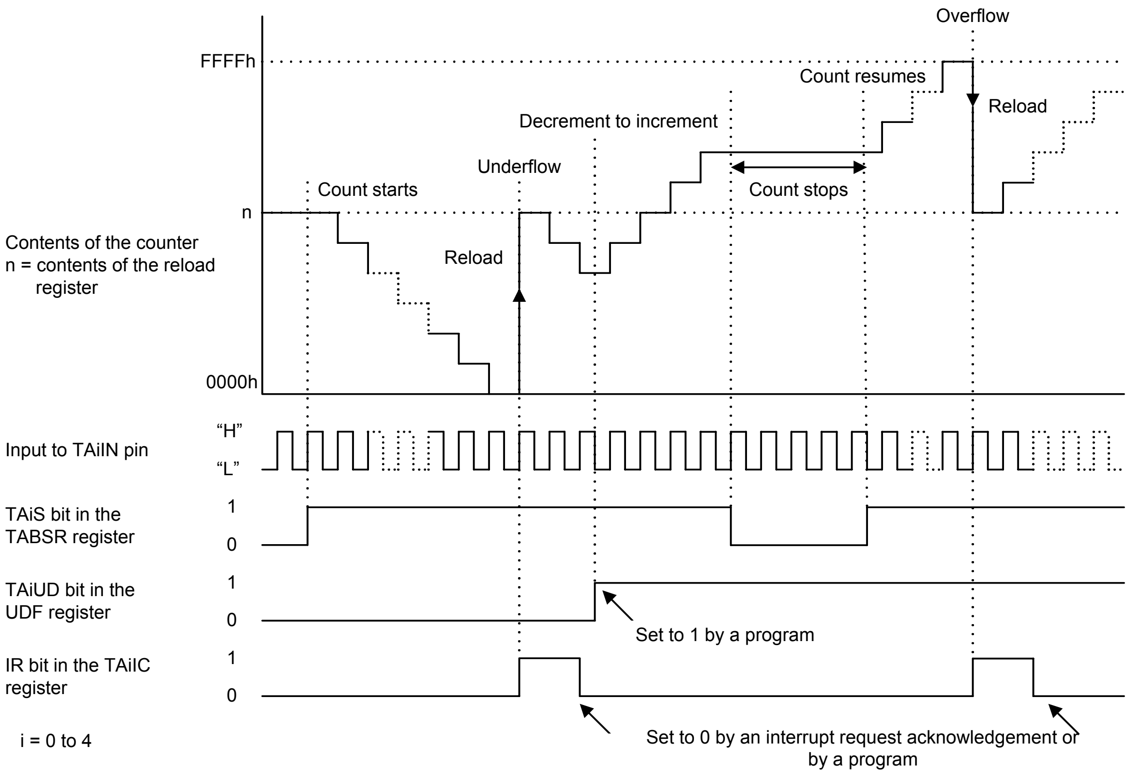
15.1.2 Event Counter Mode

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.



(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.

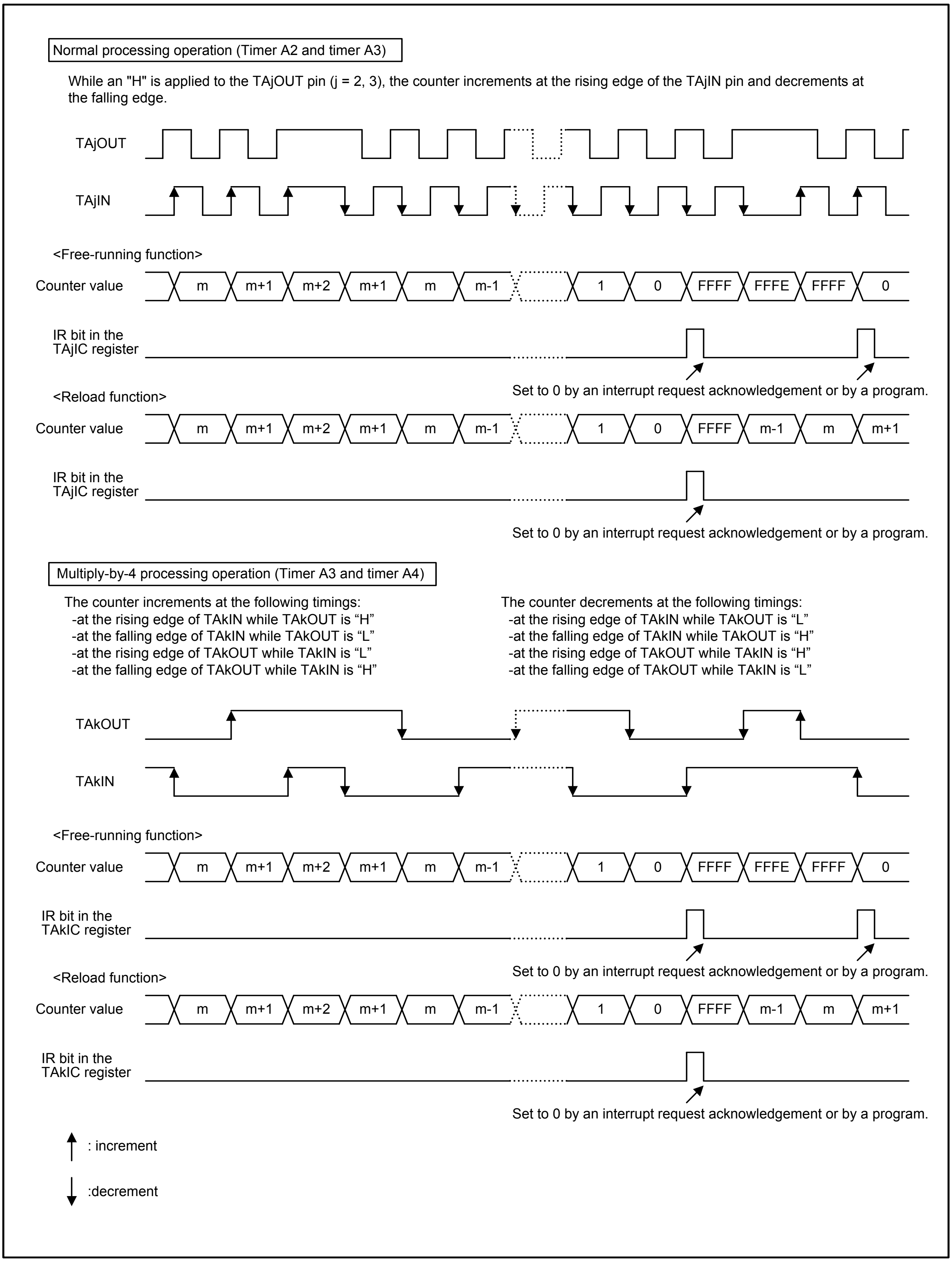


Figure 15.16 Operation in Event Counter Mode When Handling Two-Phase Pulse Signals on Timers A2, A3, and A4

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.

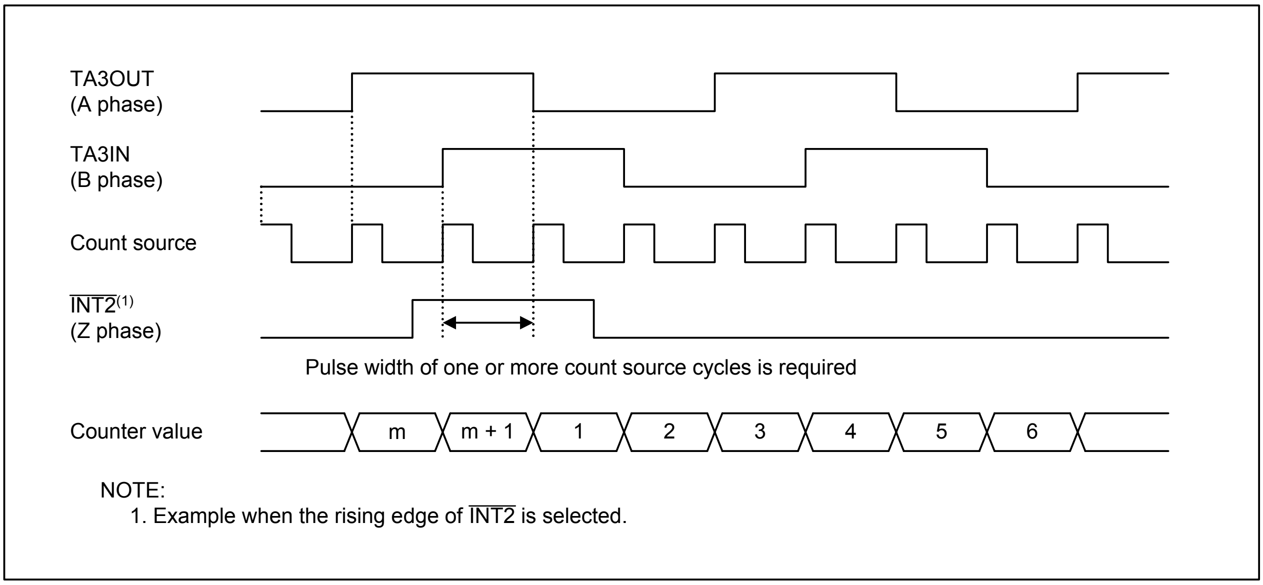


Figure 15.17 Relations between Two-Phase Pulses (A-Phase and B-Phase) and Z-Phase Pulse

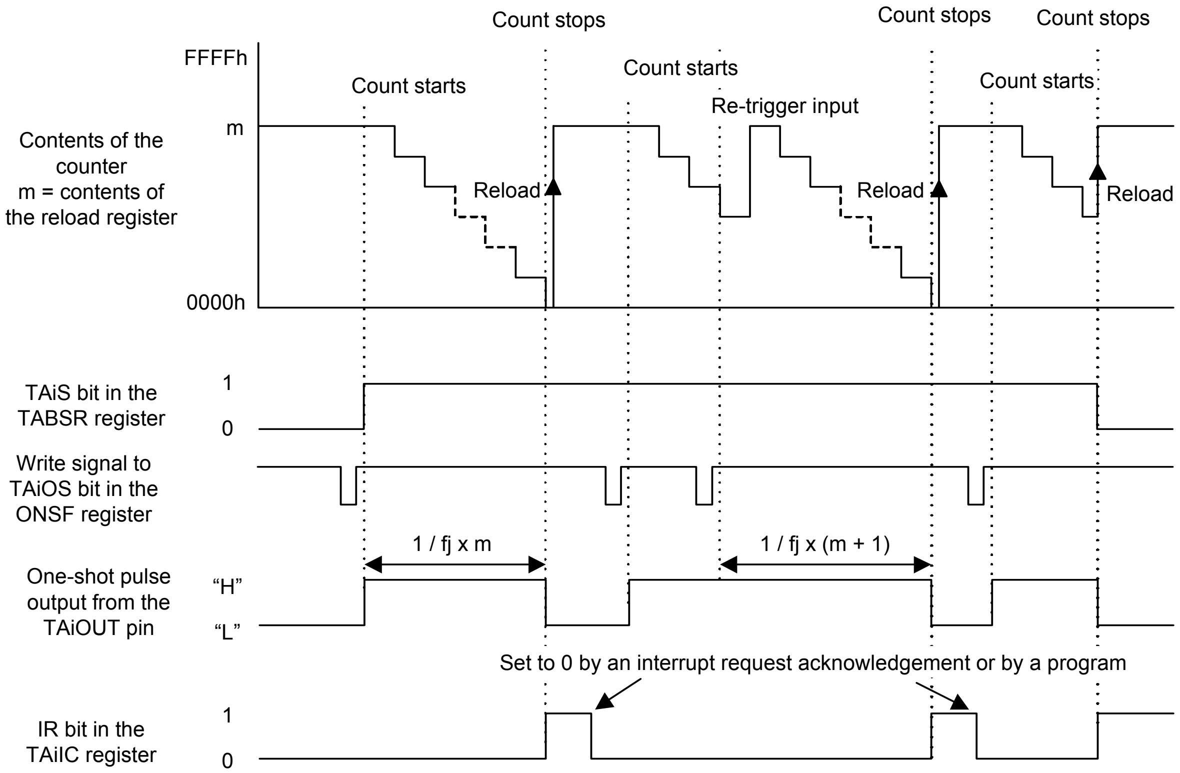
15.1.3 One-Shot Timer Mode

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.



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

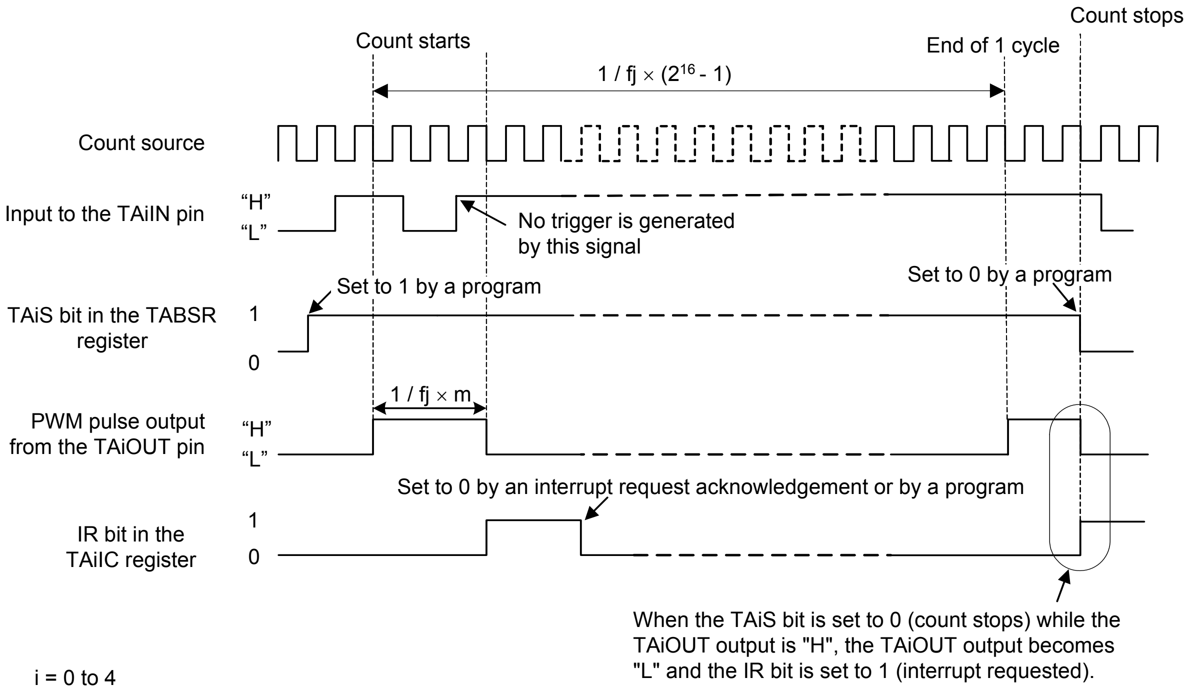
In pulse width modulation mode, the timer outputs pulse signals of a given width repeatedly. The counter functions as an 8-bit pulse width modulator or 16-bit pulse width modulator.

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.

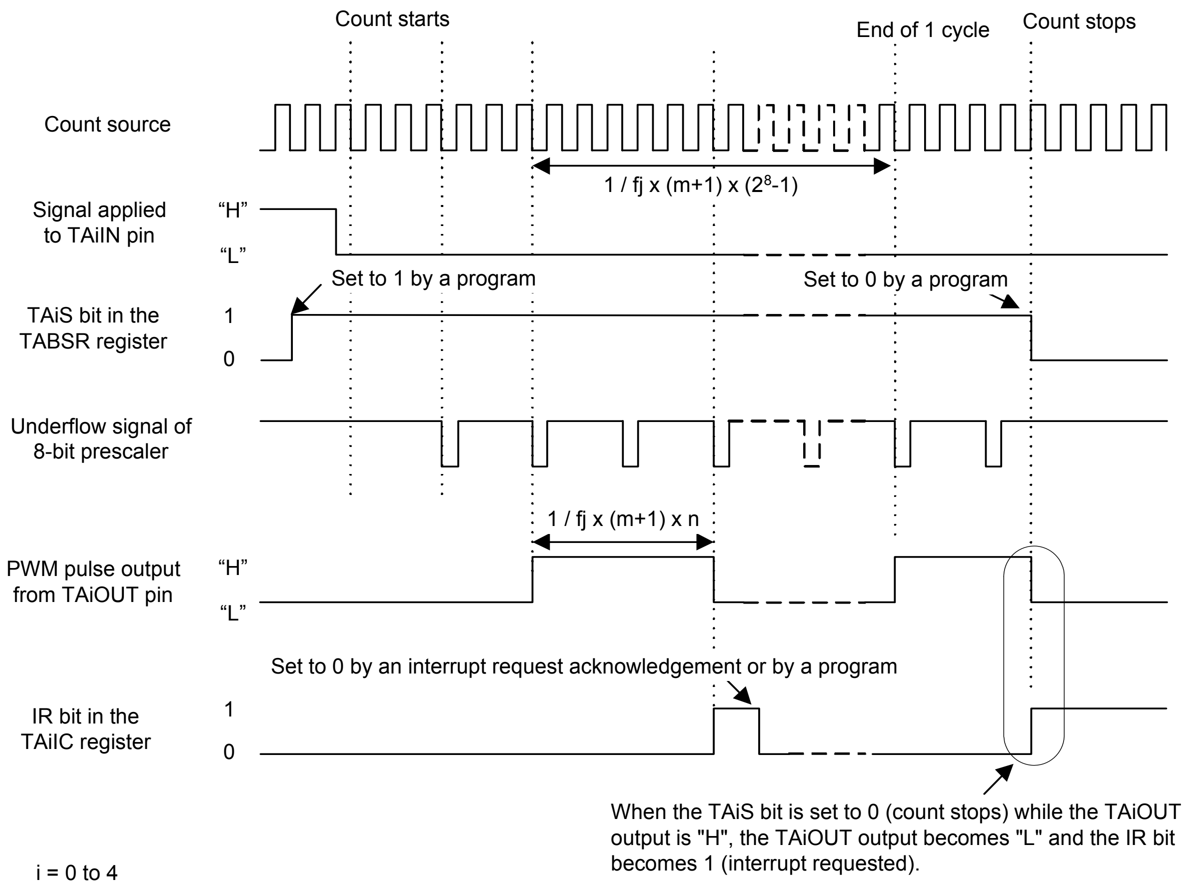


i = 0 to 4
fj: Count source frequency (f1, f8, f2n(1), fC32)
m: Setting value of the TAi register (0000h to FFFEh)

(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)



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.

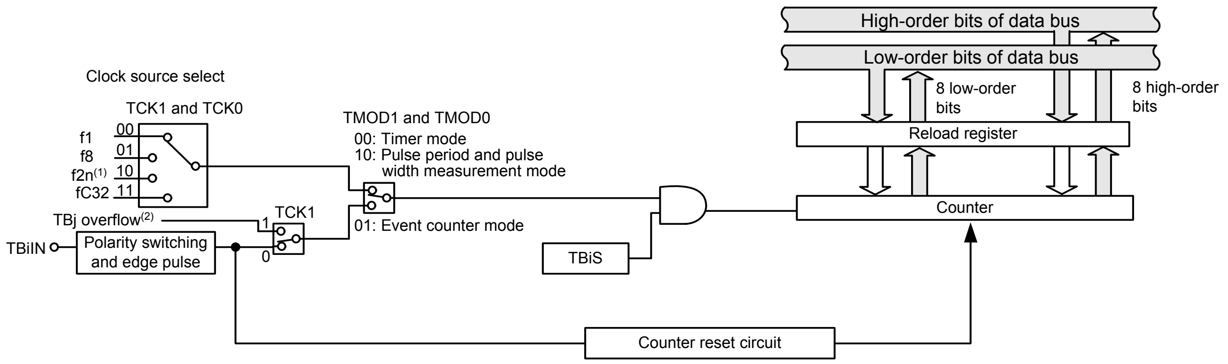
Figure 15.20 8-bit Pulse Width Modulator Operation (Timer A)

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)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

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)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

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)

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

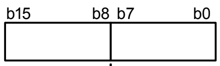
Symbol	Address	After Reset
TB0MR to TB5MR	035Bh, 035Ch, 035Dh, 031Bh, 031Ch, 031Dh	00XX 0000b

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)



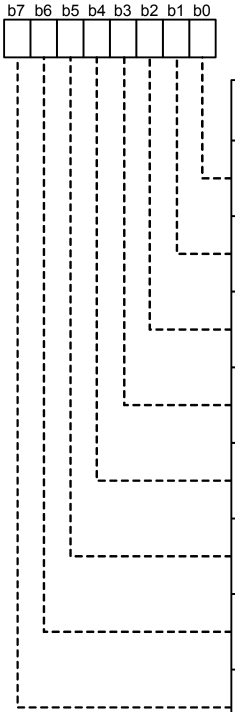
Symbol	Address	After Reset
TB0 to TB2	0351h - 0350h, 0353h - 0352h, 0355h - 0354h	Undefined
TB3 to TB5	0311h - 0310h, 0313h - 0312h, 0315h - 0314h	Undefined

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

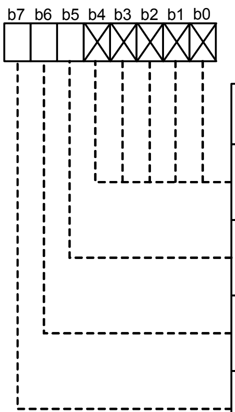
Count Start Register



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.

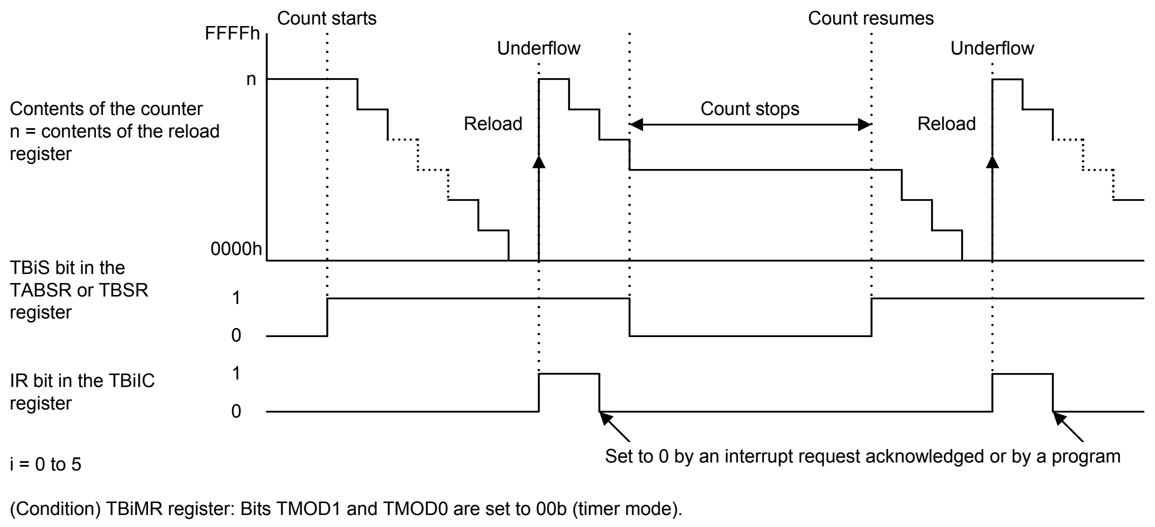


Figure 15.27 Operation in Timer Mode (Timer B)

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.

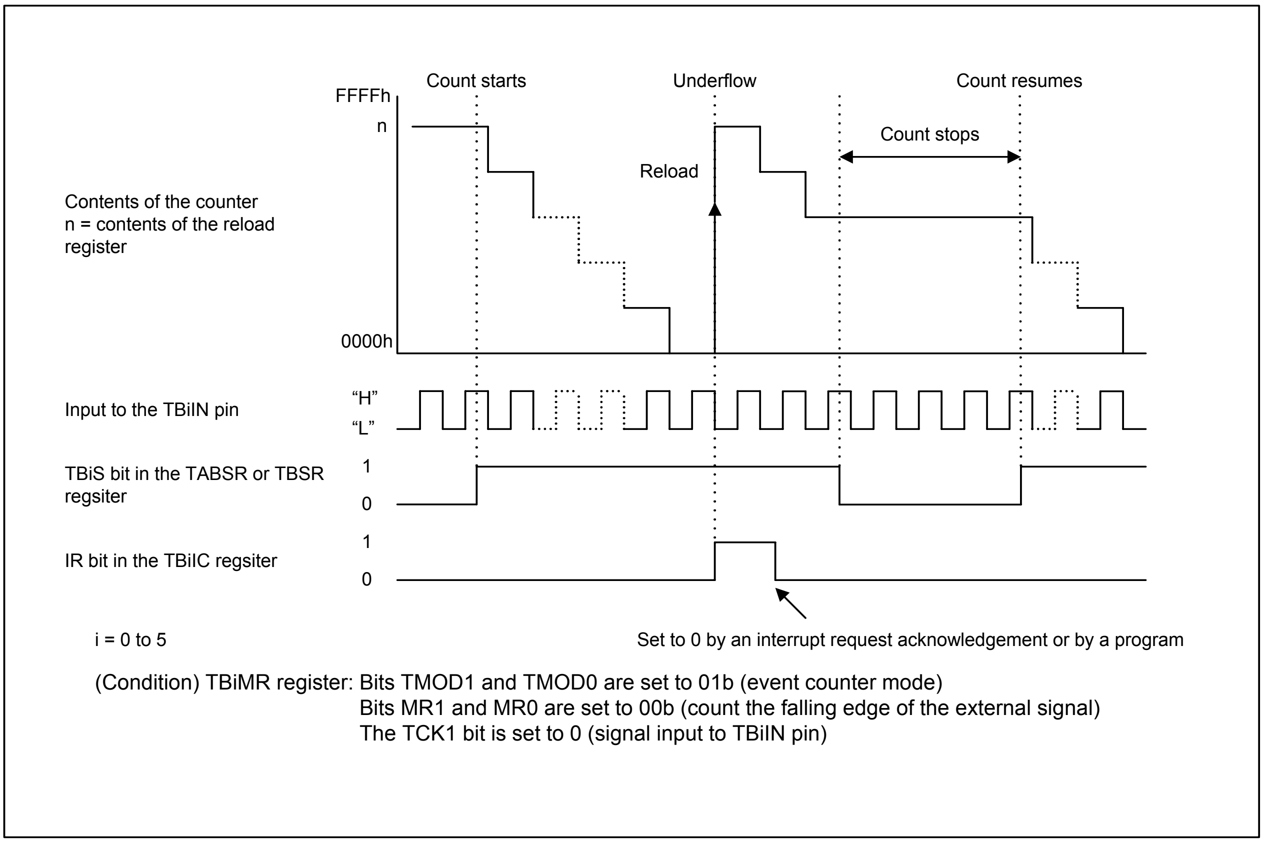


Figure 15.28 Operation in Event Counter Mode (Timer B)

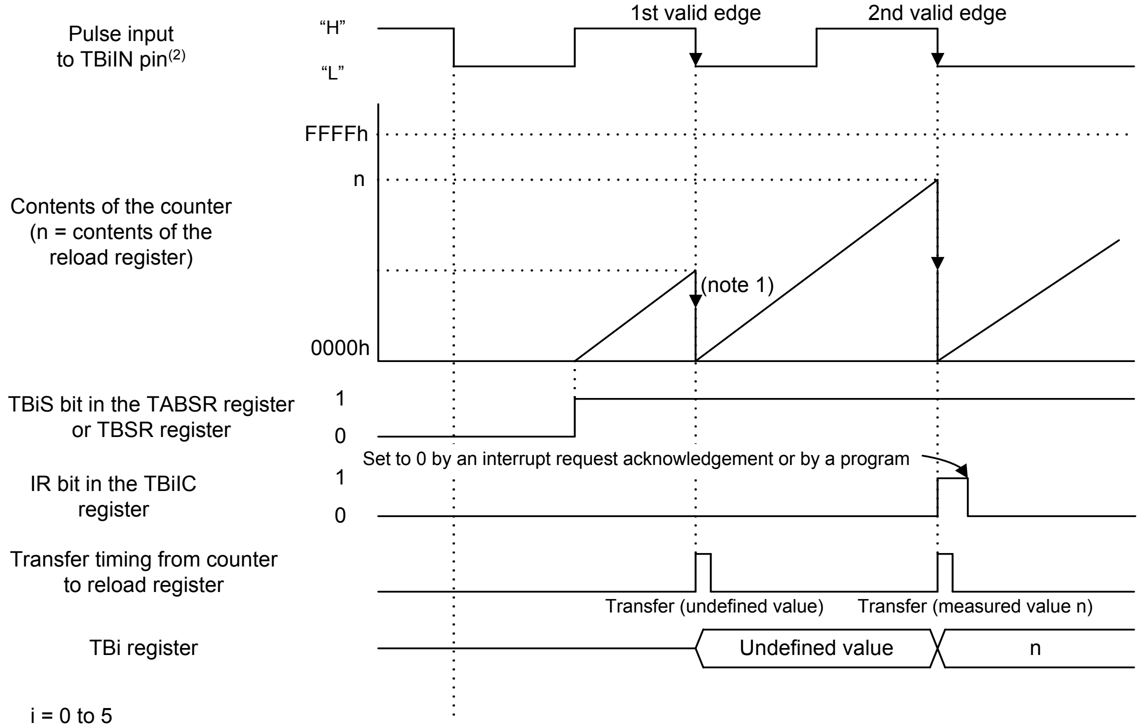
15.2.3 Pulse Period Measurement Mode, Pulse Width Measurement Mode

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

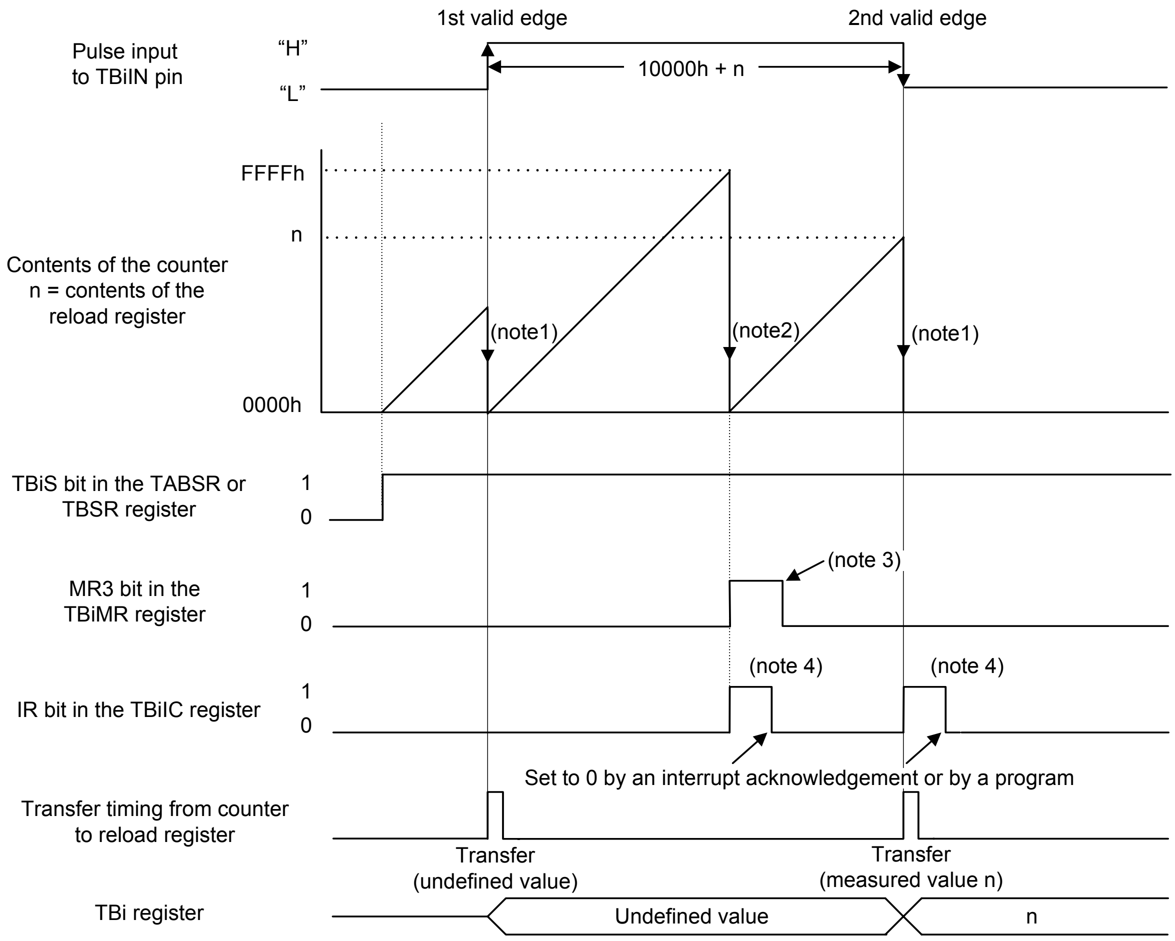
NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. An interrupt request is not generated when the first valid edge is input after the count starts.
3. To set the MR3 bit to 0 (no overflow), wait for one or more count source cycles to write to 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.

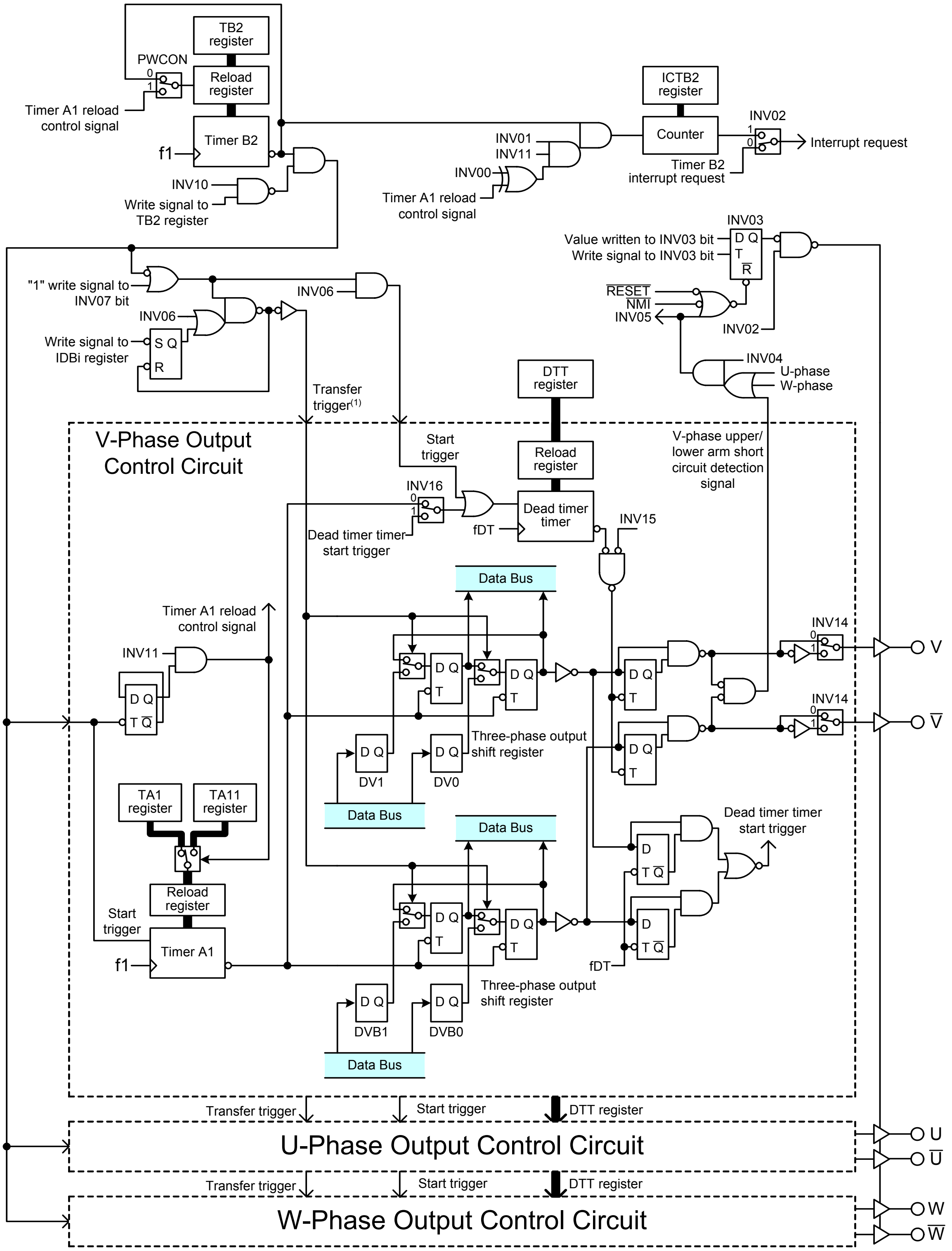
Figure 15.30 Operation in Pulse Width Measurement Mode (Timer B)

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 $\overline{NMI}$ input



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

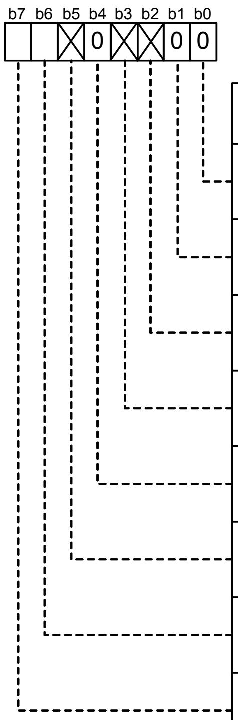
Bit Symbol	Bit Name	Function	RW
INV10	Timers A1, A2, and A4 start trigger select bit	0: Timer B2 underflow 1: Timer B2 underflow and a write to the TB2 register	RW
INV11	Timers A11, A21, and A41 control bit	0: Timers A11, A21, and A41 not used (Three-phase mode 0) 1: Timers A11, A21, and A41 used (Three-phase 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

b7 b6 b5 b4 b3 b2 b1 b0

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[1]

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

Symbol	Address	After Reset
TB2SC	035Eh	00h

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 reload timing switch bit	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)	Reserved bits	Set to 0	RW

NOTE:
1. Set the TB2SC register after the PRC1 bit in the PRCR register is set to 1 (write enable).

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

b7 b6 b5 b4 b3 b0

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]

b15 b8 b7 b0

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]

b7 b0

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

b7 b6 b5 b4 b3 b2 b1 b0

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 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.

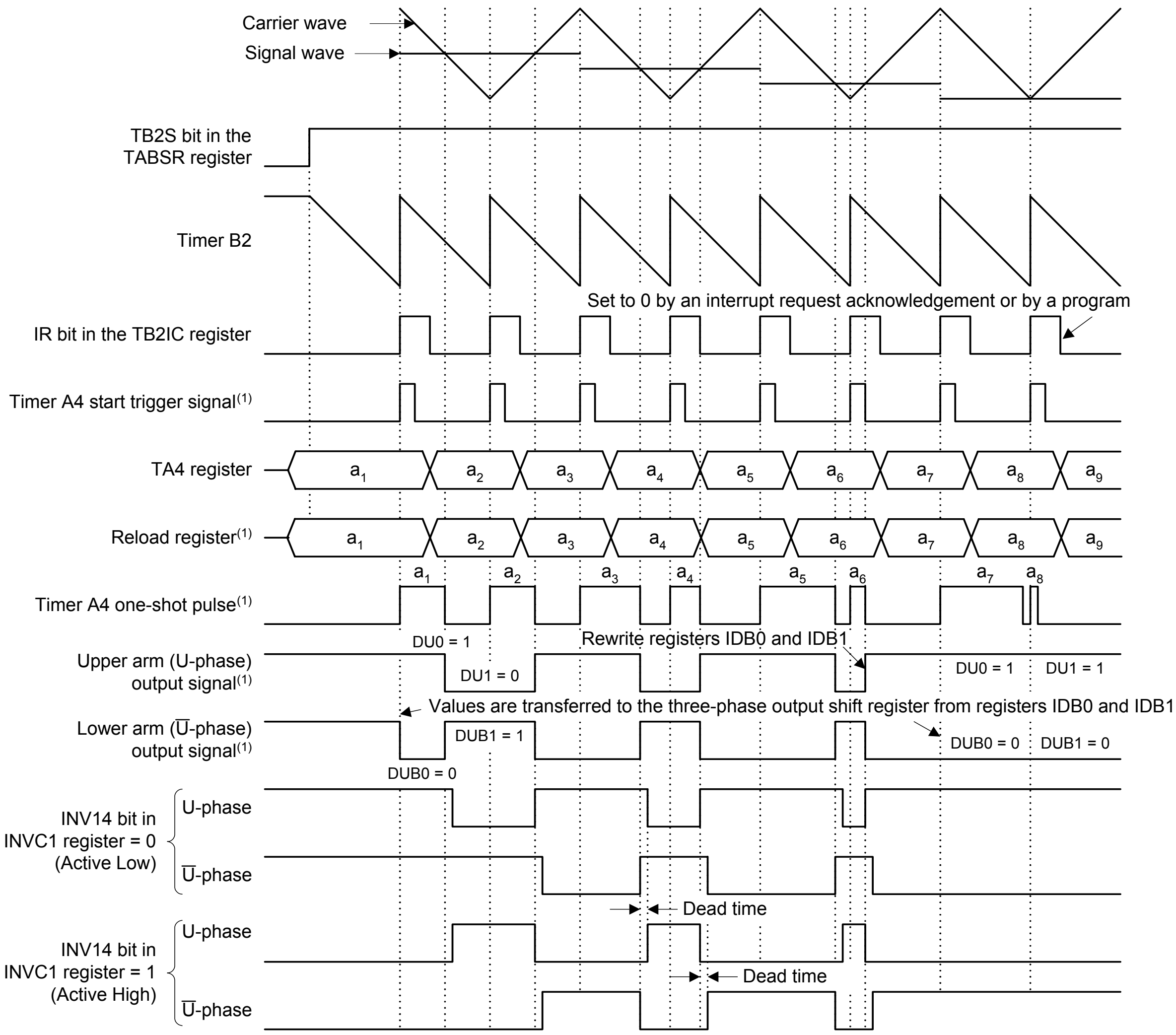
Table 16.3 Specifications and Settings of Triangular Wave Modulation Mode

<table>
<tr><th>Item</th><th>Three-Phase Mode 0</th><th colspan="3">Three-Phase Mode 1</th></tr>
<tr><td>INV06 bit</td><td>0</td><td colspan="3">0</td></tr>
<tr><td>INV11 bit</td><td>0</td><td colspan="3">1</td></tr>
<tr><td>Bits INV01 and INV00</td><td>00b or 01b</td><td>00b</td><td>10b</td><td>11b</td></tr>
<tr><td>PWCON bit</td><td>0</td><td colspan="3">0 or 1</td></tr>
<tr><td>ICTB2 register</td><td>1</td><td colspan="3">n</td></tr>
<tr><td>Carrier wave cycle</td><td>$\frac{2}{f1} \times (m + 1)$</td><td colspan="3">$\frac{2}{f1} \times (m+1)$</td></tr>
<tr><td>Upper arm active level output width</td><td>$\frac{1}{f1} \times (m+1 - a_{2k-1}+a_{2k})$</td><td colspan="3">$\frac{1}{f1} \times (m+1 - b_k+a_k)$</td></tr>
<tr><td>INV13 bit</td><td>0 or 1</td><td colspan="3">Indicates the timer A1 reload control signal state.</td></tr>
<tr><td rowspan="2">Timer B2 interrupt generation timing</td><td rowspan="2">Timer B2 underflow</td><td rowspan="2">Every n-th timer B2 underflow</td><td colspan="2">Every 2n-th timer B2 underflow</td></tr>
<tr><td>Every odd-numbered (2n × j - 1) timer B2 underflow</td><td>Every even-numbered (2n × j) timer B2 underflow</td></tr>
<tr><td>Timer B2 reload timing</td><td>Timer B2 underflow</td><td colspan="3">• Timer B2 underflow (PWCON = 0)
• Timer B2 underflow at the rising edge of the timer A1 reload control signal (PWCON = 1)</td></tr>
<tr><td>Transfer timing from IDBp register to three-phase output shift register</td><td colspan="4">When a value is written to the IDBp register (p = 0, 1), the value is transferred only once by the first transfer trigger.</td></tr>
<tr><td>Dead time timer start timing</td><td colspan="4">• 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)</td></tr>
</table>

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)



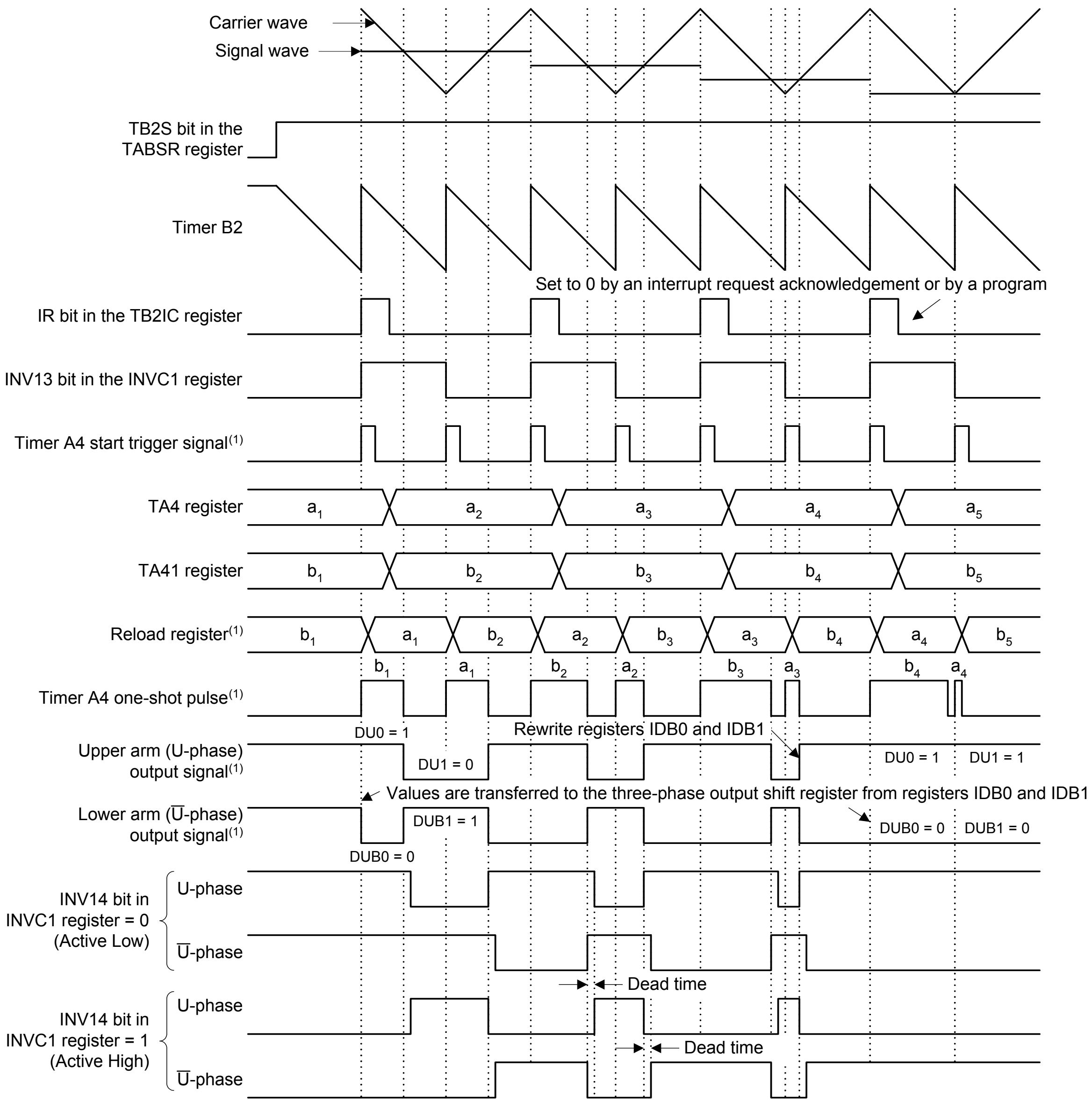
NOTE:
1. Internal signals. See **Three-Phase Motor Control Timer Function Block Diagram.**

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)



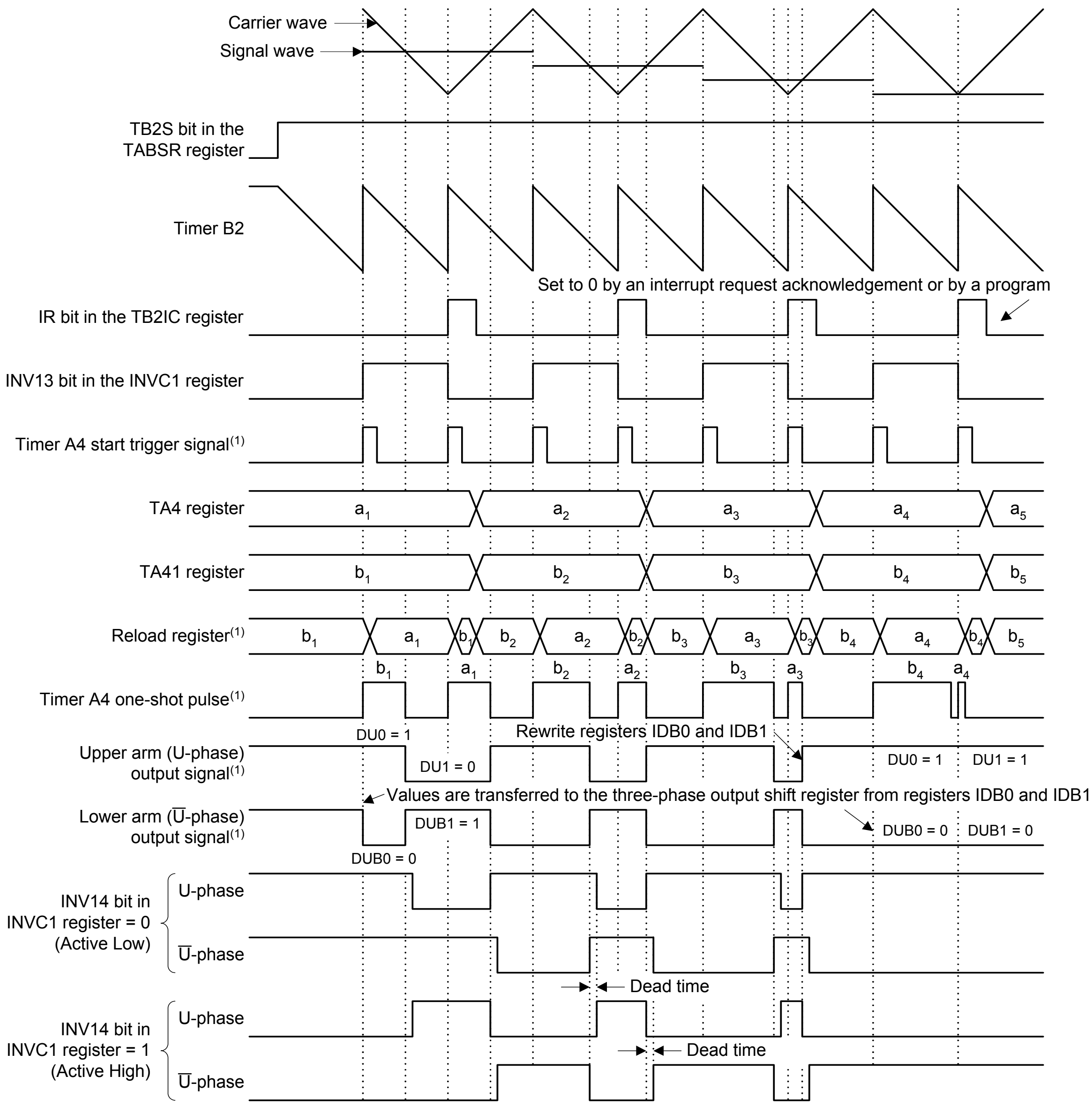
NOTE:
1. Internal signals. See **Three-Phase Motor Control Timer Function Block Diagram.**

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, T21, 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)



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.

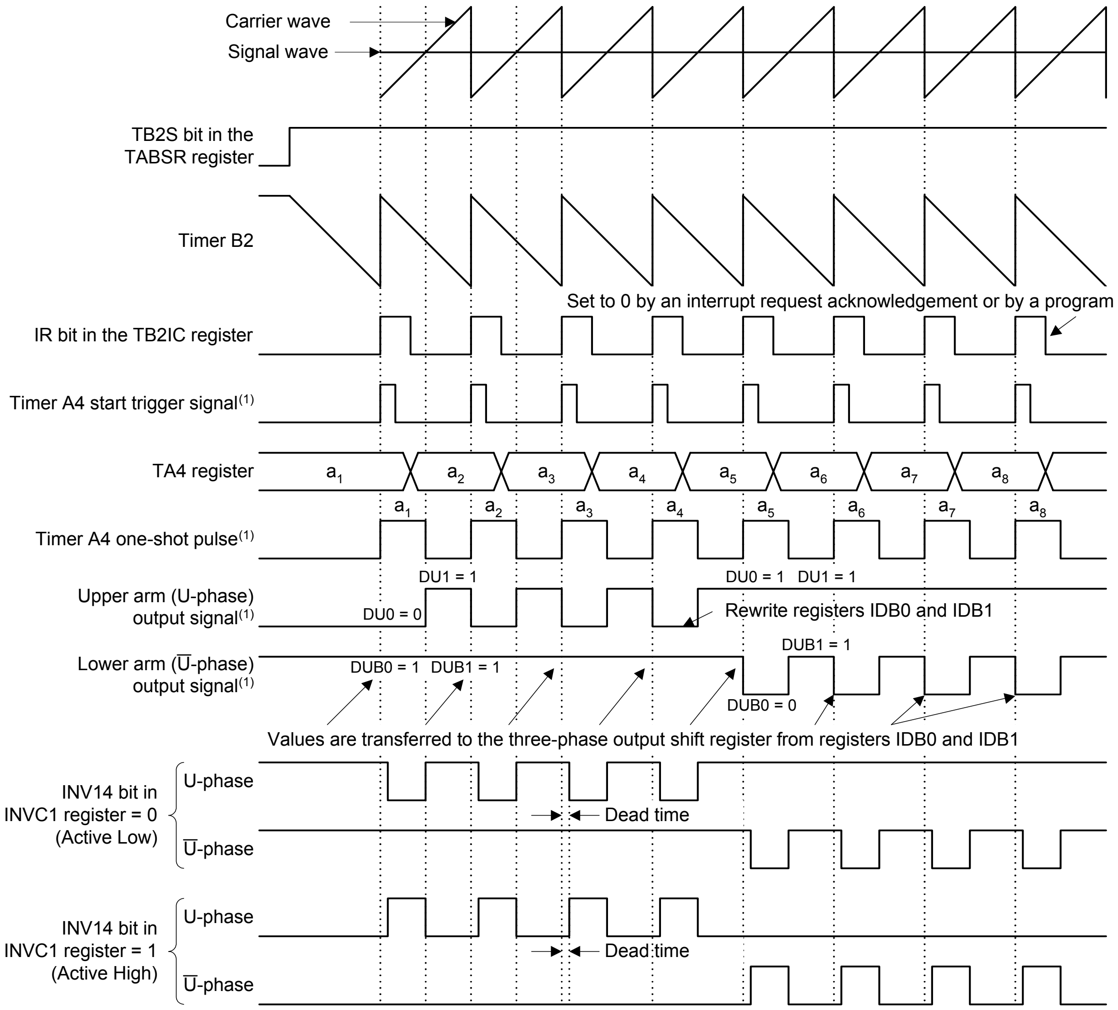
Table 16.4 Specifications and Settings of Sawtooth Wave Modulation Mode

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



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.

$$\frac{1}{f1} \times n \quad (\text{INV12} = 0)$$

$$\frac{2}{f1} \times n \quad (\text{INV12} = 1) \qquad \text{n: Value in the DTT register}$$

Figure 16.15 shows an example of dead time timer operation.

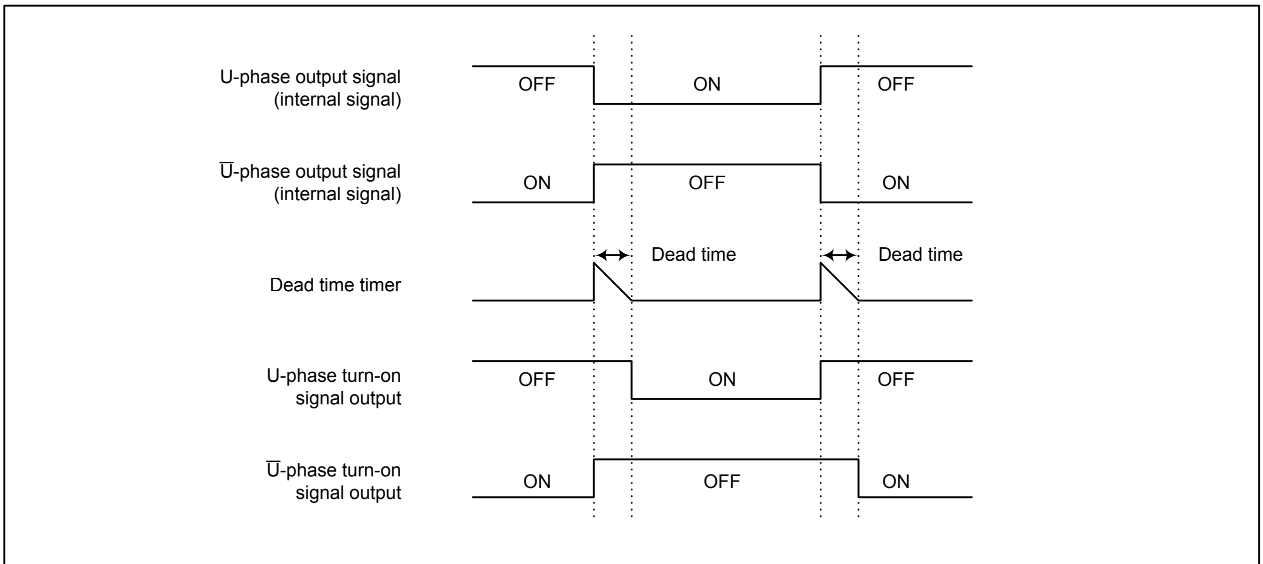


Figure 16.15 Dead Time Timer Operation

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

NOTE

The 144-pin package is described as an example in this chapter.
UART6 is not provided in the 100-pin package.

Serial interfaces consist of seven channels (UART0 to UART6).
Each UARTi (i = 0 to 6) has an exclusive timer to generate the serial clock and operates independently of each other.
Table 17.1 lists a UART0 to UART6 function comparison.

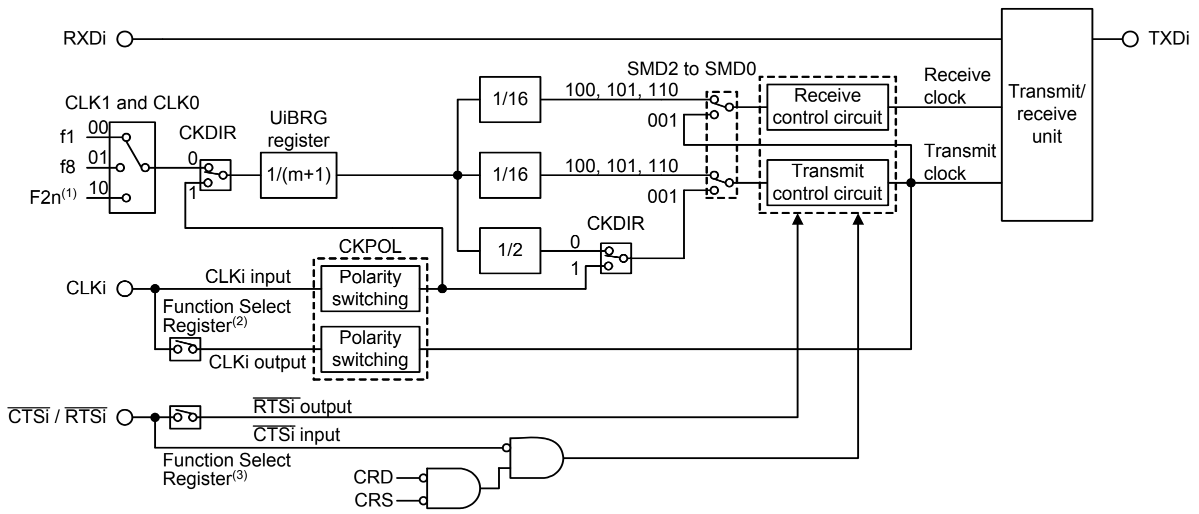
Table 17.1 UART0 to UART6 Function Comparison

Mode	UART0	UART1 to UART4	UART5, UART6
Clock synchronous mode	Provided	Provided	Provided
Clock asynchronous mode (UART mode)	Provided	Provided	Provided
Special mode 1 (I^2C mode)	Provided	Provided	Not provided
Special mode 2	Provided	Provided	Not provided
Special mode 3 (clock-divided synchronous function, GCI mode)	Provided	Provided	Not provided
Special mode 4 (SIM mode)	Provided	Provided	Not provided
Special mode 5 (IrDA mode)	Provided	Not provided	Not provided
Special mode 6 (bus conflict detect function, IE mode) (optional)(1)	Provided	Provided	Not provided

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

17.1 UART0 to UART4

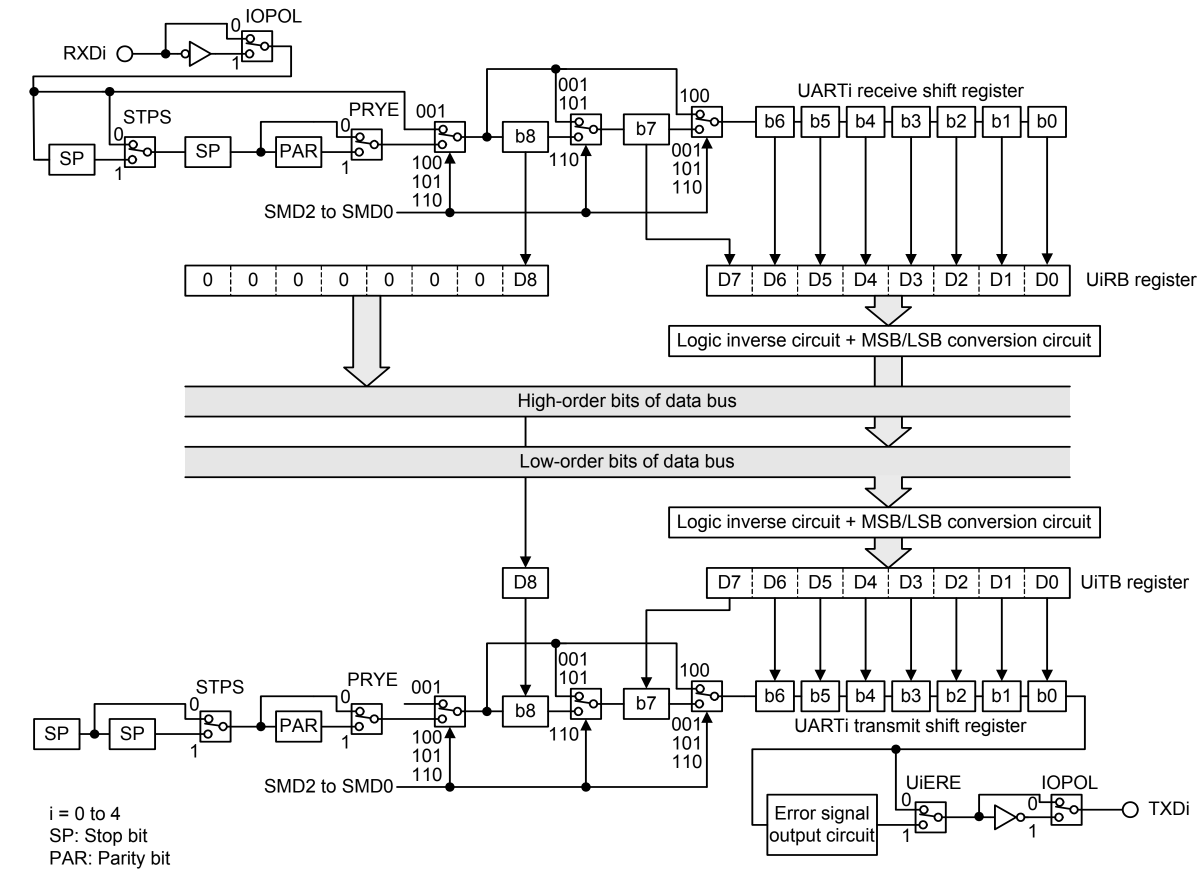
Figure 17.1 shows a UART0 to UART4 block diagram. Figures 17.2 to 17.10 show the registers associated with UART0 to UART4. Refer to the tables listing for register and pin settings in each mode.



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)

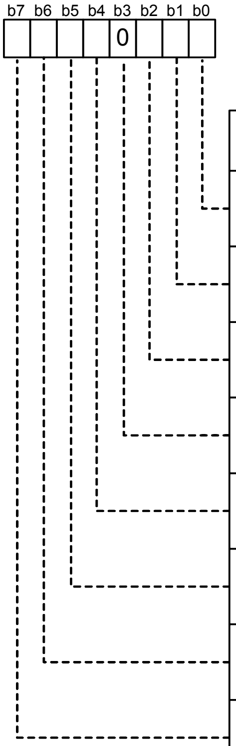
b7 b6 b5 b4 b3 b2 b1 b0

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^2C mode select bit	0 : Other than I^2C mode 1 : I^2C 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^2C mode.
2. The BBS bit is set to 0 by writing a 0. 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)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
U0SMR2 to U2SMR2	0366h, 02E6h, 0336h	00h
U3SMR2, U4SMR2	0326h, 02F6h	00h

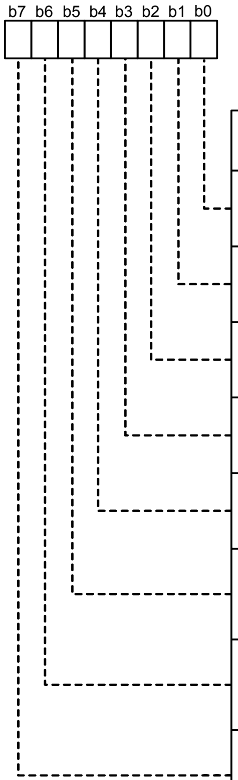
Bit Symbol	Bit Name	Function	RW
IICM2	I^{2}C mode select bit 2	0: ACK/NACK interrupt used 1: Transmit/receive interrupt used	RW
CSC	Clock synchronous bit[1]	0: Not clock synchronized 1: Clock synchronized	RW
SWC	SCL wait output bit[2]	0: No wait state/release wait states 1:SCLi pin is held "L" after 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^{2}C mode.
2. These bits are used when the MCU is in slave mode in I^{2}C 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 input 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 $\overline{SS}$ pin is set to 1, set the CRD bit in the UiC0 register to 1 (CTS function disabled).
3. The ERR bit is 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)

b7 b6 b5 b4 b3 b2 b1 b0

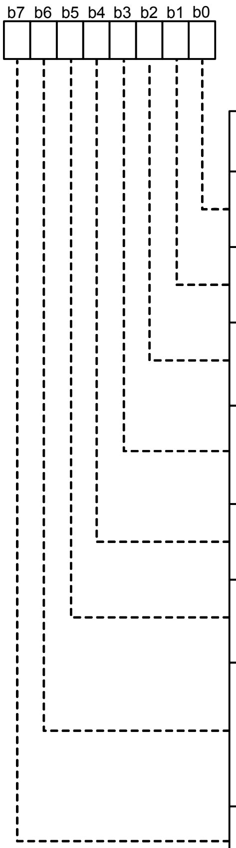
Symbol	Address	After Reset
U0SMR4 to U2SMR4	0364h, 02E4h, 0334h	00h
U3SMR4, U4SMR4	0324h, 02F4h	00h

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)	0: Serial input/output circuit selected 1: Start/stop condition generation circuit selected(4)	RW
ACKD	ACK data bit(2)	0: ACK 1: NACK	RW
ACKC	ACK data output enable bit(2)	0: Serial data output 1: ACK data output	RW
SCLHI	SCL output stop bit(1)	When the bus is free, 0: SCLi output not stopped 1: SCLi output stopped	RW
SWC9	SCL wait output bit 3(1)	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^2C mode.
2. These bits are used when the MCU is in slave mode in I^2C 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).

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)

b7 b0

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)

b7 b6 b5 b4 b3 b2 b1 b0 (b5 = 0)

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	RW

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^2C 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

b7 b6 b5 b4 b3 b2 b1 b0

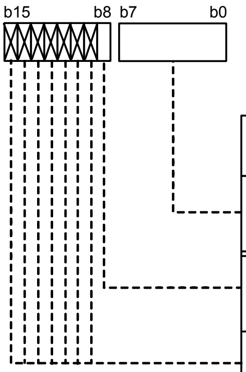
Symbol	Address	After Reset
IFSR	031Fh	00h

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)

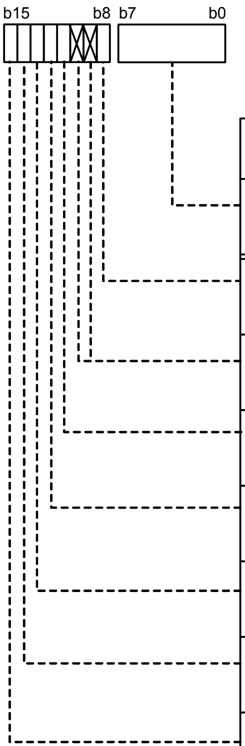


Symbol	Address	After Reset
U0TB to U2TB	036Bh - 036Ah, 02EBh - 02EAh, 033Bh - 033Ah	Undefined
U3TB, U4TB	032Bh - 032Ah, 02FBh - 02FAh	Undefined

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 17.2 lists specifications of clock synchronous mode. Table 17.3 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 17.2 Clock Synchronous Mode Specifications

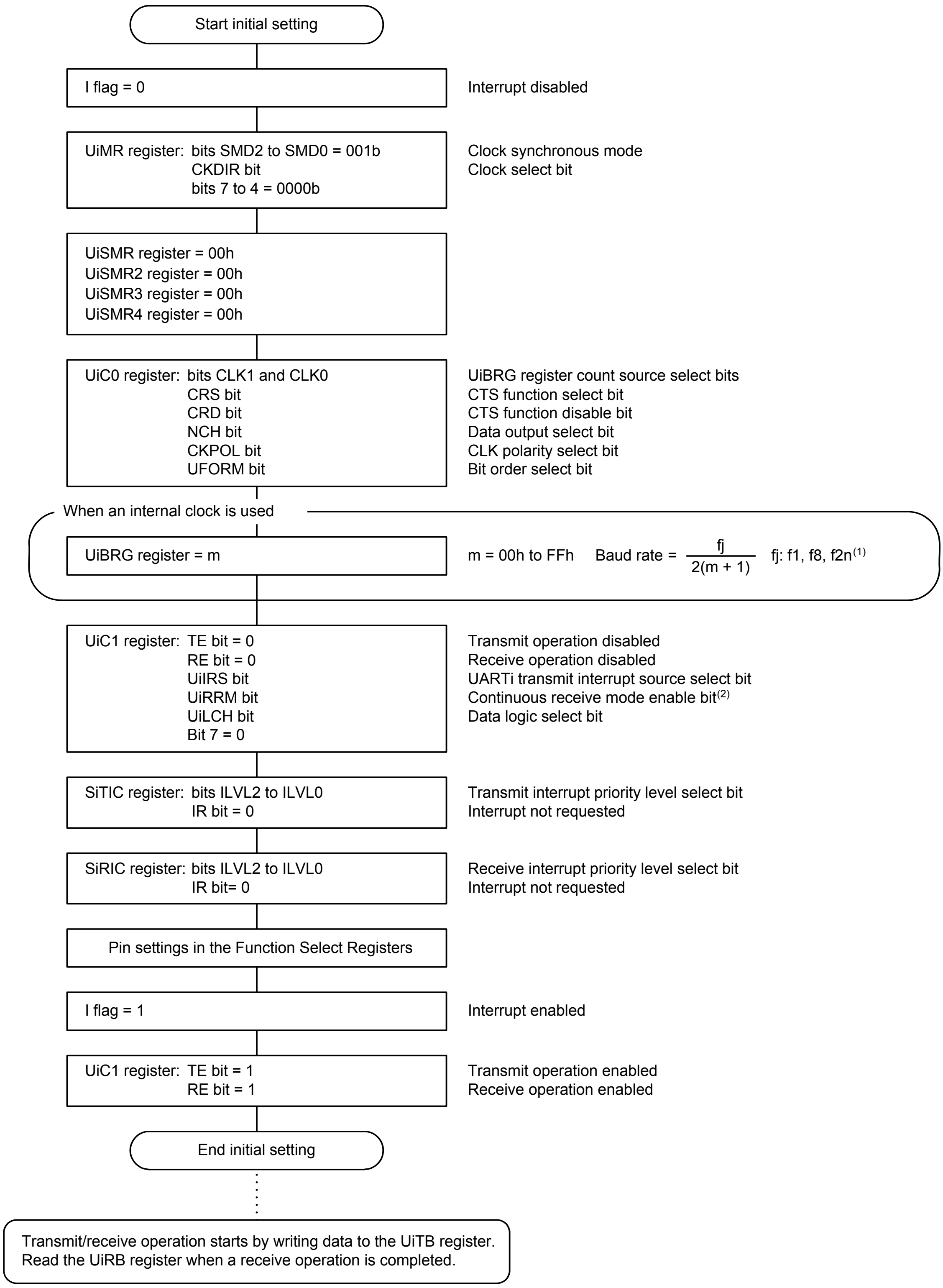
Item	Specification
Data format	Data length: 8 bits long
Serial clock	Internal clock or external clock can be selected by the CKDIR bit in the UiMR register (i = 0 to 4)
Baud rate	• When the CKDIR bit is set to 0 (internal clock): fj / (2 (m + 1) fj = f1, f8, f2n(1) m: setting value of the UiBRG register (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 $\overline{CTSi}$ 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, $\overline{RTSi}$ 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 17.3 Pin Settings in Clock Synchronous Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC, PSC3 Registers	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)	–	PSC3_6 = 0	–	PS3_6 = 1
P9_7	RXD4 input	PD9_7 = 0	–	–	PS3_7 = 0

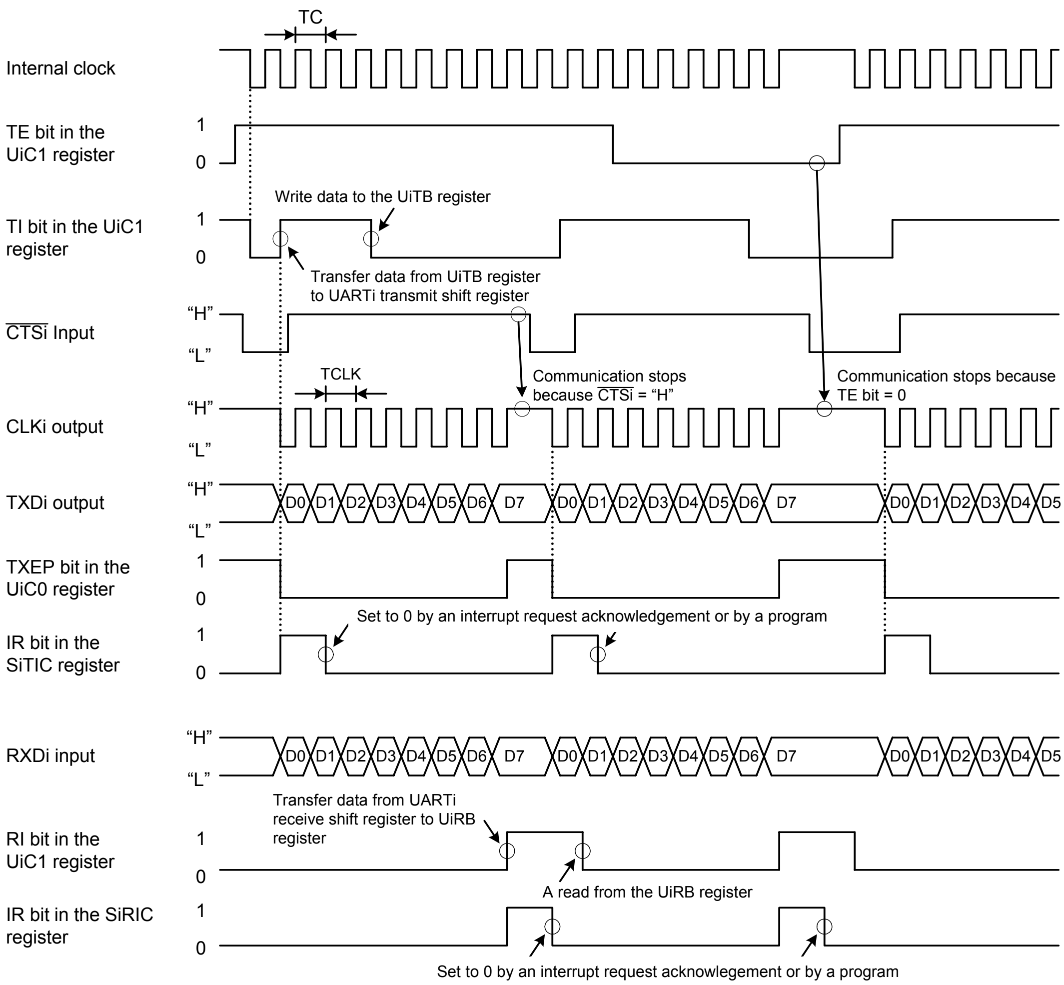
NOTES:
1. Set registers PS0, PS1, and PS3 after setting the other registers.
2. Set the PD9 or PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.
3. P7_0 is an N-channel 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).



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



i = 0 to 4

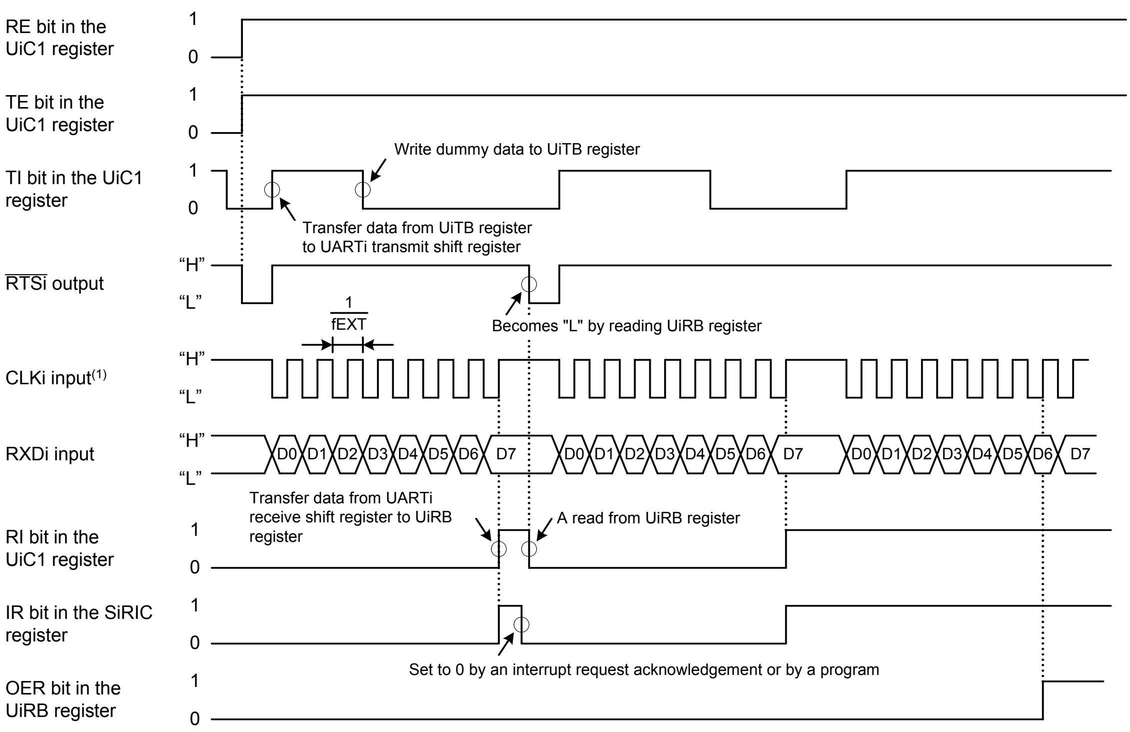
$$TC = TCLK = \frac{2(m + 1)}{fj}$$

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



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)

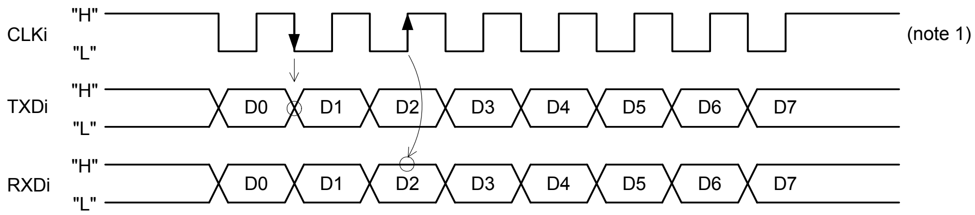
NOTE:
1. Satisfy the following conditions, while the CLKi pin input is "H" before the data receive operation.
   - UiC1 register: TE bit = 1 (transmit operation enabled)
     RE bit = 1 (receive operation enabled)
   - Write dummy data to the UiTB register

Figure 17.13 Receive Operations when External Clock is Selected

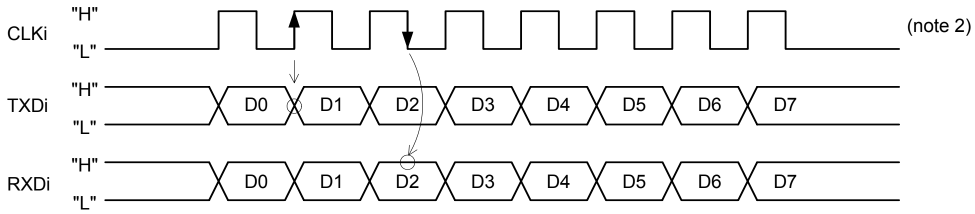
17.1.1.1 CLK Polarity

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.

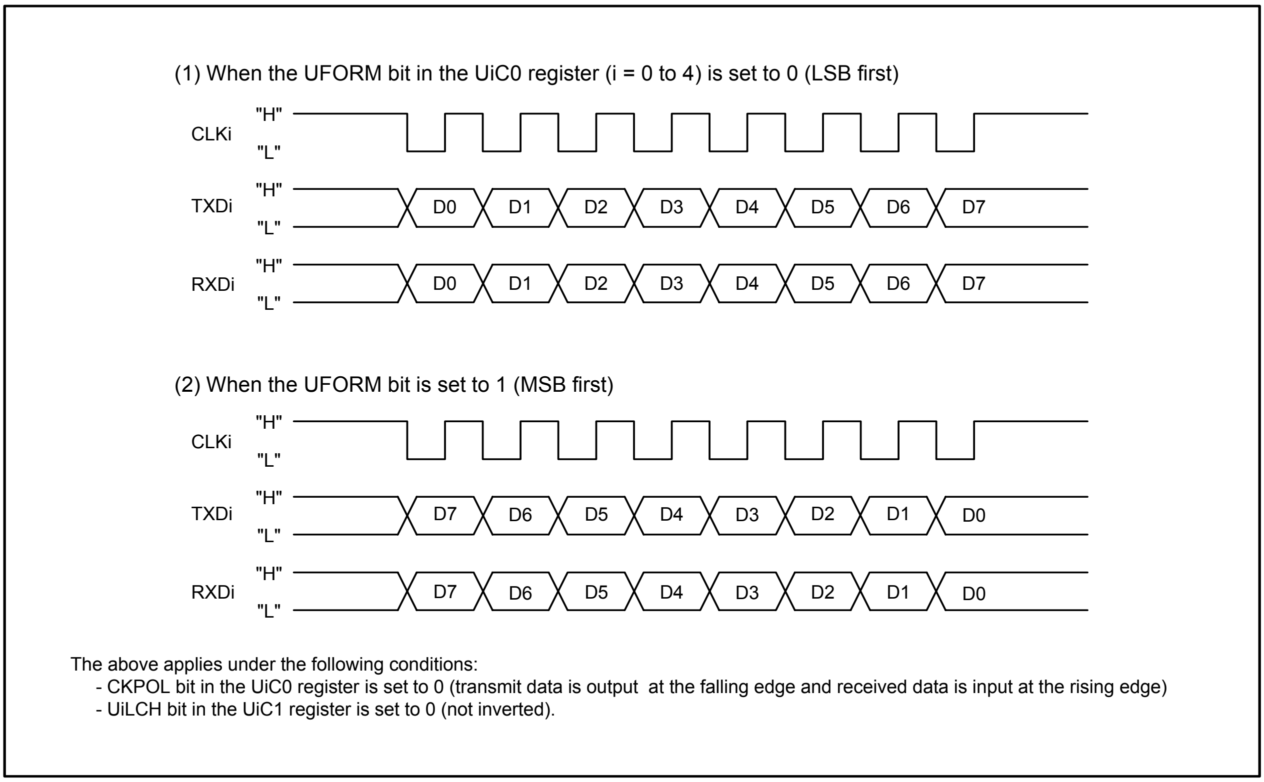


Figure 17.15 Bit Order (8-Bit Data Length)

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.

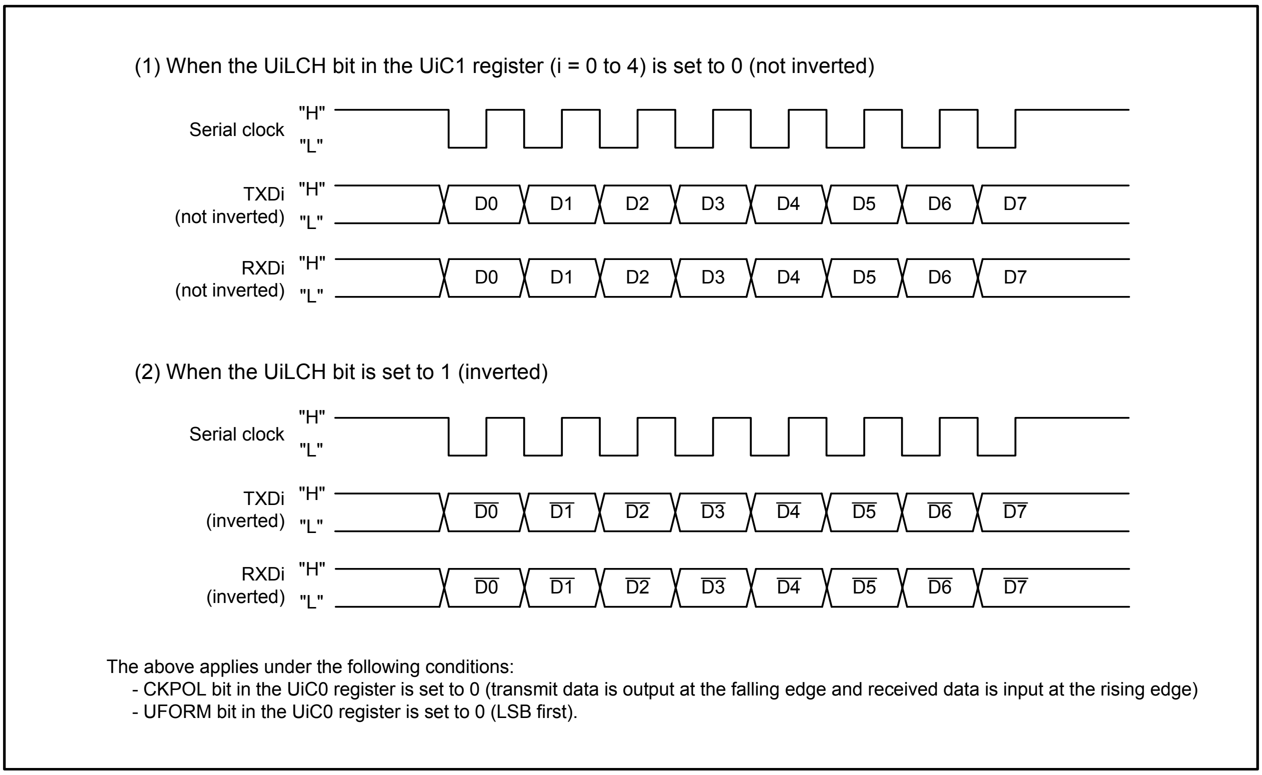


Figure 17.16 Serial Data Logic Inverse

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 the Communication Error is Occurred

Follow the procedure below when 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 17.4 lists specifications of UART mode. Table 17.5 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 17.4 UART Mode Specifications

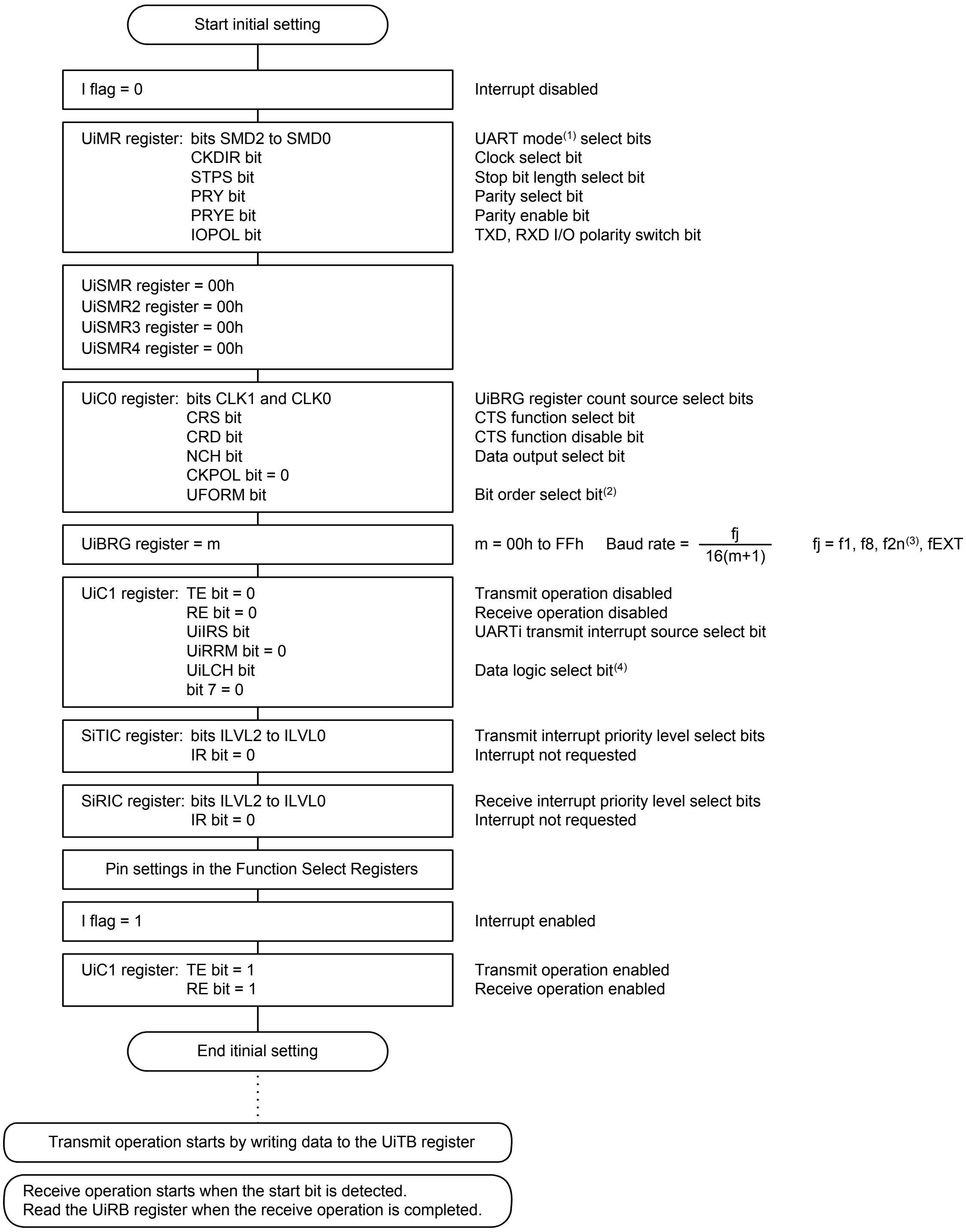
Item	Specification
Data format	• Data length: selectable among 7 bits, 8 bits, or 9 bits long • Start bit: 1 bit long • Parity bit: selectable among odd, even, or none • Stop bit: selectable from 1 bit or 2 bits long
Baud rate	fj / (16 (m + 1)) fj = f1, f8, f2n(1), fEXT m: setting value of the UiBRG 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 17.5 Pin Settings in UART Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers[2]	PSC, PSC3 Registers	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers[1][2]
P6_0	$\overline{CTS0}$ input	PD6_0 = 0	–	–	PS0_0 = 0
	$\overline{RTS0}$ 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	$\overline{CTS1}$ input	PD6_4 = 0	–	–	PS0_4 = 0
	$\overline{RTS1}$ 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	$\overline{CTS2}$ input	PD7_3 = 0	–	–	PS1_3 = 0
	$\overline{RTS2}$ 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	$\overline{CTS3}$ input	PD9_3 = 0	–	PSL3_3 = 0	PS3_3 = 0
	$\overline{RTS3}$ output	–	–	–	PS3_3 = 1
P9_4	$\overline{CTS4}$ input	PD9_4 = 0	–	PSL3_4 = 0	PS3_4 = 0
	$\overline{RTS4}$ output	–	–	–	PS3_4 = 1
P9_5	CLK4 input	PD9_5 = 0	–	PSL3_5 = 0	PS3_5 = 0
P9_6	TXD4 output[4]	–	PSC3_6 = 0	–	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).

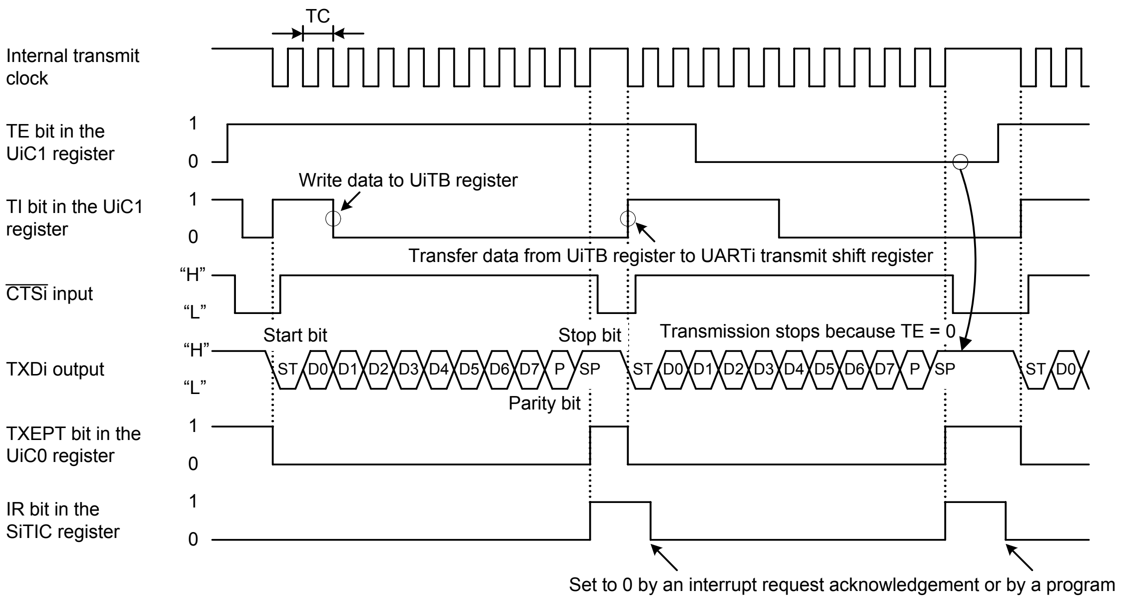


i = 0 to 4
fEXT: clock input to the CLKi pin when the external clock is selected

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.

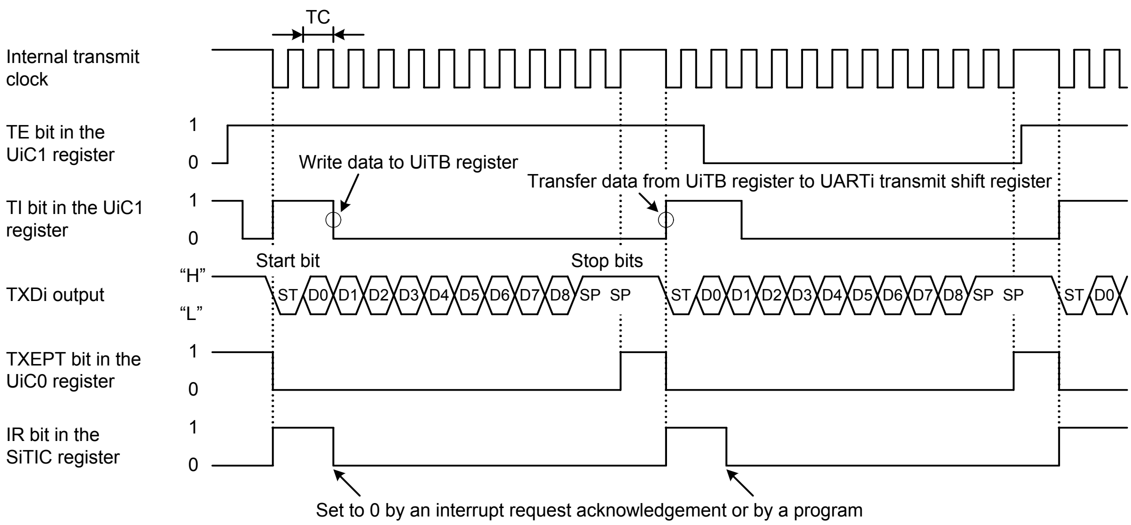
Figure 17.17 Register Settings in UART Mode

(1) Example of the transmit operation timing in 8-bit data length (parity enabled, 1 stop bit)



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)



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)

$$TC = \frac{16(m + 1)}{fj}$$

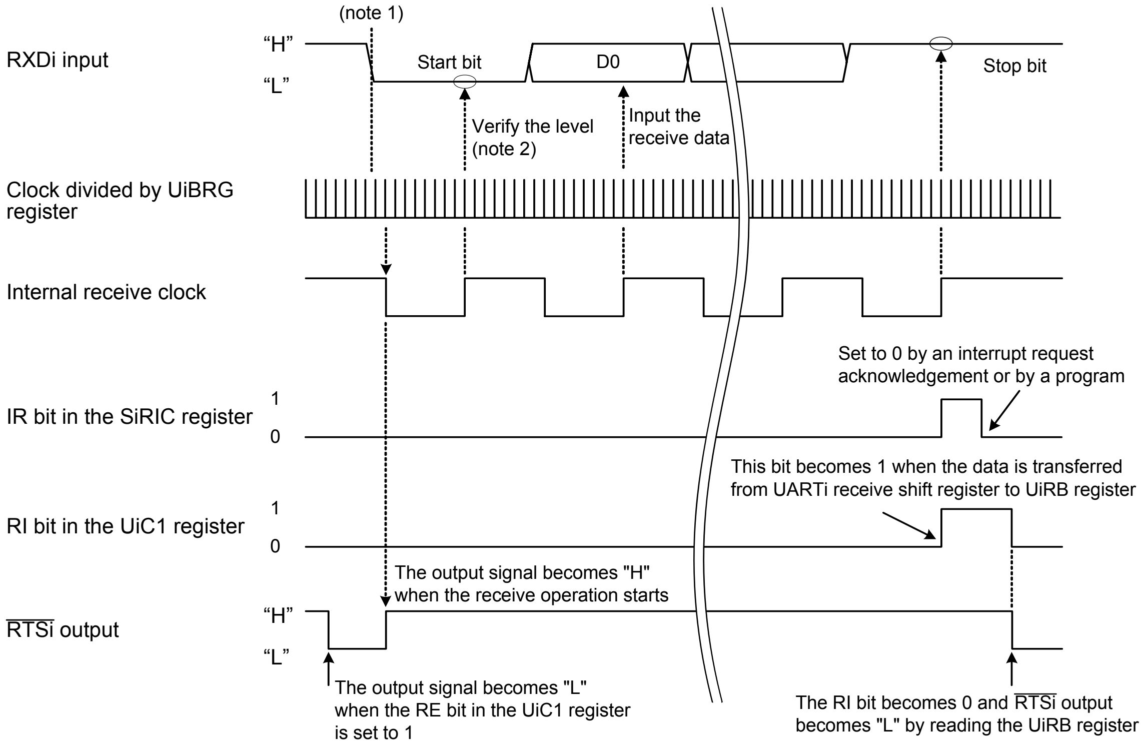
fj: f1, f8, f2n[(1)], fEXT
fEXT: clock input to the CLKi pin when the external clock is selected
m: setting value of the UiBRG register (00h to FFh)

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)



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 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 17.6 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 17.6 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.

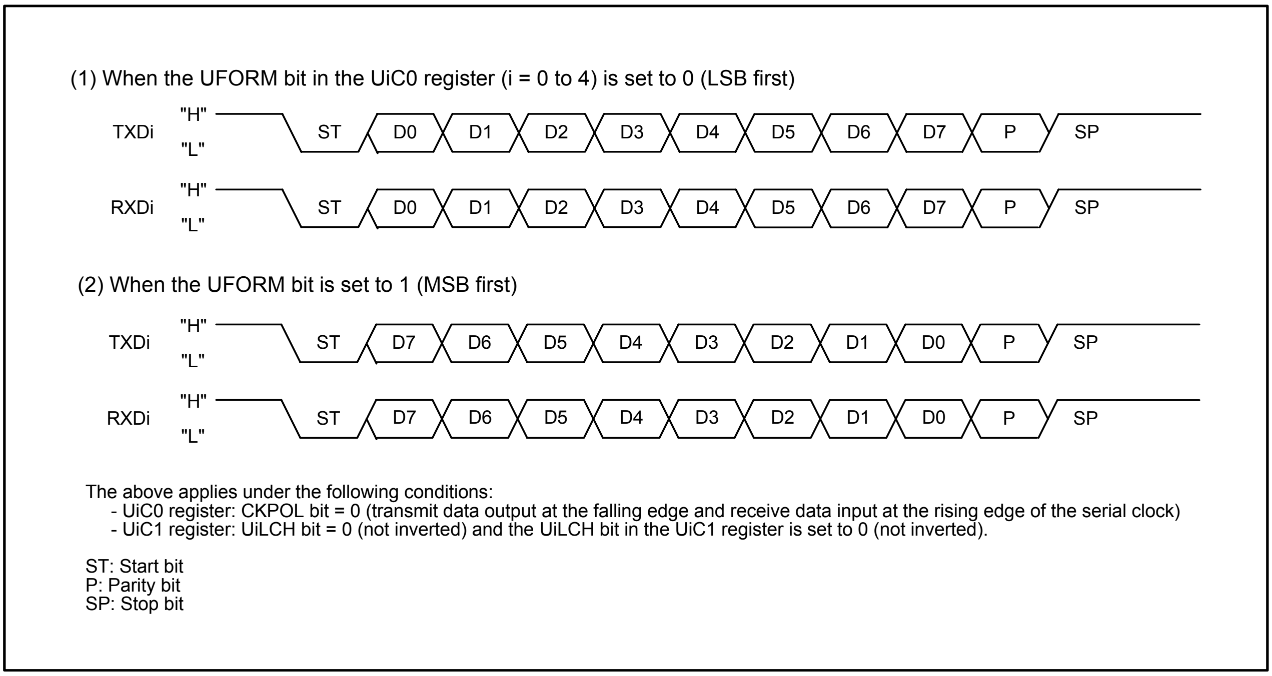


Figure 17.20 Bit Order

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.

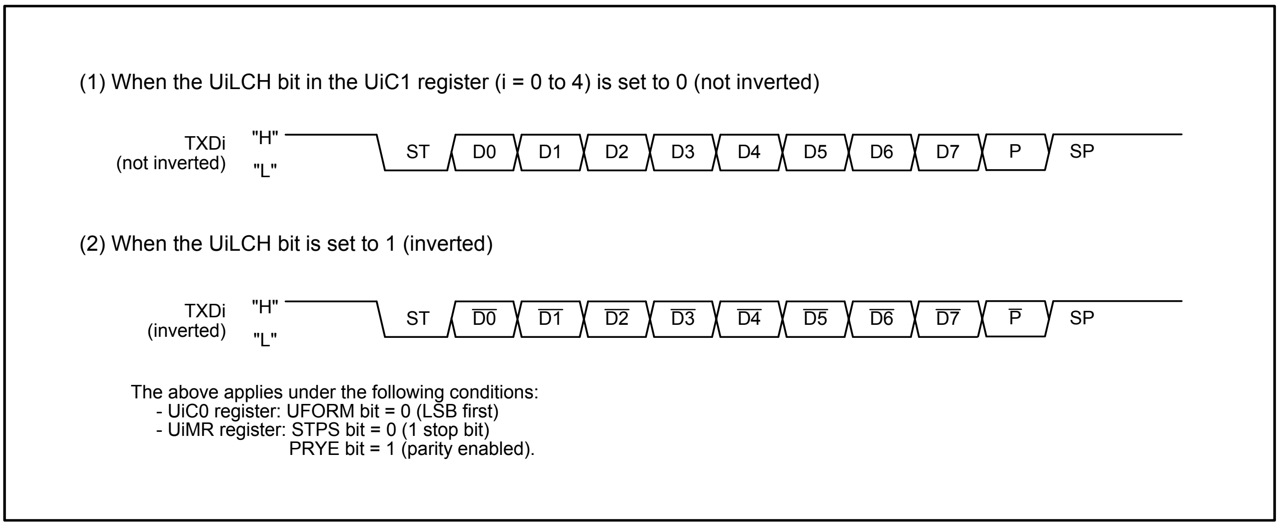


Figure 17.21 Serial Data Logic Inverse

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.

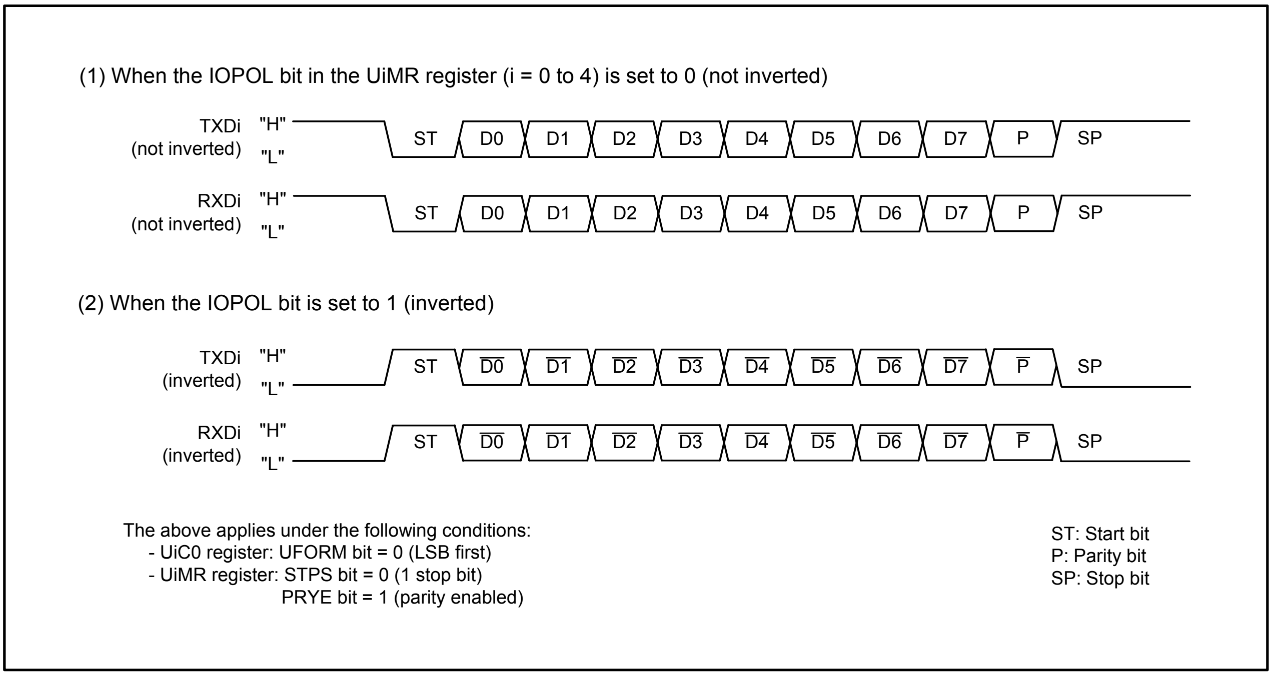


Figure 17.22 TXD and RXD I/O Polarity Inverse

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 the Communication Error is Occurred

Follow the procedure below when 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).

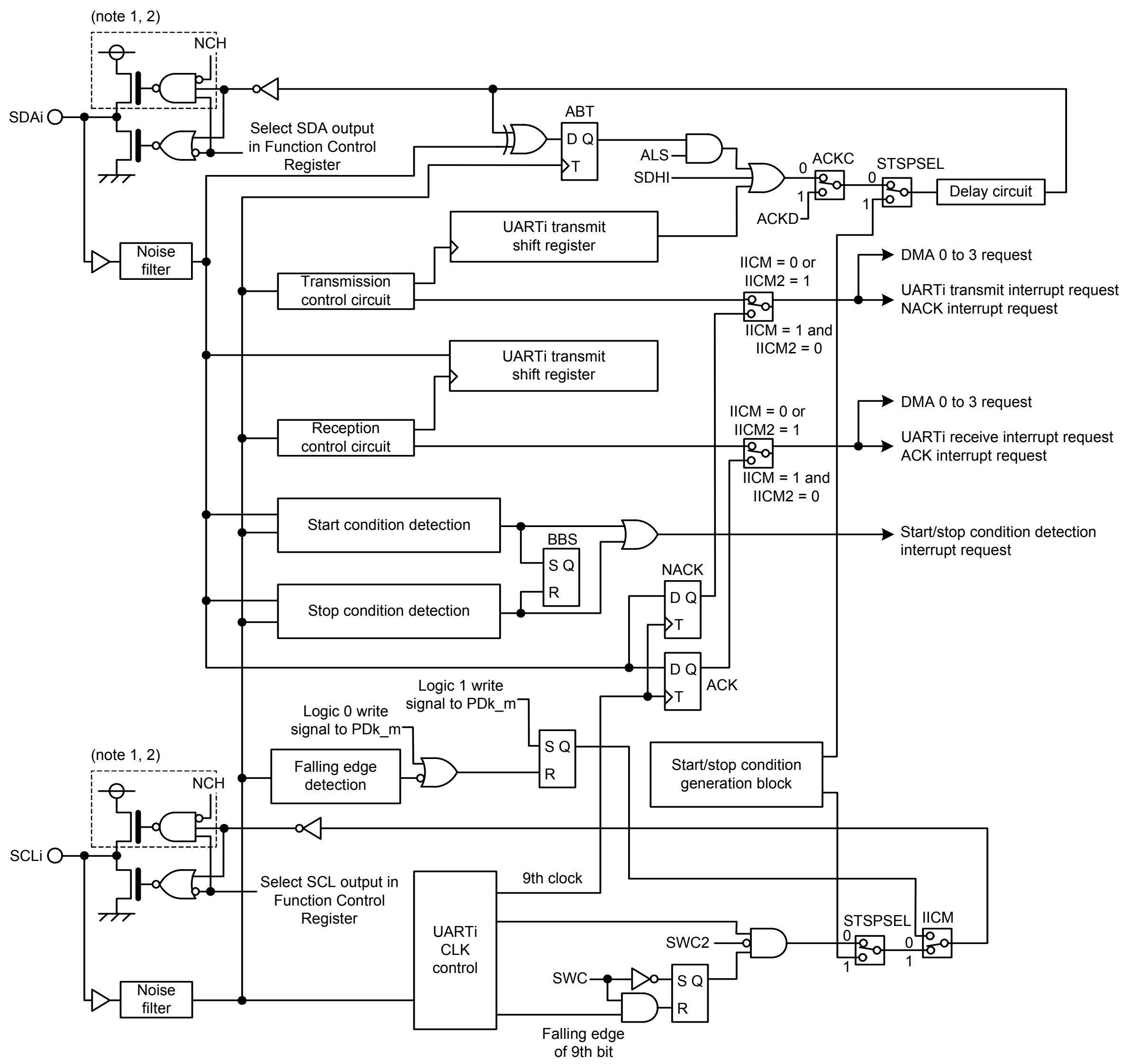
17.1.3 Special Mode 1 (I^2C Mode)

In I^2C mode, the simplified I^2C helps to communicate with external devices.
Table 17.7 lists specifications of I^2C mode. Tables 17.8 and 17.9 list register settings. Tables 17.10 and 17.11 list individual functions in I^2C mode. Table 17.12 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 17.7 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.



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 I2C Mode Block Diagram

Table 17.8 Register Settings in I²C Mode (1/2)

Register	Bit	Setting Value	
		Master	Slave
UiMR	SMD2 to SMD0	Set to 010b	
	CKDIR	Set to 0	Set to 1
	IOPOL	Set to 0	
UiSMR	IICM	Set to 1	
	ABC	Select an arbitration lost detect timing	Disabled
	BBS	Bus busy flag	
	7 to 3	Set to 00000b	
UiSMR2	IICM2	See **Tables 17.10 and 17.11 Functions in I²C Mode**	
	CSC	Set to 1 to enable clock synchronization	Set to 0
	SWC	Set to 1 to hold an “L” signal output from SCLi at the falling edge of the ninth bit of the serial clock	
	ALS	Set to 1 to abort an SDAi output when detecting the arbitration lost	Set to 0
	STC	Set to 0	Set to 1 to initialize UARTi by detecting the start condition
	SWC2	Set to 1 to forcibly make a signal output from SCL an “L”	
	SDHI	Set to 1 to disable SDA output	
	SU1HIM	Set to 0	
UiSMR3	SSE	Set to 0	
	CKPH	See **Tables 17.10 and 17.11 Functions in I²C Mode**	
	DINC, NODC, ERR	Set to 0	
	DL2 to DL0	Set SDAi digital delay value	
UiSMR4	STAREQ	Set to 1 to generate the start condition	Set to 0
	RSTAREQ	Set to 1 to generate the restart condition	
	STPREQ	Set to 1 to generate the stop condition	
	STSPSEL	Set to 1 when using a condition generation function	
	ACKD	Select ACK or NACK	
	ACKC	Set to 1 to output ACK data	
	SCLHI	Set to 1 to enable SCL output stop when detecting the stop condition	Set to 0
	SWC9	Set to 0	Set to 1 to hold an “L” signal output from SCLi at the falling edge of the ninth bit of the serial clock

i = 0 to 4

Table 17.9 Register Settings in I^2C 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 Table 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 17.10 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 17.13 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 17.11 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).

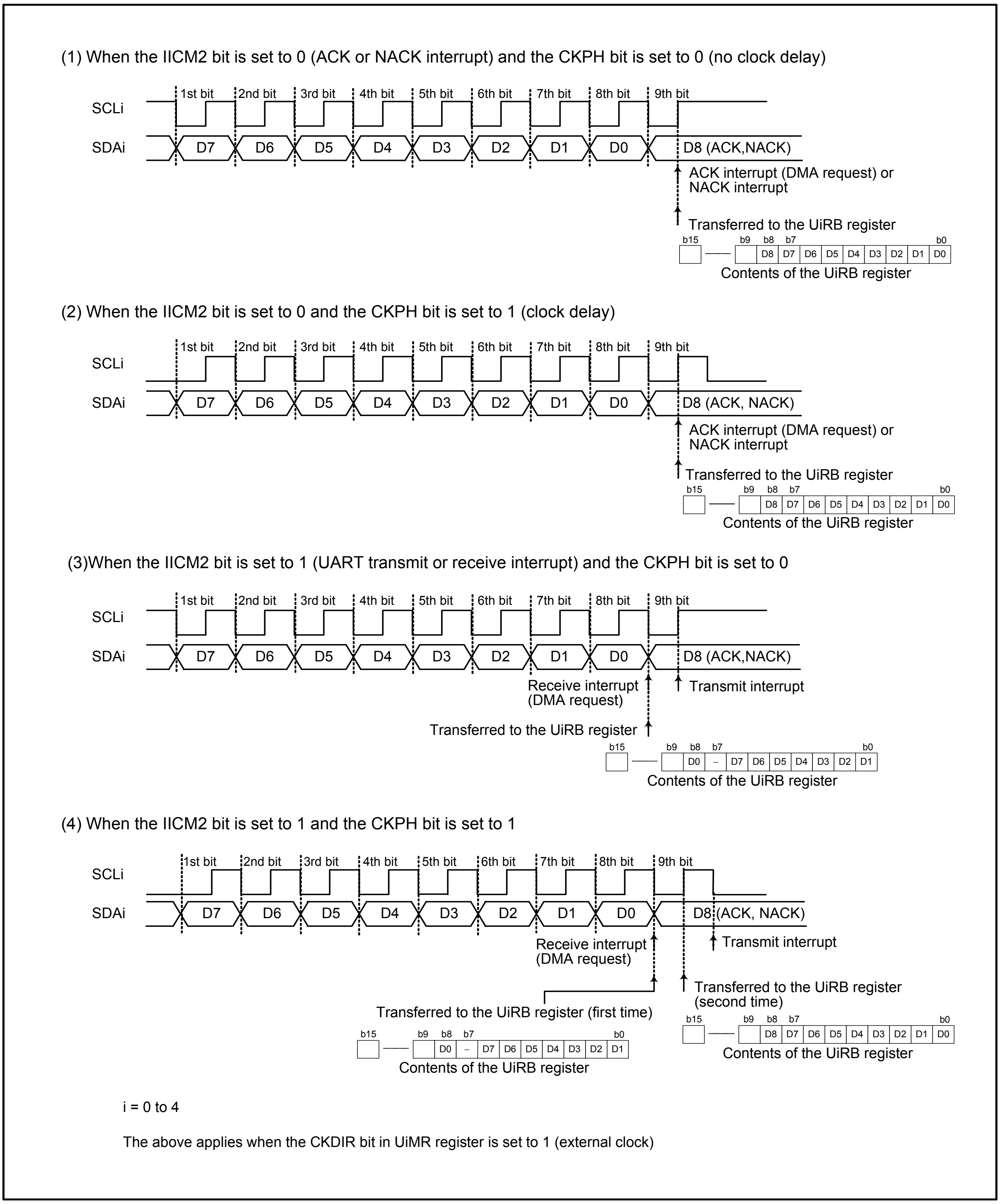


Figure 17.24 Transfer Timing to the UiRB Register and Interrupt Timing

Table 17.12 Pin Settings in I2C Mode

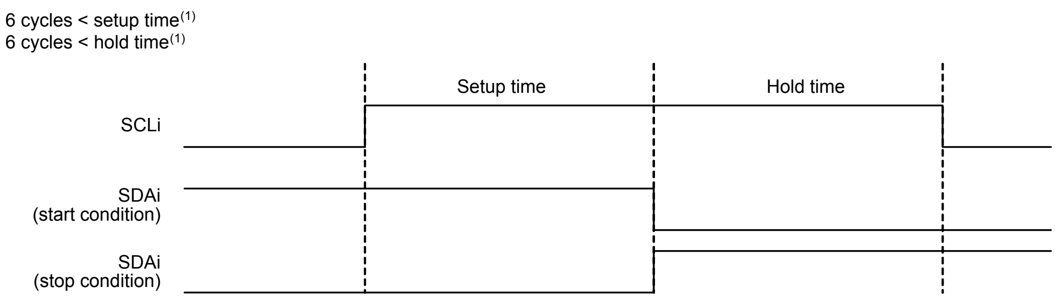
Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC, PSC3 Registers	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers(1)(2)
P6_2	SCL0 output	–	–	PSL0_2 = 0	PS0_2 = 1
	SCL0 input	PD6_2 = 0	–	–	PS0_2 = 0
P6_3	SDA0 output	–	–	PSL0_3 = 0	PS0_3 = 1
	SDA0 input	PD6_3 = 0	–	–	PS0_3 = 0
P6_6	SCL1 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	–	PSC3_6 = 0	–	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 17.13 and Figure 17.26 show functions of the STSPSEL bit.

Table 17.13 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

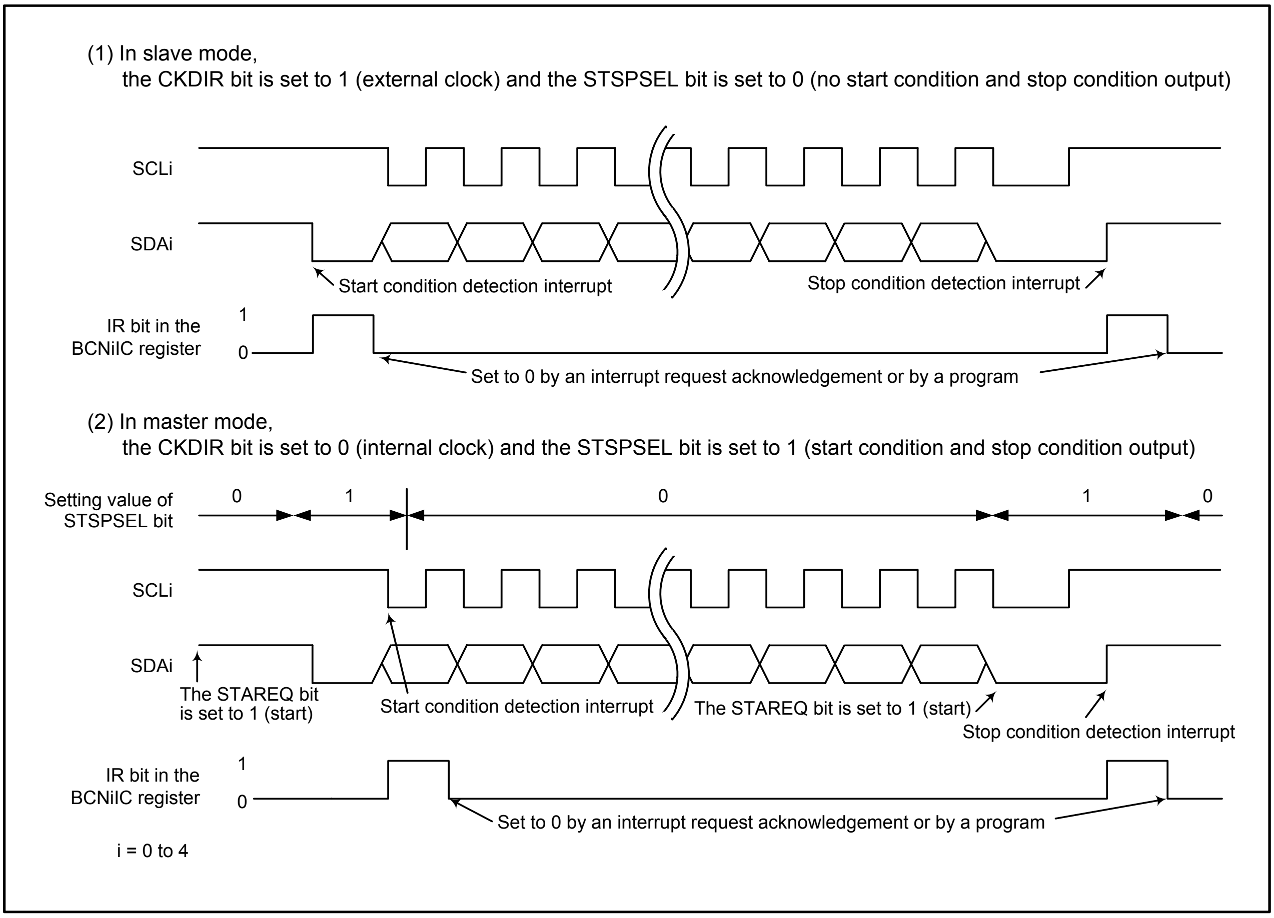


Figure 17.26 STSPSEL Bit Function

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 17.14 lists specifications of special mode 2. Table 17.15 lists pin settings. Figure 17.27 shows register settings.

Table 17.14 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

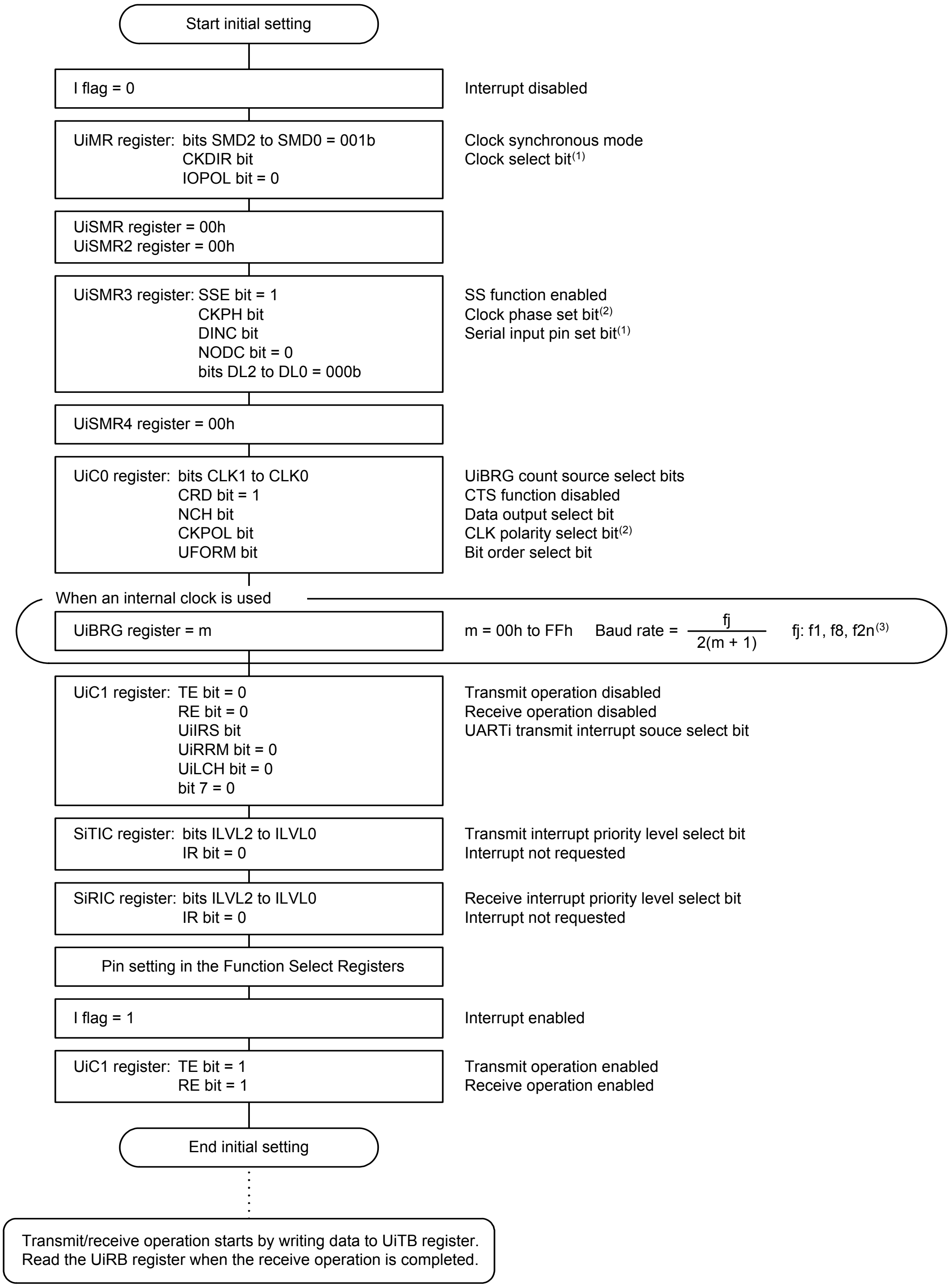
NOTES:
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. If an external clock is selected, ensure that an “H” signal is applied to the CLKi pin when the CKPOL bit in the UiC0 register is set to 0, and that an “L” signal is applied when the CKPOL bit 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 17.15 Pin Settings in Special Mode 2

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC, PSC3 Registers	PSL0, PSL1, PSL3 Registers	PS0, PS1, PS3 Registers(1)(2)
P6_0	$\overline{SS0}$ 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	$\overline{SS1}$ 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	$\overline{SS2}$ 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	$\overline{SS3}$ input	PD9_3 = 0	–	PSL3_3 = 0	PS3_3 = 0
P9_4	$\overline{SS4}$ 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)	–	PSC3_6 = 0	–	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.



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 1. 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 0. 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.

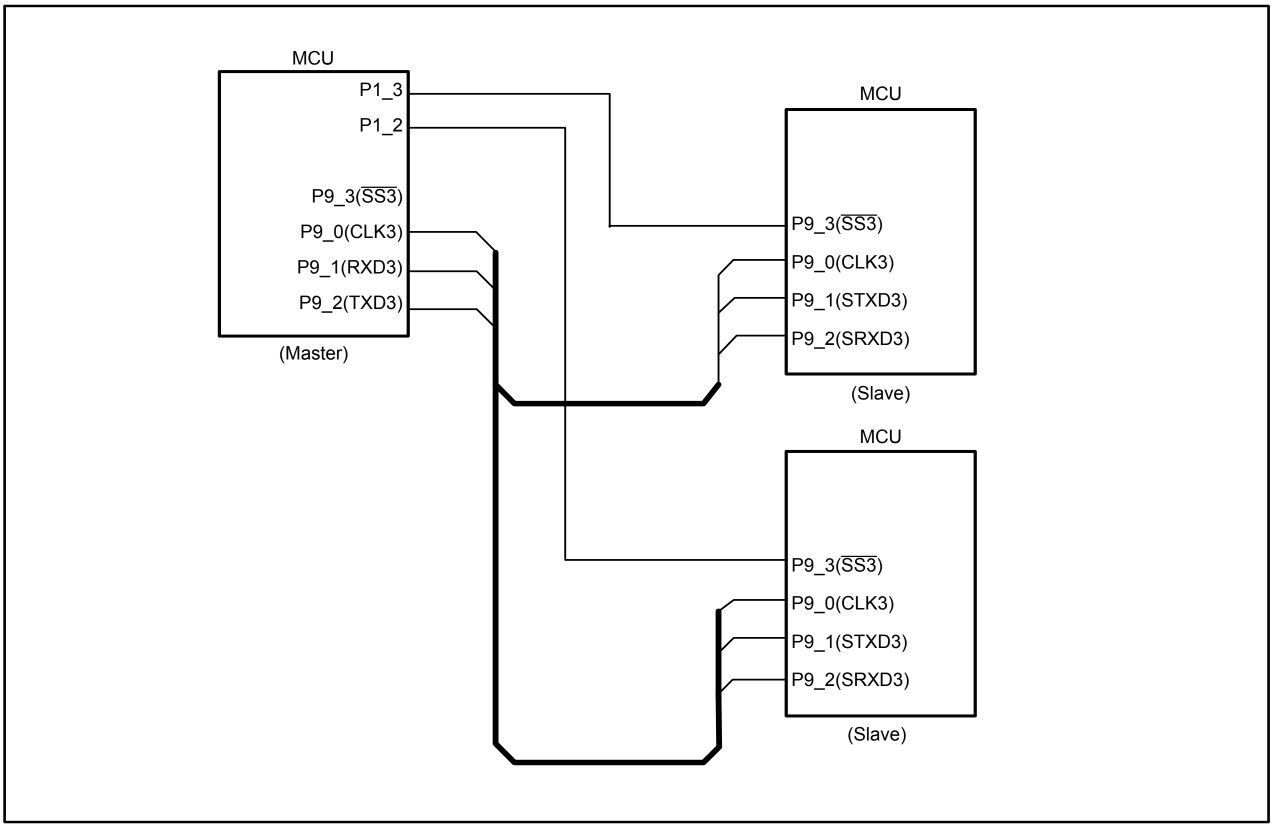


Figure 17.28 Serial Bus Communication Control with $\overline{\text{SSi}}$ Pin

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.

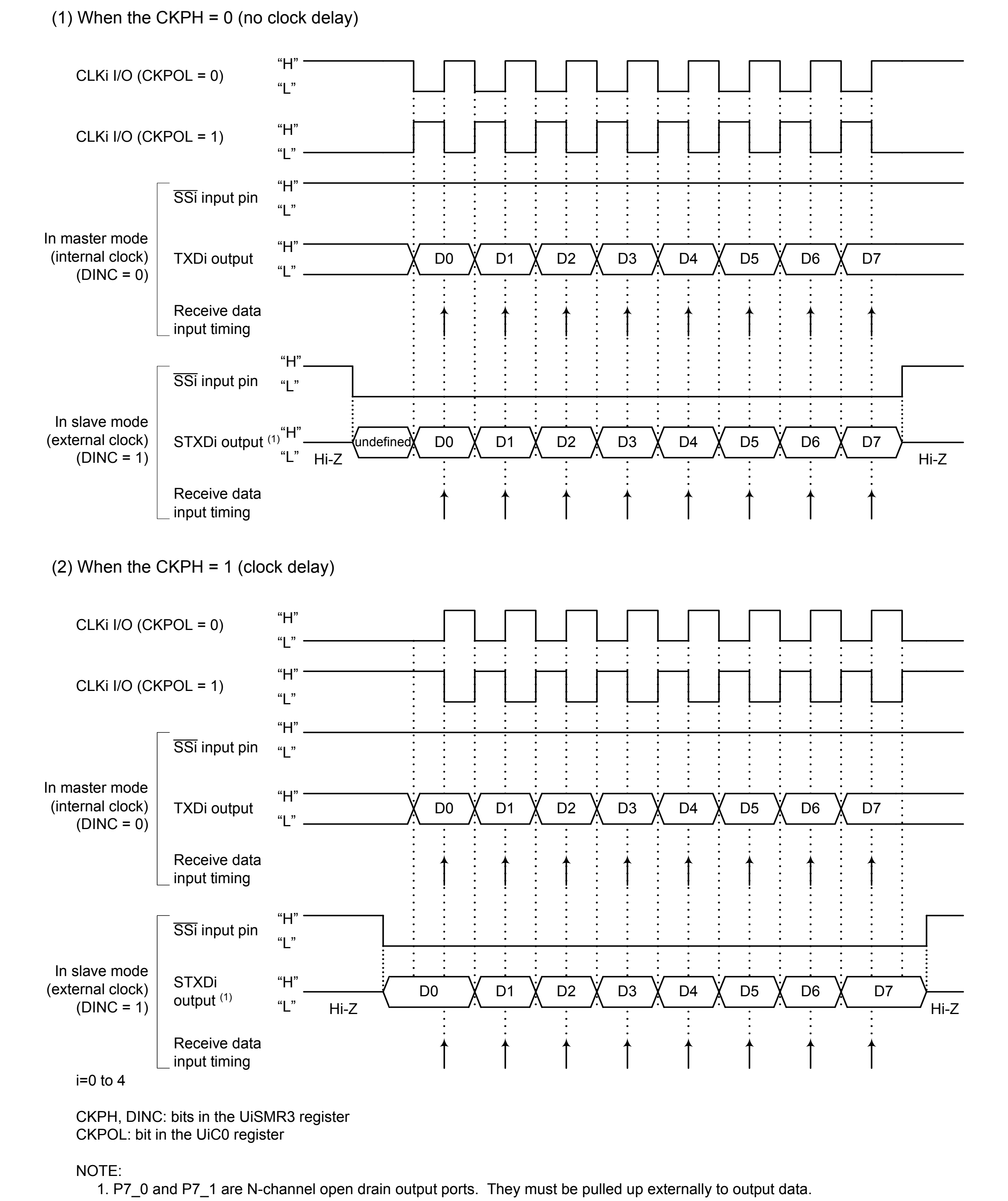


Figure 17.29 Transmit and Receive Operation Timing in Special Mode 2

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 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 17.16 lists specifications of GCI mode. Table 17.17 lists pin settings. Figure 17.30 shows register settings.

Table 17.16 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 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 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

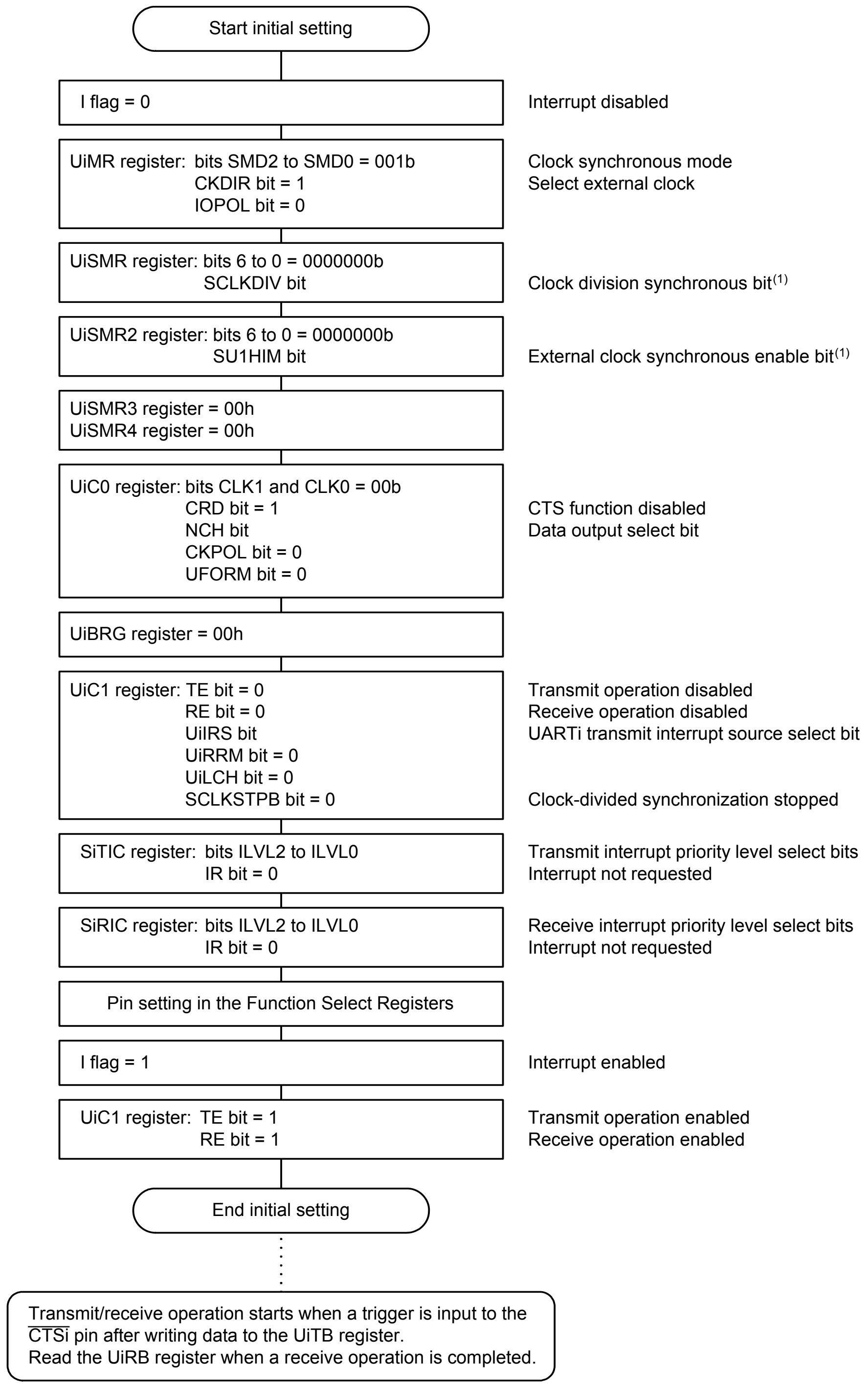
NOTE:

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 17.17 Pin Settings in GCI Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC, PSC3 Registers	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	–	PSC3_6 = 0	–	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.



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 17.18, and apply a trigger signal to the $\overline{CTSi}$ 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 17.18 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

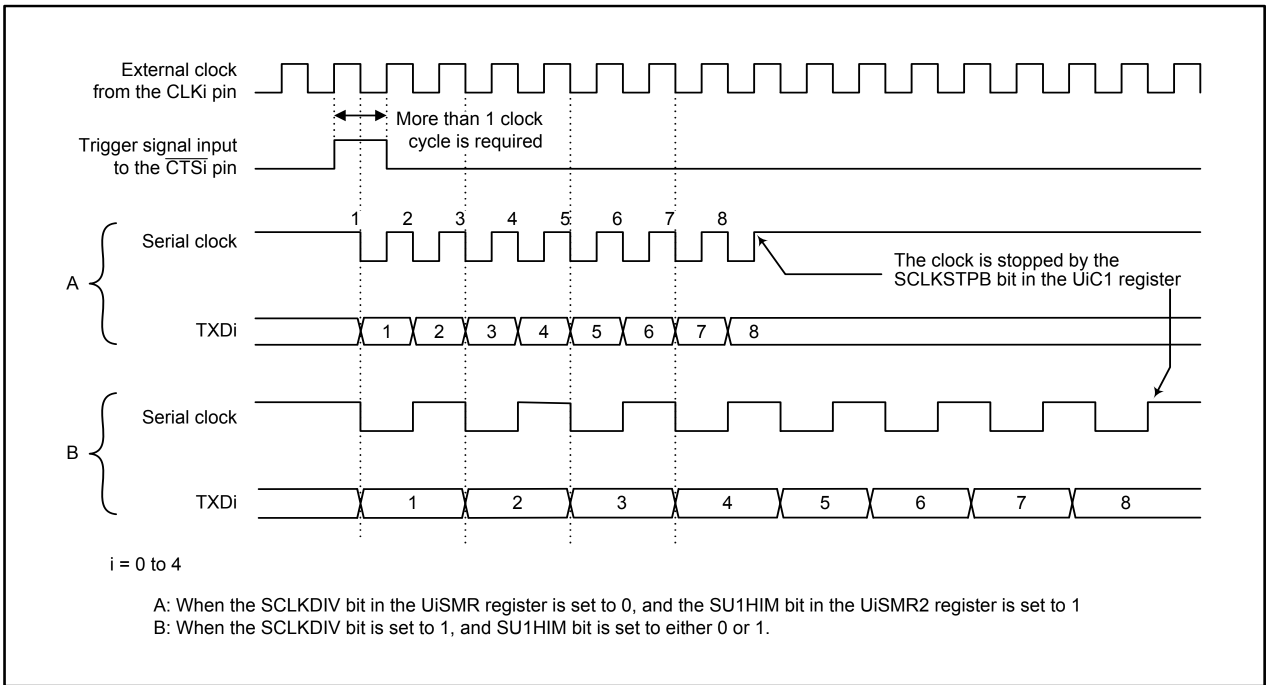


Figure 17.31 Clock-Divided Synchronous Function

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 17.19 lists specifications of SIM mode. Table 17.20 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 17.19 SIM Mode Specifications

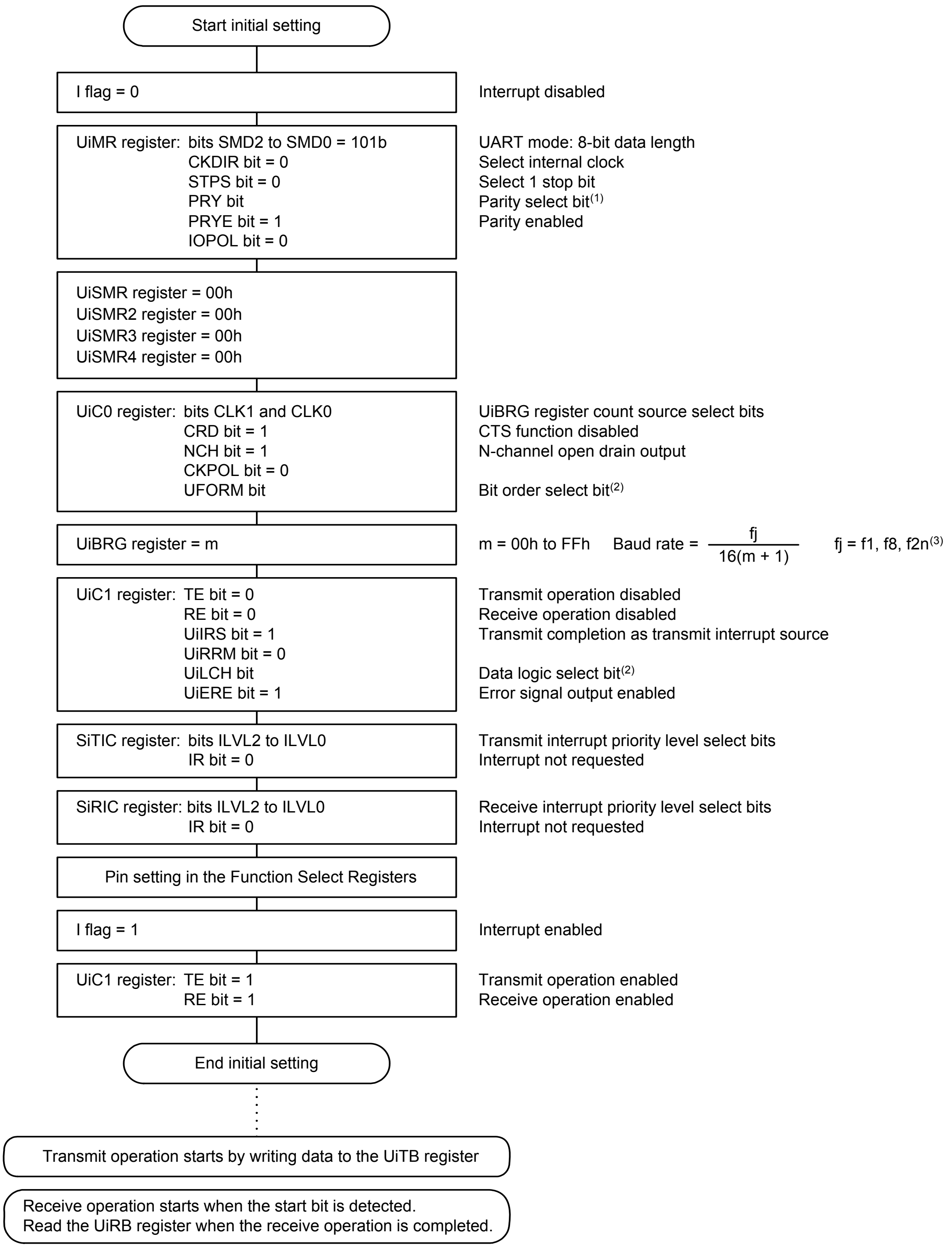
Item	Specification
Data format	• Data length 8-bit UART mode • One stop bit • Direct format: Parity: even Data logic: direct (not inverted) Bit order: LSB first • Inverse format: Parity: odd Data logic: inverse (inverted) Bit order: MSB first
Baud rate	Set the CKDIR bit in the UiMR register is 0 (internal 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 17.20 Pin Settings in SIM Mode

Port	Function	Bit Setting			
		PD6, PD7, PD9 Registers(2)	PSC, PSC3 Registers	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	–	PSC3_6 = 0	–	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.

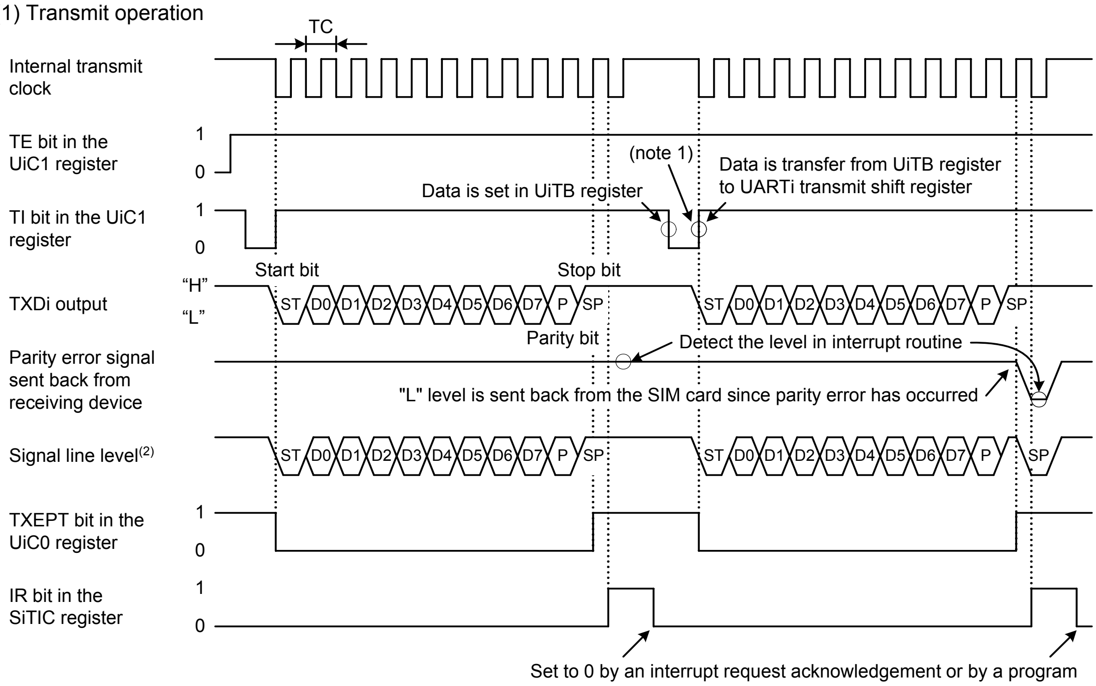


i = 0 to 4

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.

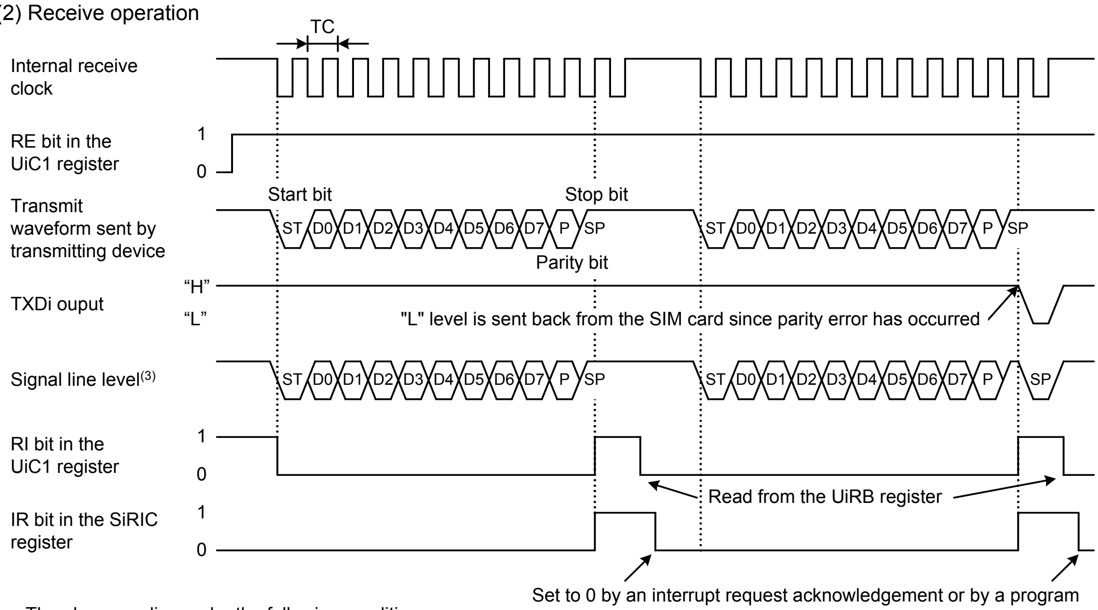
Figure 17.32 Register Settings in SIM Mode

(1) Transmit operation



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)

(2) Receive operation



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

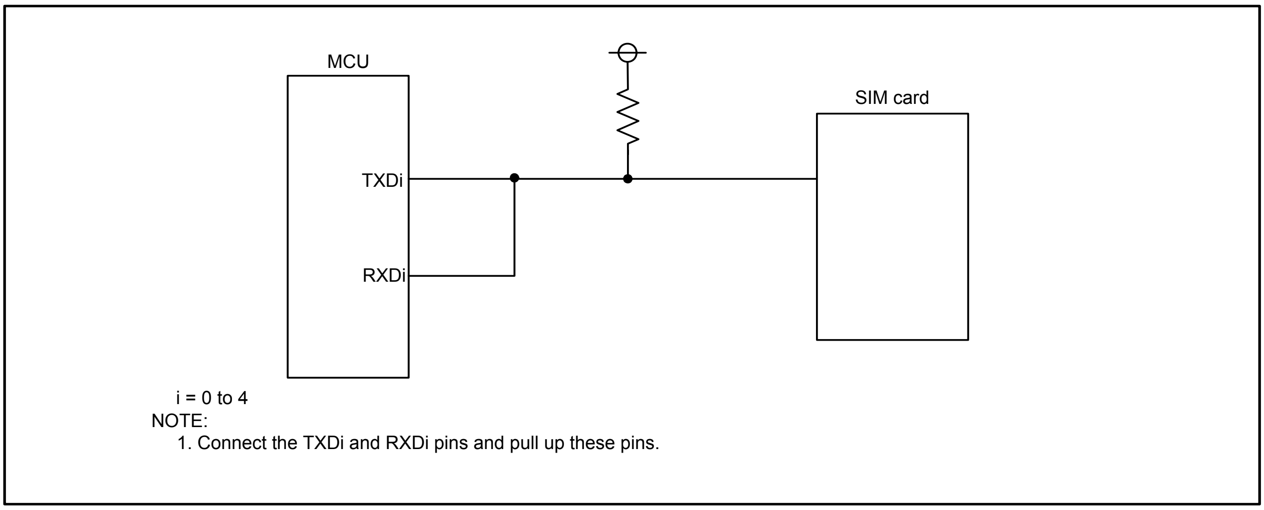


Figure 17.34 SIM Interface Connection

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.

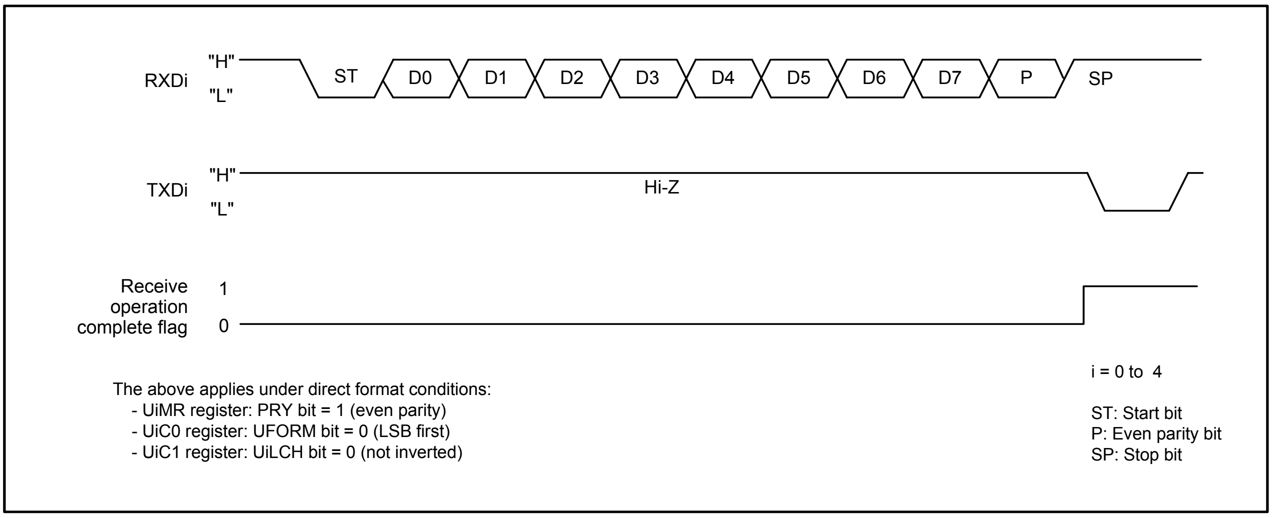


Figure 17.35 Parity Error Signal Output Timing

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).

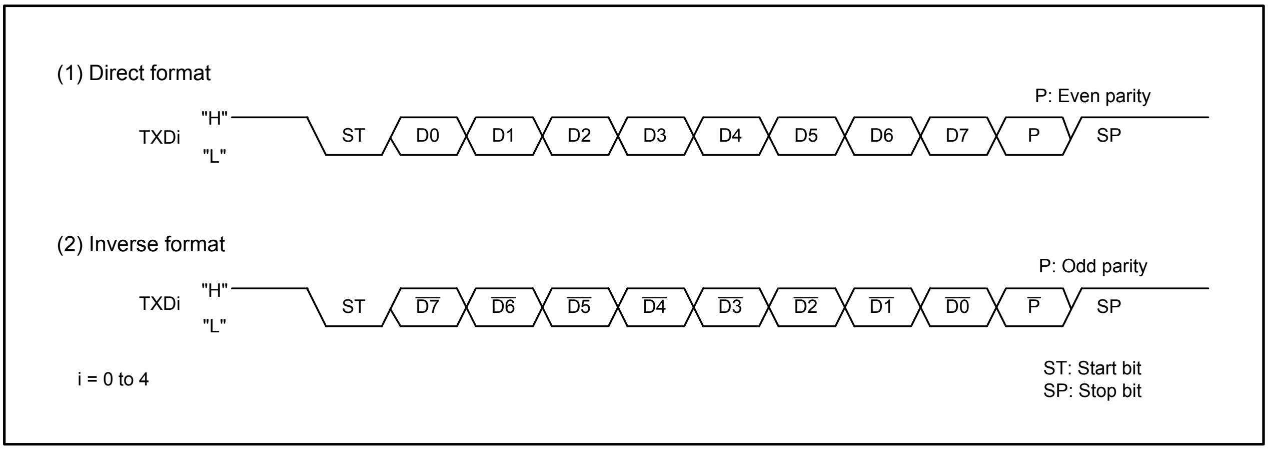


Figure 17.36 SIM Interface Formats

17.1.7 Special Mode 5 (IrDA mode) • • • UART0

Input and output data in clock asynchronous mode are converted into the format supporting IrDA physical layer specification v.1.0. The UART0 transmit data is encoded and output in the RZI (Return to Zero Inverted) format. Input data in the RZI format is decoded to the NRZ (None Return to Zero) format and becomes the UART0 reception input data. Refer to the **17.1.2** Clock Asynchronous (UART) Mode for details on clock asynchronous mode.

Table 17.21 lists specifications of IrDA mode. Figure 17.37 shows a block diagram. Figure 17.38 shows a register associated with IrDA mode. Figure 17.39 shows an IrDA operation.

Table 17.21 IrDA Mode Specifications

Item	Specification
“0” output pulse width	• PLSSEL bit in the IRCON register is set to 0 (3/16 of the bit rate) $\frac{3}{16}$ bit time • PLSSEL bit is set to 1 (set by bits IRPD0, IRPD1, IRCK) Selectable among $\frac{1}{fi}$, $\frac{2}{fi}$, $\frac{4}{fi}$, $\frac{8}{fi}$ fi = f1 or f8
“0” input pulse width	Input the pulse which is longer than $\frac{3}{fi}$
I/O polarity	Encode logic “0” to a high pulse, decode a high pulse as logic “0” Encode logic “0” to a low pulse, decode a low pulse as logic “0”

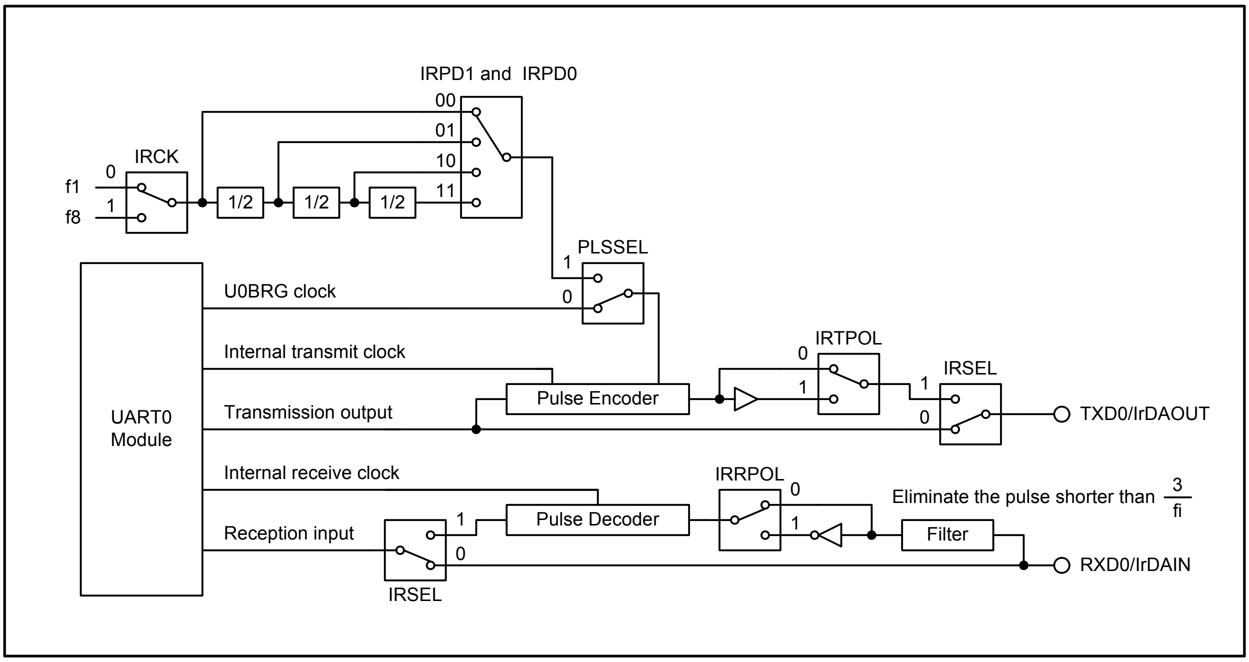


Figure 17.37 IrDA Mode Block Diagram

IrDA Control Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
IRCON	0372h	X000 0000b

Bit Symbol	Bit Name	Function	RW
IRSEL	IrDA select bit	0: TXD0, RXD0 1: IrDAOUT, IrDAIN	RW
PLSSEL	Logic "0" output pulse width select bit	0: 3/16 of the bit rate 1: Set by bits IRPD0, IRPD1, IRCK	RW
IRTPOL	IrDAOUT output polarity switch bit	0: Encode logic "0" as a high pulse 1: Encode logic "0" as a low pulse	RW
IRRPOL	IrDAIN input polarity switch bit	0: Decode high pulse as logic "0" 1: Decode low pulse as logic "0"	RW
IRPD0	Logic "0" output pulse width set bits	b5 b4 0 0: 1/fi (fi = f1, f8) 0 1: 2/fi 1 0: 4/fi 1 1: 8/fi	RW
IRPD1			
IRCK	Logic "0" output pulse count source select bit[1]	0: f1 1: f8	RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTE
1. IRCK bit is enabled when the PLSSEL bit is set to 1.

Figure 17.38 IRCON Register

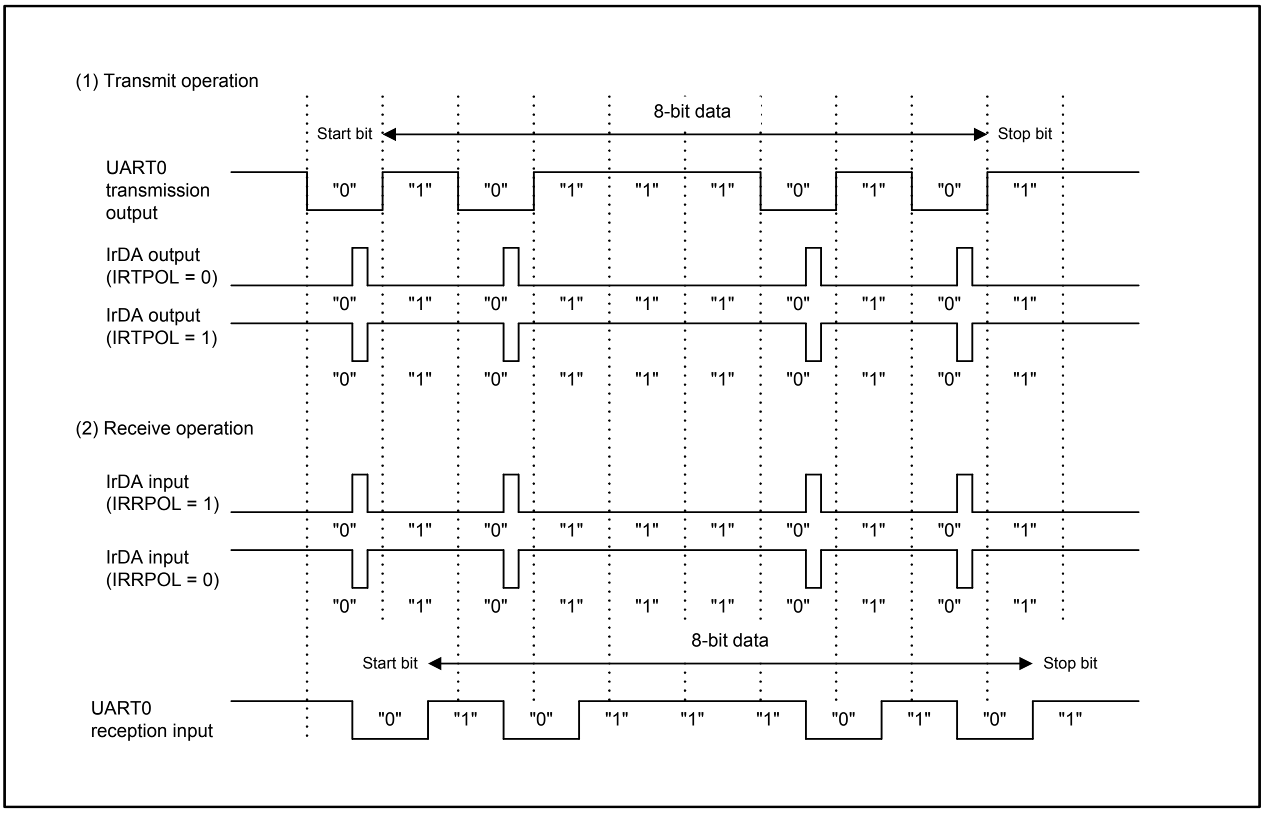
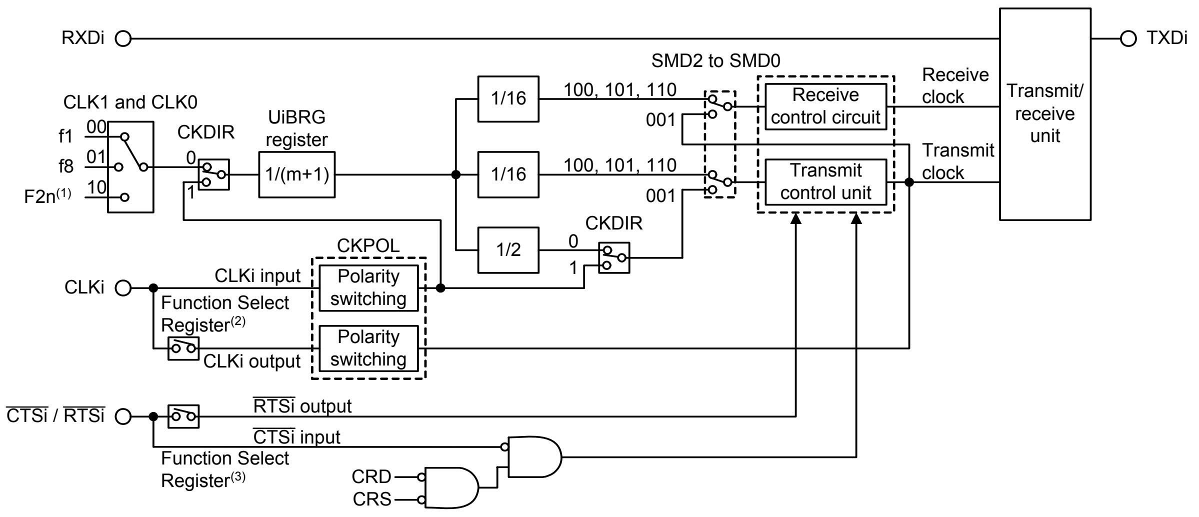


Figure 17.39 IrDA Operation

17.2 UART5 and UART6

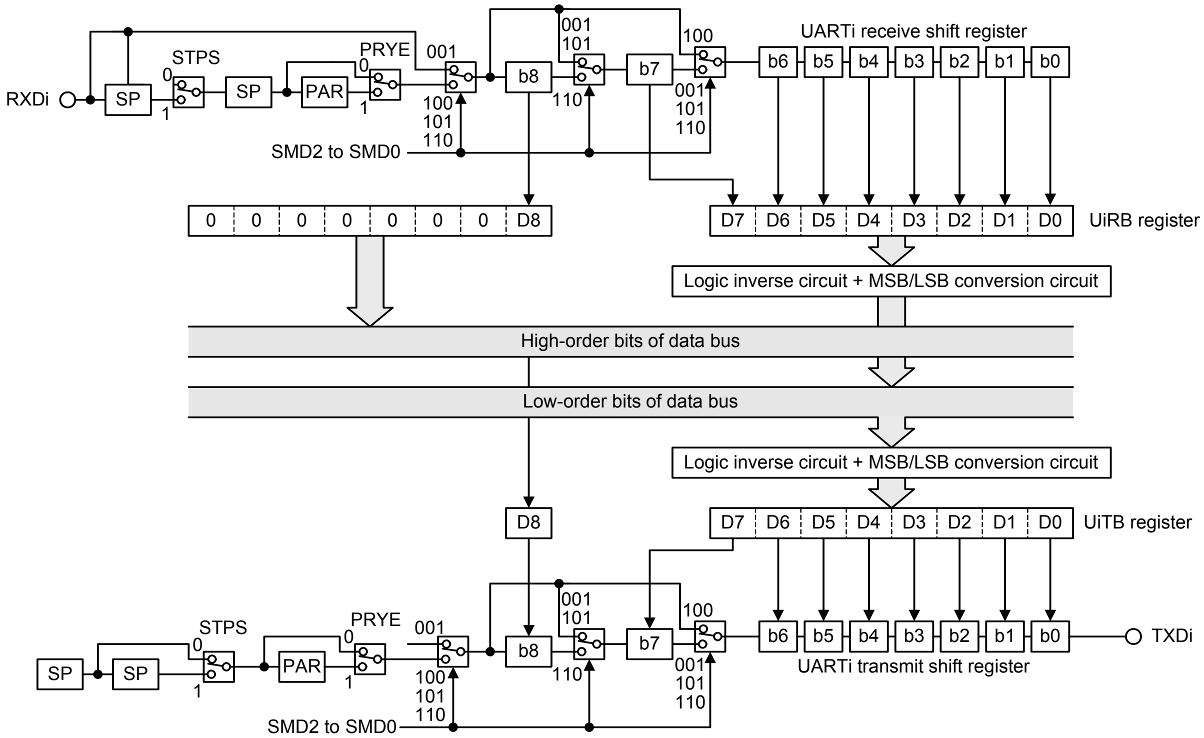
Figure 17.40 shows a UART5 and UART6 block diagram. Figures 17.41 to 17.45 show the registers associated with UART5 and UART6. Refer to the tables listing register and pin settings in each mode. Refer to **11.11 Intelligent I/O, CAN, UART5, UART6, and INT6 to INT8 Interrupts** for details on UART5 and UART6 transmit/receive interrupts.



m: Setting value of the UiBRG register

NOTES:

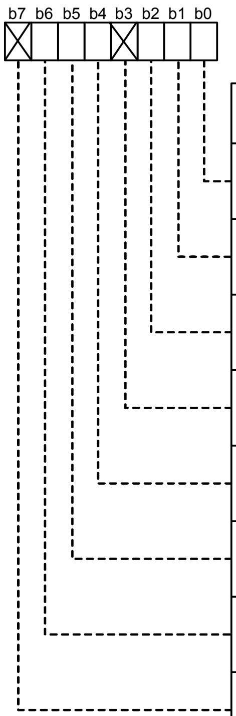
1. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).
2. Select either I/O port (CLKi input) or CLKi output in the Function Select Registers. (Refer to the chapter **Programmable I/O Ports**.)
3. Select either I/O port or $\overline{\text{RTSi}}$ output in the Function Select Registers. (Refer to the chapter **Programmable I/O Ports**.)



i = 5, 6
SP: Stop bit
PAR: Parity bit
SMD2 to SMD0, STPS, PRYE, CKDIR: bits in the UiMR register
CLK1 to CLK0, CKPOL, CRD, CRS: bits in the UiC0 register

Figure 17.40 UART5 and UART6 Block Diagram

UART5, UART6 Input Pin Function Select Register



Symbol	Address	After Reset
U56IS	01D1h	X000 X000b

Bit Symbol	Bit Name	Function	RW
U5CLK	CLK5 input pin select bit[1]	0: P7_7 1: P15_1	RW
U5RXD	RXD5 input pin select bit[1]	0: P8_0 1: P15_2	RW
U5CTS	$\overline{\text{CTS5}}$ input pin select bit[1]	0: P8_1 1: P15_3	RW
– (b3)	Unimplemented. Write 0. Read as undefined value.		–
U6CLK	CLK6 input pin select bit[2]	0: P15_6 1: P12_1	RW
U6RXD	RXD6 input pin select bit[2]	0: P15_5 1: P12_2	RW
U6CTS	$\overline{\text{CTS6}}$ input pin select bit[2]	0: P15_7 1: P12_3	RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Set bits U5CLK, U5RXD, and U5CTS to 0 in the 100-pin package.
2. Bits U6CLK, U6RXD, and U6CTS are provided in the 144-pin package only.

Figure 17.41 U56IS Register

UARTi Transmit/Receive Mode Register (i = 5, 6)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	After Reset
U5MR, U6MR	01C0h, 01C8h	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 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 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	Enables when PRYE = 1 0: Odd parity 1: Even parity	RW
PRYE	Parity enable bit	0: Parity disabled 1: Parity enabled	RW
– (b7)	Reserved bit	Set to 0	RW

Figure 17.42 U5MR and U6MR Registers

UARTi Transmit/Receive Control Register 0 (i = 5, 6)

b7 b6 b5 b4 b3 b2 b1 b0
(b5 = 0)

Symbol	Address	After Reset
U5C0, U6C0	01C4h, 01CCh	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
– (b5)	Reserved bit	Set to 0	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(3)	0: LSB first 1: MSB first	RW

NOTES:
1. Set bits CLK1 and CLK0 before setting the UiBRG register.
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. 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 0 when bits SMD2 to SMD0 are set to 100b (UART mode, 7-bit data length) or 110b (UART mode, 9-bit data length).

UARTi Baud Rate Register(1, 2) (i = 5, 6)

b7 b0

Symbol	Address	After Reset
U5BRG, U6BRG	01C1h, 01C9h	Undefined

Function	Setting Range	RW
If the setting value is *n*, the UiBRG register divides the 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.

Figure 17.43 U5C0 and U6C0 Registers, U5BRG and U6BRG Registers

UART5, UART6 Transmit/Receive Control Register

b7 b6 b5 b4 b3 b2 b1 b0

X 0 0 0

Symbol	Address	After Reset
U56CON	01D0h	X000 0000b

Bit Symbol	Bit Name	Function	RW
U5IRS	UART5 transmit interrupt source select Bit	0: No data in the U5TB register (TI = 1) 1: Transmit operation is completed (TXEPT = 1)	RW
U6IRS	UART6 transmit interrupt source select Bit	0: No data in the U6TB register (TI = 1) 1: Transmit operation is completed (TXEPT = 1)	RW
U5RRM	UART5 continuous receive mode enable bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled(1)	RW
U6RRM	UART6 continuous receive mode enable bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled(1)	RW
– (b6-b4)	Reserved bits	Set to 0	RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:

1. When the UiRRM bit (i = 5, 6) is set to 1, set the CKDIR bit in the UiMR register to 1 (external clock) and also disable the RTS function.

UARTi Transmit/Receive Control Register 1 (i = 5, 6)

b7 b6 b5 b4 b3 b2 b1 b0

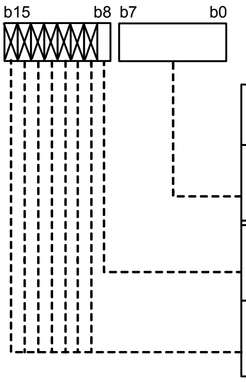
X X X X

Symbol	Address	After Reset
U5C1, U6C1	01C5h, 01CDh	XXXX 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
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

Figure 17.44 U56CON Register, U5C1 and U6C1 Registers

UARTi Transmit Buffer Register[1] (i = 5, 6)

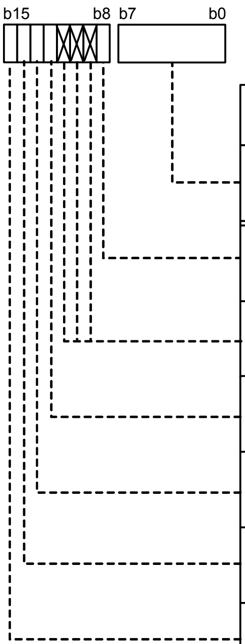


Symbol	Address	After Reset
U5TB, U6TB	01C3h - 01C2h, 01CBh - 01CAh	Undefined

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 = 5, 6)



Symbol	Address	After Reset
U5RB, U6RB	01C7h - 01C6h,01CFh - 01CEh	Undefined

Bit Symbol	Bit Name	Function	RW
– (b7-b0)	——	Receive data (D7 to D0)	RO
– (b8)	——	Receive data (D8)	RO
– (b11-b9)	Unimplemented. Write 0. Read as undefined value.		–
OER	Overrun error flag[1]	0 : No overrun error 1 : Overrun error	RO
FER	Framing error flag[1, 2]	0 : No framing error 1 : Framing error	RO
PER	Parity error flag[1, 2]	0 : No parity error 1 : Parity error	RO
SUM	Error sum flag[1, 2]	0: No error occurred 1: Error occurred	RO

NOTES:
1. 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.
2. Bits FER, PER, and SUM are disabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous mode) . A read from these bits returns undefined value.

Figure 17.45 U5TB and U6TB Registers, U5RB and U6RB Registers

17.2.1 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 17.22 lists specifications of clock synchronous mode. Table 17.23 lists pin settings. Figure 17.46 shows register settings. Figure 17.47 shows an example of a transmit and receive operation when an internal clock is selected. Figure 17.48 shows an example of a receive operation when an external clock is selected.

Table 17.22 Clock Synchronous Mode Specifications

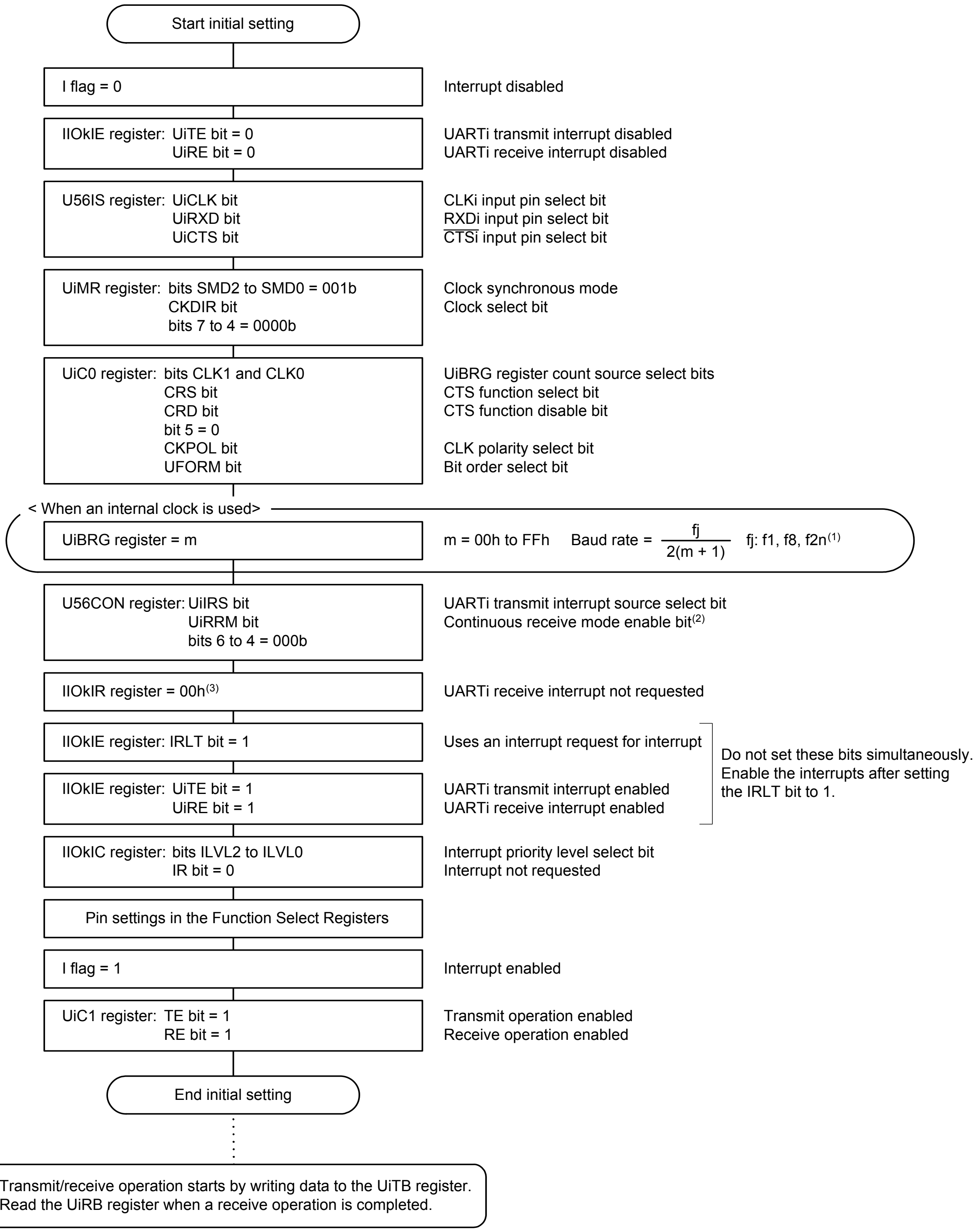
Item	Specification
Data format	Data length: 8 bits long
Serial clock	Internal clock or external clock can be selected with the CKDIR bit in the UiMR register (i = 5 and 6).
Baud rate	• When the CKDIR bit is set to 0 (internal clock): fj / (2 (m + 1)) fj = f1, f8, f2n(1) m: setting value of the UiBRG 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 $\overline{CTSi}$ 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, $\overline{RTSi}$ 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 U56CON 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 • 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 U5RR bit in the IIO0IR register and the U6RR bit in the IIO9IR register remain unchanged as 0 (interrupt not requested).

Table 17.23 Pin Settings in Clock Synchronous Mode

Port	Function	Bit Setting						
		PD7, PD8, PD12, PD15 Registers	U56IS Register	PSE1, PSE2 Registers	PSD1, PSD2 Registers	PSC, PSC2, PSC6 Registers	PSL1, PSL2, PSL6, PSL9 Registers	PS1, PS2, PS6, PS9 Registers [1]
P7_6	TXD5 output[2]	–	–	PSE1_6 = 1	PSD1_6 = 1	PSC_6 = 0	PSL1_6 = 0	PS1_6 = 1
P7_7	CLK5 input	PD7_7 = 0	U5CLK = 0	–	–	–	–	PS1_7 = 0
	CLK5 output	–	–	PSE1_7 = 0	PSD1_7 = 1	–	PSL1_7 = 1	PS1_7 = 1
P8_0	RXD5 input	PD8_0 = 0	U5RXD = 0	–	–	–	–	PS2_0 = 0
P8_1	CTS5 input	PD8_1 = 0	U5CTS = 0	–	–	–	–	PS2_1 = 0
	RTS5 output	–	–	PSE2_1 = 0	PSD2_1 = 1	PSC2_1 = 1	PSL2_1 = 1	PS2_1 = 1
P12_0	TXD6 output[2]	–	–	–	–	PSC6_0 = 1	PSL6_0 = 0	PS6_0 = 1
P12_1	CLK6 input	PD12_1 = 0	U6CLK = 1	–	–	–	–	PS6_1 = 0
	CLK6 output	–	–	–	–	PSC6_1 = 1	PSL6_1 = 0	PS6_1 = 1
P12_2	RXD6 input	PD12_2 = 0	U6RXD = 1	–	–	–	–	–
P12_3	CTS6 input	PD12_3 = 0	U6CTS = 1	–	–	–	–	PS6_3 = 0
	RTS6 output	–	–	–	–	PSC6_3 = 1	PSL6_3 = 0	PS6_3 = 1
P15_0	TXD5 output[2]	–	–	–	–	–	PSL9_0 = 1	PS9_0 = 1
P15_1	CLK5 input[3]	PD15_1 = 0	U5CLK = 1	–	–	–	–	PS9_1 = 0
	CLK5 output	–	–	–	–	–	PSL9_1 = 1	PS9_1 = 1
P15_2	RXD5 input[3]	PD15_2 = 0	U5RXD = 1	–	–	–	–	–
P15_3	CTS5 input[3]	PD15_3 = 0	U5CTS = 1	–	–	–	–	PS9_3 = 0
	RTS5 output	–	–	–	–	–	–	PS9_3 = 1
P15_4	TXD6 output[2]	–	–	–	–	–	PSL9_4 = 1	PS9_4 = 1
P15_5	RXD6 input[3]	PD15_5 = 0	U6RXD = 0	–	–	–	–	–
P15_6	CLK6 input[3]	PD15_6 = 0	U6CLK = 0	–	–	–	–	PS9_6 = 0
	CLK6 output	–	–	–	–	–	–	PS9_6 = 1
P15_7	CTS6 input[3]	PD15_7 = 0	U6CTS = 0	–	–	–	–	PS9_7 = 0
	RTS6 output	–	–	–	–	–	–	PS9_7 = 1

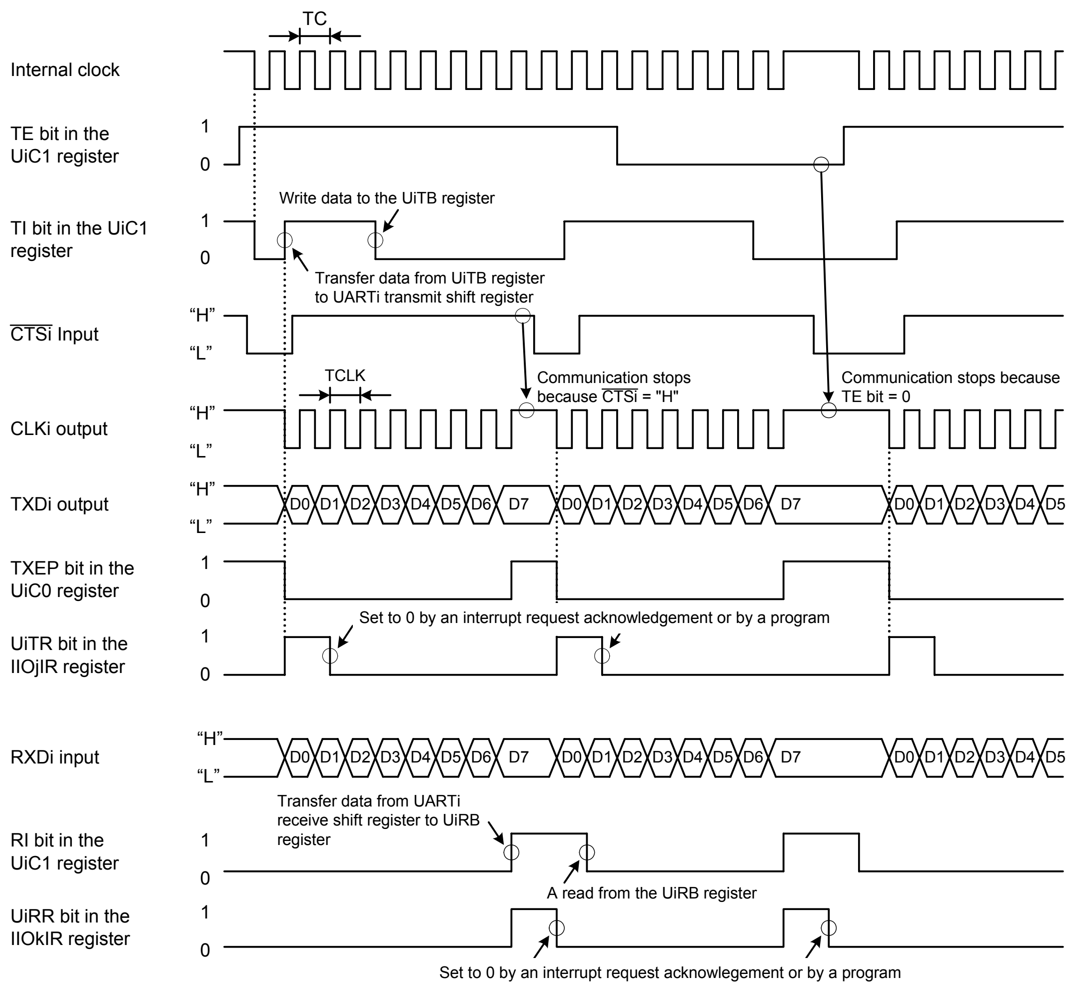
NOTE:
1. Set registers PS1, PS2, PS6 and PS9 after setting the other registers.
2. After UARTi (i = 5, 6) operating mode is selected in the UiMR register and the pin function is set in the Function Select Registers, the TXDi pin outputs an “H” signal until a transmit operation starts.
3. Set both the IPSB_k bit in the IPSB register and the IPS2 bit in the IPS register to 0, when the port P15_k (k = 0 to 7) is used for a peripheral function input.



k = 0, 1 when i = 5, k = 9, 10 when i = 6

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.
3. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated. (An interrupt does not be occur.)

Figure 17.46 Register Settings in Clock Synchronous Mode



j = 1, k = 0 when i = 5,   j = 10, k = 9 when i = 6

$$TC = TCLK = \frac{2(m + 1)}{fj}$$

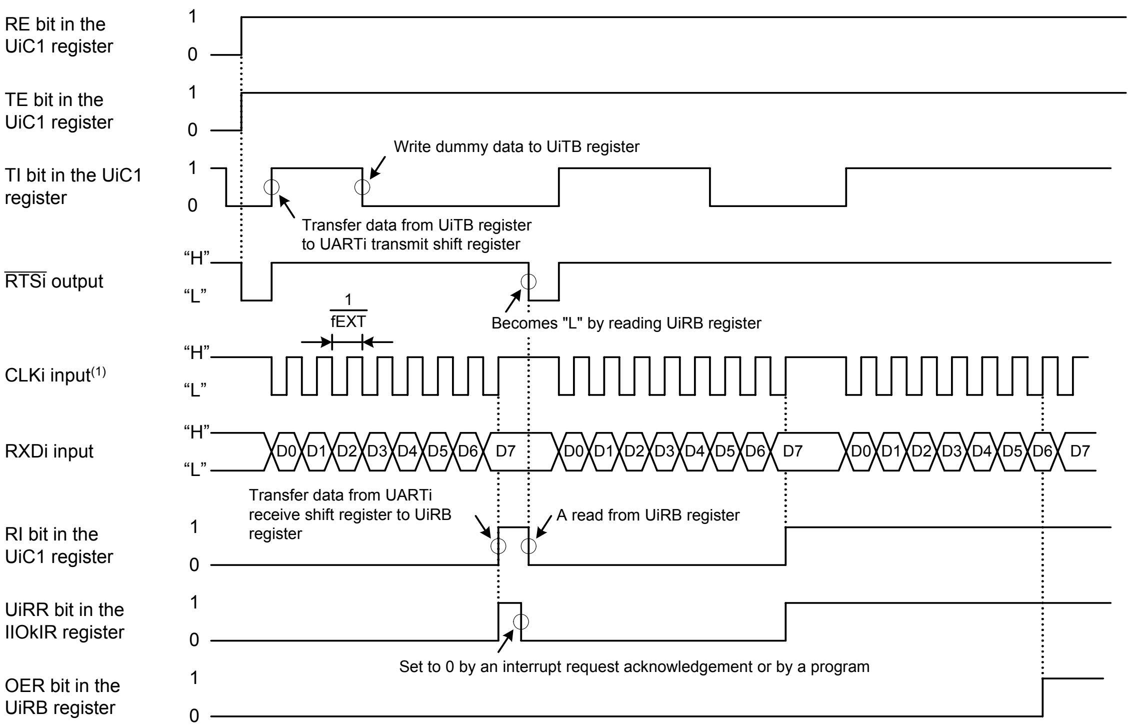
fj = f1, f8, f2n(1)

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)
- U56CON 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 17.47 Transmit and Receive Operation when Internal Clock is Selected



k = 0 when i = 5, k = 9 when i = 6

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)

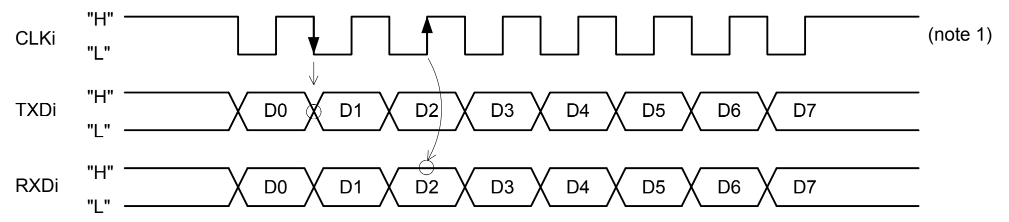
NOTE:
1. Satisfy the following conditions, while the CLKi pin input is "H" before the data receive operation.
   - UiC1 register: TE bit = 1 (transmit operation enabled)
     RE bit = 1 (receive operation enabled)
   - Write dummy data to the UiTB register

Figure 17.48 Receive Operation when External Clock is Selected

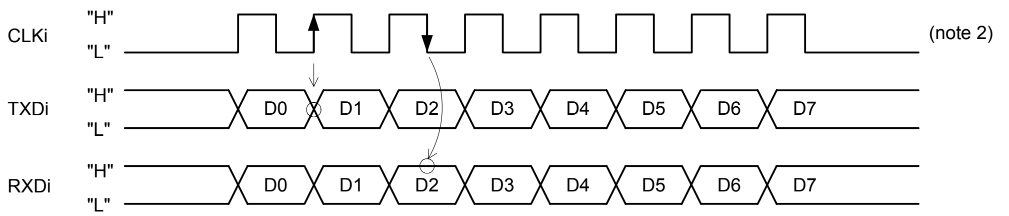
17.2.1.1 CLK Polarity

As shown in Figure 17.49, the CKPOL bit in the UiC0 register (i = 5, 6) determines the polarity of the serial clock.

(1) When the CKPOL bit in the UiC0 register (i = 5, 6) 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:
- UiC0 regsiter: UFORM bit = 0 (LSB first).

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 17.49 Serial Clock Polarity

17.2.1.2 LSB First or MSB First

As shown in Figure 17.50, the UFORM bit in the UiC0 register (i = 5, 6) determines a bit order.

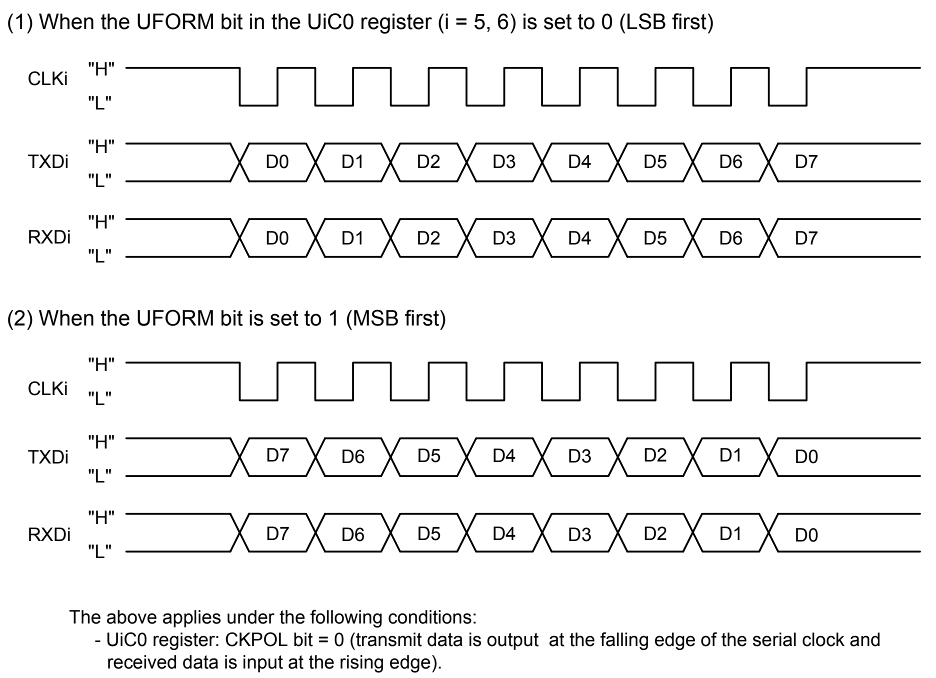


Figure 17.50 Bit order (8-Bit Data Length)

17.2.1.3 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 = 5 and 6) 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 U56CON 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.

17.2.1.4 CTS/RTS Function

- CTS Function

  Transmit and receive operation is controlled by using the input signal to the $\overline{\mathrm{CTSi}}$ pin (i = 5 and 6). 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 $\overline{\mathrm{CTSi}}$ 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 $\overline{\mathrm{CTSi}}$ 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 $\overline{\mathrm{RTSi}}$ pin. To use the RTS function, select the $\overline{\mathrm{RTSi}}$ pin in the Function Select Register.

  With the RTS function used, the $\overline{\mathrm{RTSi}}$ 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)

17.2.1.5 Procedure When the Communication Error is Occurred

Follow the procedure below when a communication error is occurred in clock synchronous mode.

(1) Set the TE bit in the UiC1 register (i = 5 and 6) 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).

17.2.2 Clock Asynchronous (UART) Mode

Full-duplex asynchronous serial communications are allowed in this mode. Table 17.24 lists specifications of UART mode. Table 17.25 lists pin settings. Figure 17.51 shows register settings. Figure 17.52 shows an example of a transmit operation. Figure 17.53 shows an example of a receive operation.

Table 17.24 UART Mode Specifications

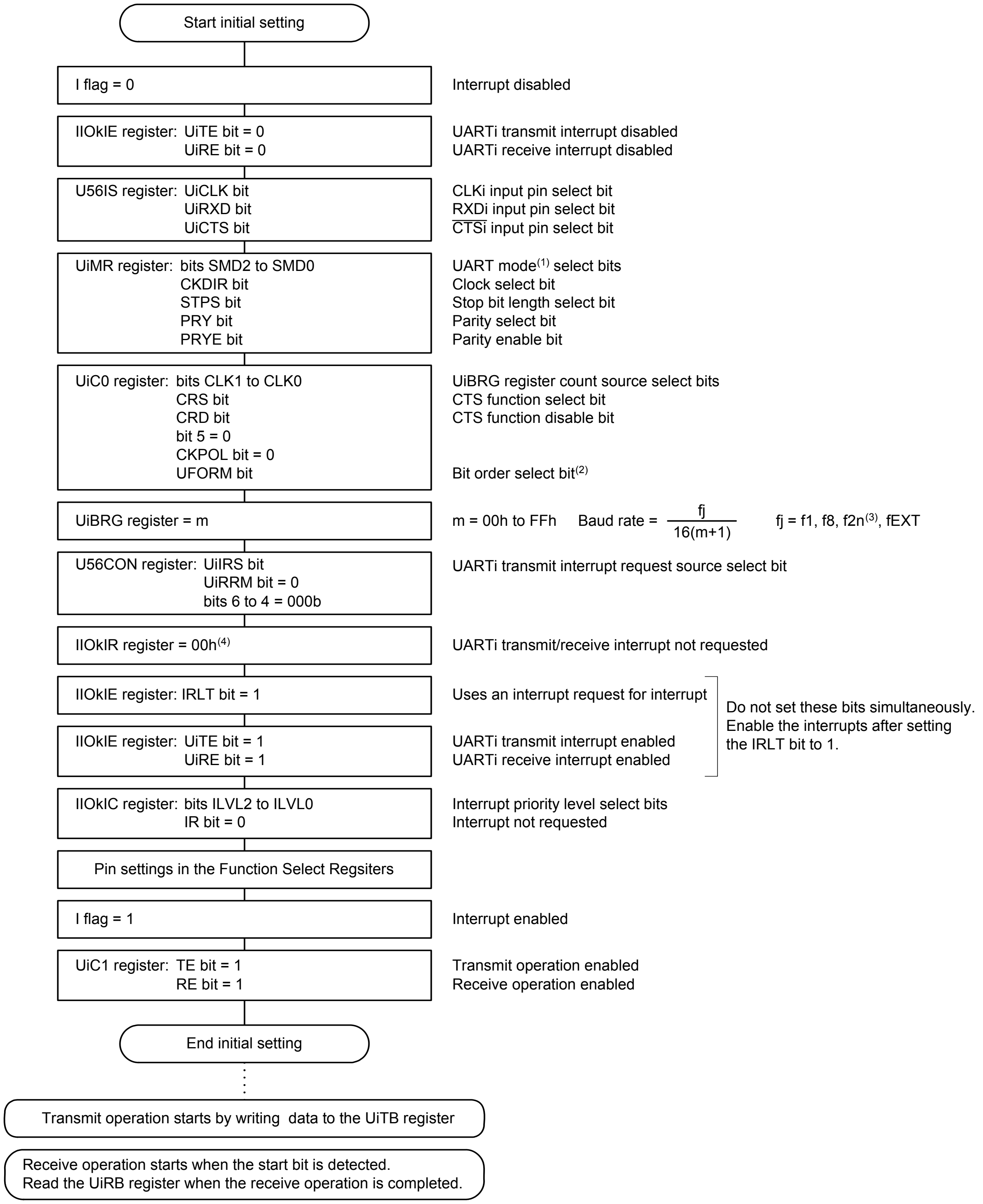
Item	Specification
Data format	• Data length: selectable among 7 bits, 8 bits, or 9 bits long • Start bit: 1 bit long • Parity bit: selectable among odd, even, or none • Stop bit: selectable from 1 bit or 2 bits long
Baud rate	fj / (16 (m + 1)) fj = f1, f8, f2n(1), fEXT m: setting value of the UiBRG register (00h to FFh) (i = 5, 6) 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 U56CON 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

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, the content of the UiRB register is undefined. The U5RR bit in the IIO0IR register and the U6RR bit in the IIO9IR register remain unchanged as 0 (interrupt not requested).

Table 17.25 Pin Settings in UART Mode

Port	Function	Bit Setting						
		PD7, PD8, PD12, PD15 Registers	U56IS Register	PSE1, PSE2 Registers	PSD1, PSD2 Registers	PSC, PSC2, PSC6 Registers	PSL1, PSL2, PSL6, PSL9 Registers	PS1, PS2, PS6, PS9 Registers (1)
P7_6	TXD5 output(2)	–	–	PSE1_6 = 1	PSD1_6 = 1	PSC_6 = 0	PSL1_6 = 0	PS1_6 = 1
P7_7	CLK5 input	PD7_7 = 0	U5CLK = 0	–	–	–	–	PS1_7 = 0
P8_0	RXD5 input	PD8_0 = 0	U5RXD = 0	–	–	–	–	PS2_0 = 0
P8_1	$\overline{\text{CTS5}}$ input	PD8_1 = 0	U5CTS = 0	–	–	–	–	PS2_1 = 0
	$\overline{\text{RTS5}}$ output	–	–	PSE2_1 = 0	PSD2_1 = 1	PSC2_1 = 1	PSL2_1 = 1	PS2_1 = 1
P12_0	TXD6 output(2)	–	–	–	–	PSC6_0 = 1	PSL6_0 = 0	PS6_0 = 1
P12_1	CLK6 input	PD12_1 = 0	U6CLK = 1	–	–	–	–	PS6_1 = 0
P12_2	RXD6 input	PD12_2 = 0	U6RXD = 1	–	–	–	–	–
P12_3	$\overline{\text{CTS6}}$ input	PD12_3 = 0	U6CTS = 1	–	–	–	–	PS6_3 = 0
	$\overline{\text{RTS6}}$ output	–	–	–	–	PSC6_3 = 1	PSL6_3 = 0	PS6_3 = 1
P15_0	TXD5 output(2)	–	–	–	–	–	PSL9_0 = 1	PS9_0 = 1
P15_1	CLK5 input(3)	PD15_1 = 0	U5CLK = 1	–	–	–	–	PS9_1 = 0
P15_2	RXD5 input(3)	PD15_2 = 0	U5RXD = 1	–	–	–	–	–
P15_3	$\overline{\text{CTS5}}$ input(3)	PD15_3 = 0	U5CTS = 1	–	–	–	–	PS9_3 = 0
	$\overline{\text{RTS5}}$ output	–	–	–	–	–	–	PS9_3 = 1
P15_4	TXD6 output(2)	–	–	–	–	–	PSL9_4 = 1	PS9_4 = 1
P15_5	RXD6 input(3)	PD15_5 = 0	U6RXD = 0	–	–	–	–	–
P15_6	CLK6 input(3)	PD15_6 = 0	U6CLK = 0	–	–	–	–	PS9_6 = 0
P15_7	$\overline{\text{CTS6}}$ input(3)	PD15_7 = 0	U6CTS = 0	–	–	–	–	PS9_7 = 0
	$\overline{\text{RTS6}}$ output	–	–	–	–	–	–	PS9_7 = 1

NOTES:
1. Set registers PS1, PS2, PS6, and PS9 after setting the other registers.
2. After UARTi (i = 5, 6) 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.
3. Set both the IPSB_k bit in the IPSB register and the IPS2 bit in the IPS register to 0, when the port P15_k (k = 0 to 7) is used for a peripheral function input.

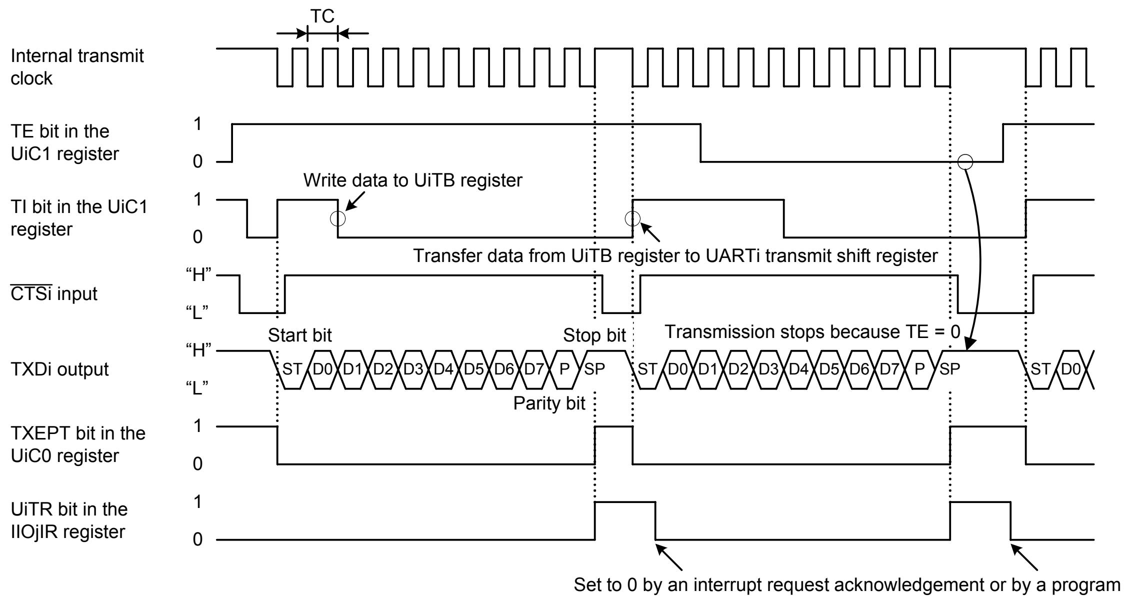


k = 0, 1 when i = 5, k = 9, 10 when i = 6
fEXT: clock input to the CLKi pin when the external clock is selected

NOTES:
1. Set bits SMD2 to SMD0 to the following: 100b (7 bits long), 101b (8 bits long), 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. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated. (An interrupt does not be occur.)

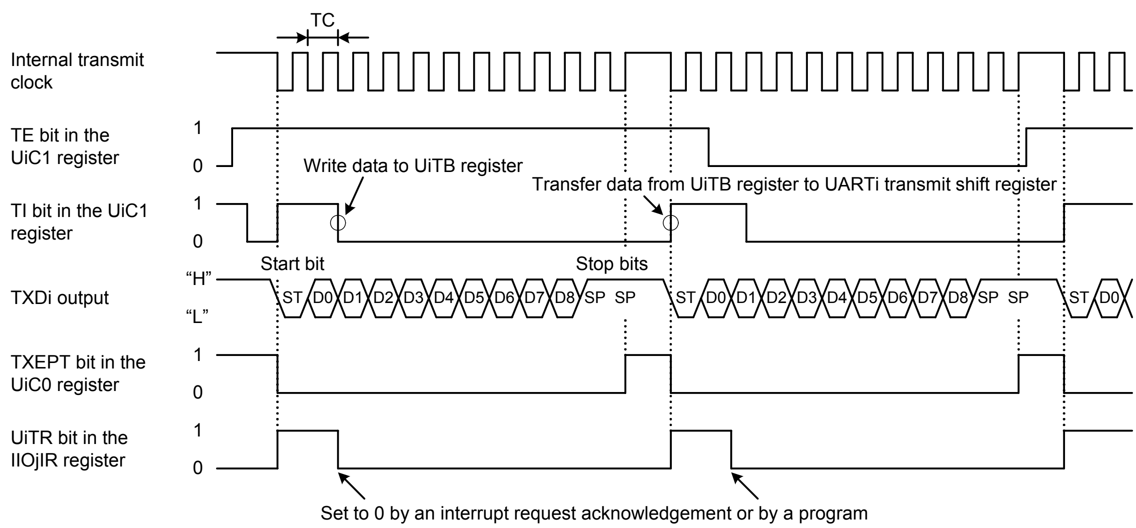
Figure 17.51 Register Settings in UART Mode

(1) Example of the transmit operation timing in 8-bit data length (parity enabled, 1 stop bit)



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)
- U56CON 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)



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)
- U56CON register: UiIRS bit = 0 (transmit interrupt is generated when no data in the UiTB register)

$TC = \frac{16(m + 1)}{fj}$

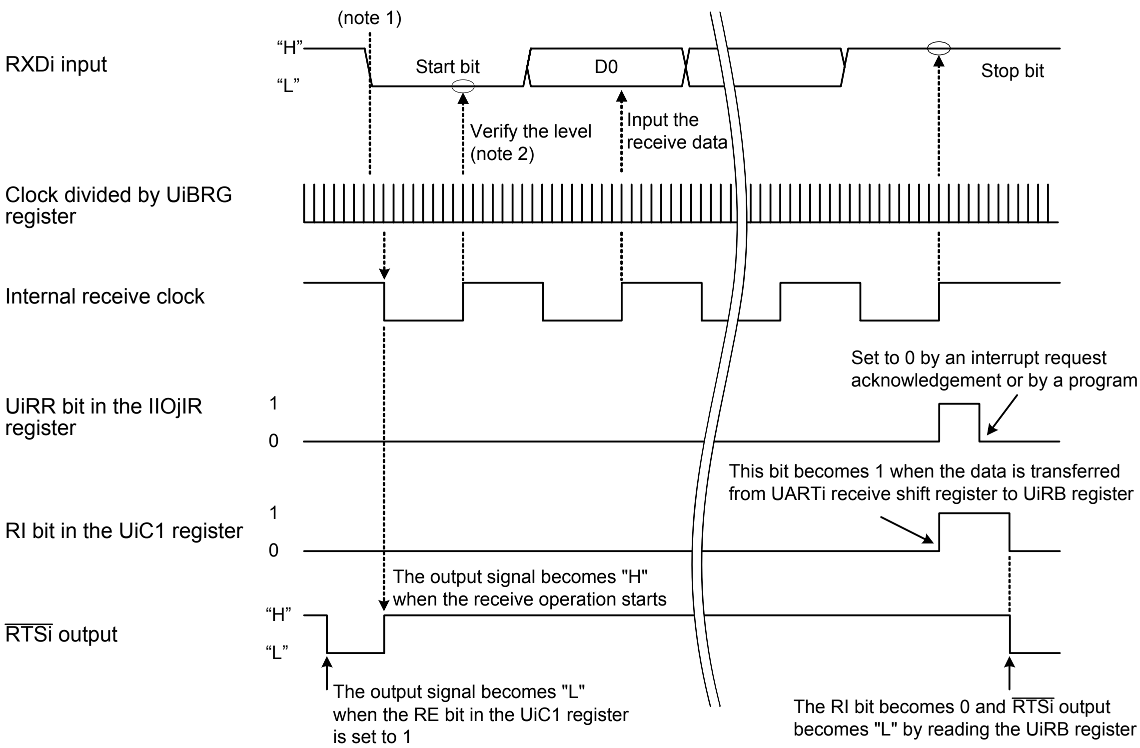
fj: f1, f8, f2n[1], fEXT
fEXT: clock input to the CLKi pin when the external clock is selected
m: setting value of the UiBRG register (00h to FFh)

j = 1 when i = 5, j = 10 when i = 6

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

Figure 17.52 Transmit Operation in UART mode

Example of the receive operation timing (1 stop bit)



j = 0 when i = 5, j = 9 when i = 6

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 17.53 Receive Operation in UART Mode

17.2.2.1 Baud Rate

In UART mode, the baud rate is the clock frequency divided by the setting value of the UiBRG register (i = 5 and 6) and again divided by 16. Table 17.26 lists an example of baud rate setting.

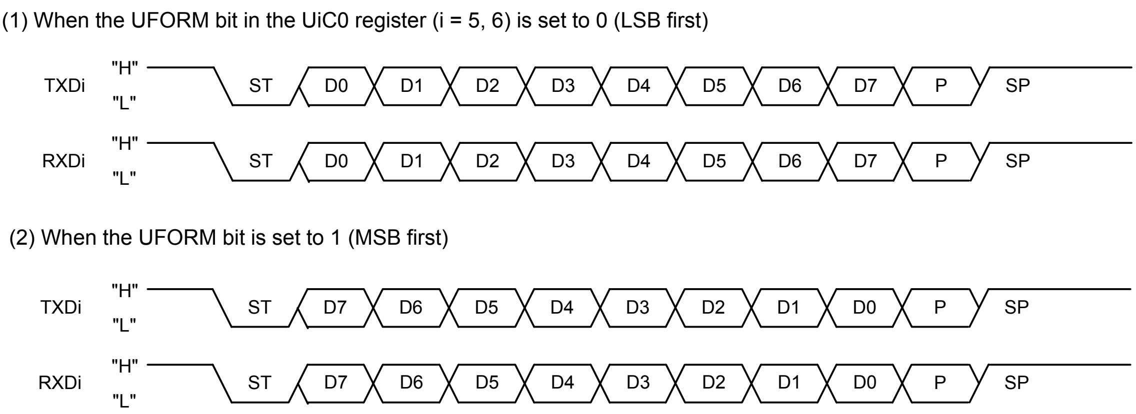
$$\text{Actual baud rate} = \frac{\text{UiBRG register count source}}{16 \times (\text{UiBRG register setting value} + 1)}$$

Table 17.26 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

17.2.2.2 LSB First or MSB First

As shown in Figure 17.54, the UFORM bit in the UiC0 register (i = 5 and 6) determines a bit order. This function is can be used when data length is 8 bits long.



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)

ST: Start bit
P: Parity bit
SP: Stop bit

Figure 17.54 Bit Order

17.2.2.3 CTS/RTS Function

• CTS Function

Transmit operation is controlled by using the input signal to the CTSi pin (i = 5 and 6). 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 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)

When a high-level (“H”) signal is applied to the 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 RTSi pin. To use the RTS function, select the RTSi pin in the Function Select Register.

With the RTS function used, the 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)

17.2.2.4 Procedure When the Communication Error is Occurred

Follow the procedure below when a communication error is occurred in UART mode.

(1) Set the TE bit in the UiC1 register (i = 5 and 6) 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).

18. A/D Converter

NOTE

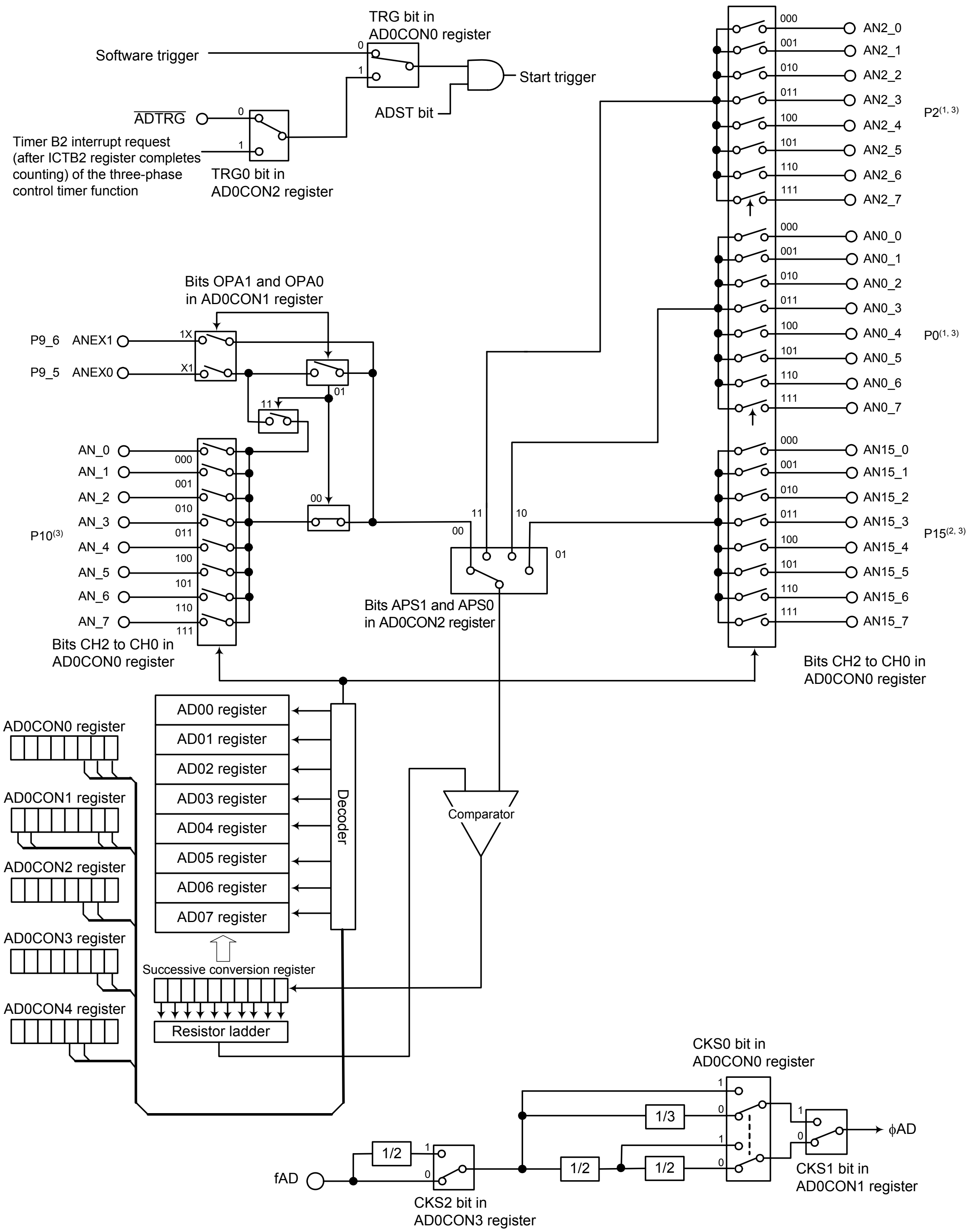
The 144-pin package is described as an example in this chapter.
Pins AN15_0 to AN15_7 are not provided in the 100-pin package.

M32C/87 Group (M32C/87, M32C/87A, M32C/87B) 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 $\overline{ADTRG}$ 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, (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]

b7 b6 b5 b4 b3 b2 b1 b0

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

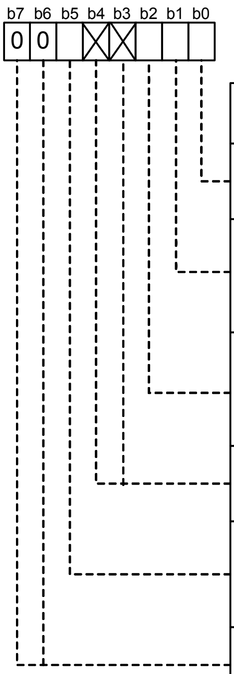
Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bits[2]	Single sweep mode and repeat sweep mode 0 b1 b0 0 0: ANi_0, ANi_1 (i = none, 0, 2, 15) 0 1: ANi_0 to ANi_3 1 0: ANi_0 to ANi_5 1 1: ANi_0 to ANi_7 Repeat sweep mode 1[3] b1 b0 0 0: ANi_0 0 1: ANi_0, ANi_1 1 0: ANi_0 to ANi_2 1 1: ANi_0 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: $\overline{ADTRG}$ 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)

b7 b6 b5 b4 b3 b2 b1 b0
0 0 0

Symbol: AD0CON3 Address: 0395h After Reset: 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]

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

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. Bits MPS11 and MPS10 cannot be set 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)

b15 b8 b7 b0
0 0 0 0 0 0

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			
		IPSB Register	IPS Register	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	–	–		–
P15_0	AN15_0	IPSB_0 = 1	IPS2 = 1(1)	–	–
P15_1	AN15_1	IPSB_1 = 1		–	–
P15_2	AN15_2	IPSB_2 = 1		–	–
P15_3	AN15_3	IPSB_3 = 1		–	–
P15_4	AN15_4	IPSB_4 = 1		–	–
P15_5	AN15_5	IPSB_5 = 1		–	–
P15_6	AN15_6	IPSB_6 = 1		–	–
P15_7	AN15_7	IPSB_7 = 1		–	–

NOTE:
1. When the IPSB_i bit (i = 0 to 7) is set to 1, the peripheral function inputs which are assigned to the P15_i pin are disconnected. When the IPS2 bit is set to 1, the peripheral function inputs which are assigned to pins P15_0 to P15_7 are all disconnected.

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 $\overline{ADTRG}$ 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 that is 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 $\overline{ADTRG}$ 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. (retrigger of external trigger is invalid)
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)

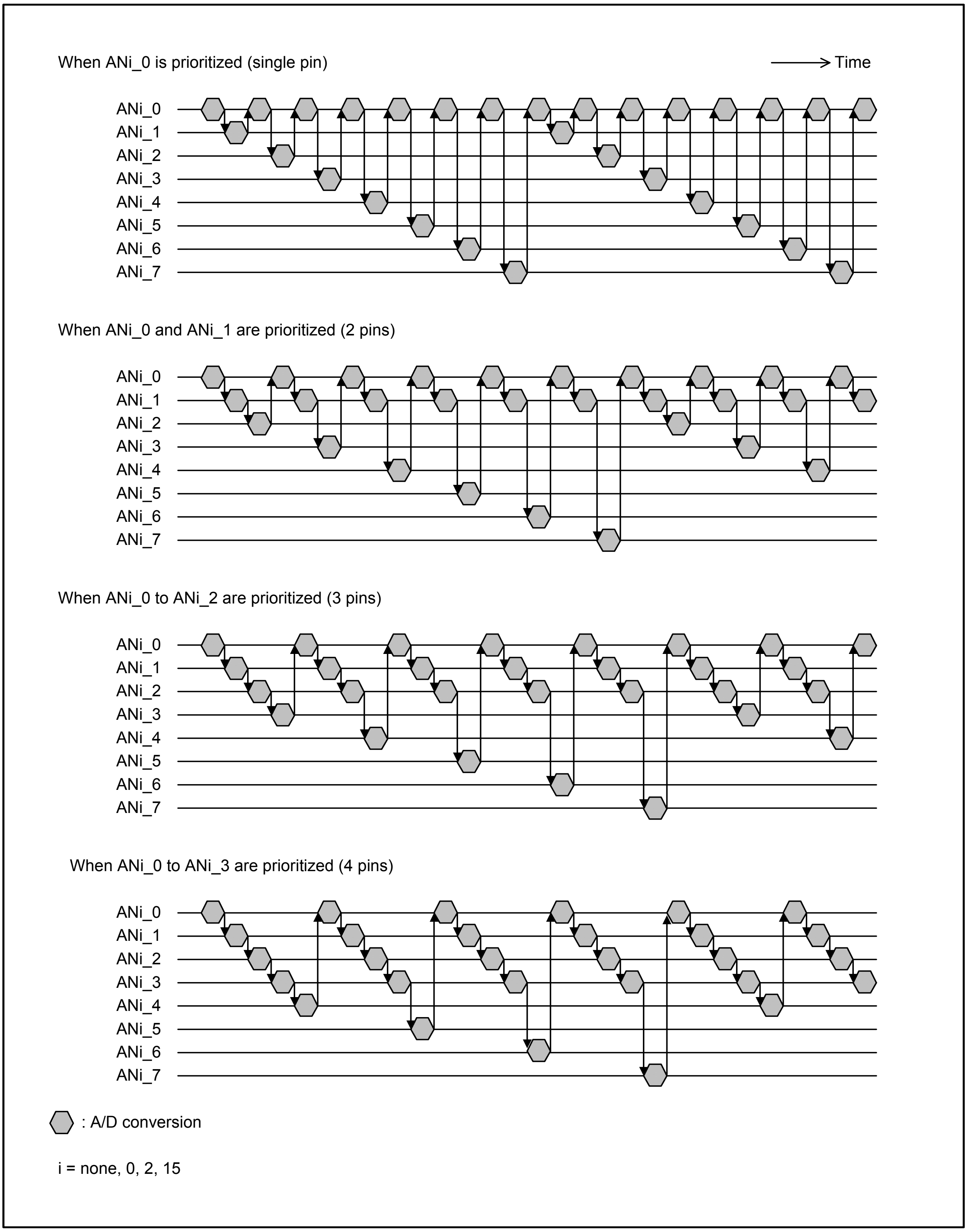


Figure 18.7 Transition Diagram of Pins used in A/D Conversion in Repeat Sweep Mode 1

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 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 that is 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 $\overline{ADTRG}$ 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 $\overline{ADTRG}$
	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 generated.
2. A/D conversion starts over from the beginning, if an external trigger or a hardware trigger is inserted during A/D conversion. (A/D conversion in progress is aborted.)

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 generate an external retrigger or hardware retrigger.

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. The ANEX0 pin cannot be connected 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

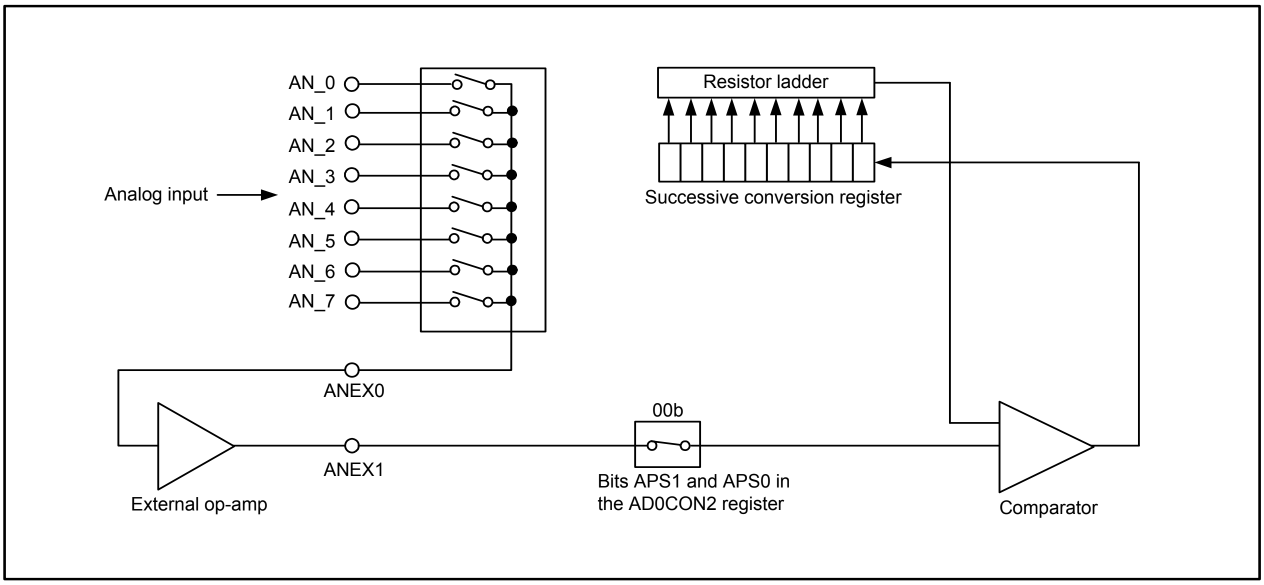


Figure 18.8 Connection Example in External Op-Amp Connection Mode

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 PM24 bit in the PM2 register to 0 (clock selected by the CM07 bit).
  (2) Set the CM07 bit in the CM0 register to 0 (clock selected by the CM21 bit divided by the MCD register).
  (3) 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:

$$VC = VIN\left\{1 - e^{-\frac{1}{C(R0+R)}t}\right\}$$

When t = T, $$VC = VIN - \frac{X}{Y}VIN = VIN\left(1 - \frac{X}{Y}\right)$$

$$e^{-\frac{1}{C(R0+R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)}T = \ln\frac{X}{Y}$$

$$R0 = -\frac{T}{C\ln\frac{X}{Y}} - R$$

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 = 9.0 pF, X = 1, Y = 1024,

$$R0 = -\frac{0.3 \times 10^{-6}}{9.0 \times 10^{-12} \cdot \ln\frac{1}{1024}} - 2.0 \times 10^{3} \cong 2.8 \times 10^{3}\Omega$$

Thus, the allowable output impedance R0 of the sensor equivalent circuit, making the accuracy (error) 1LSB or less, is approximately 2.8 kΩ maximum.

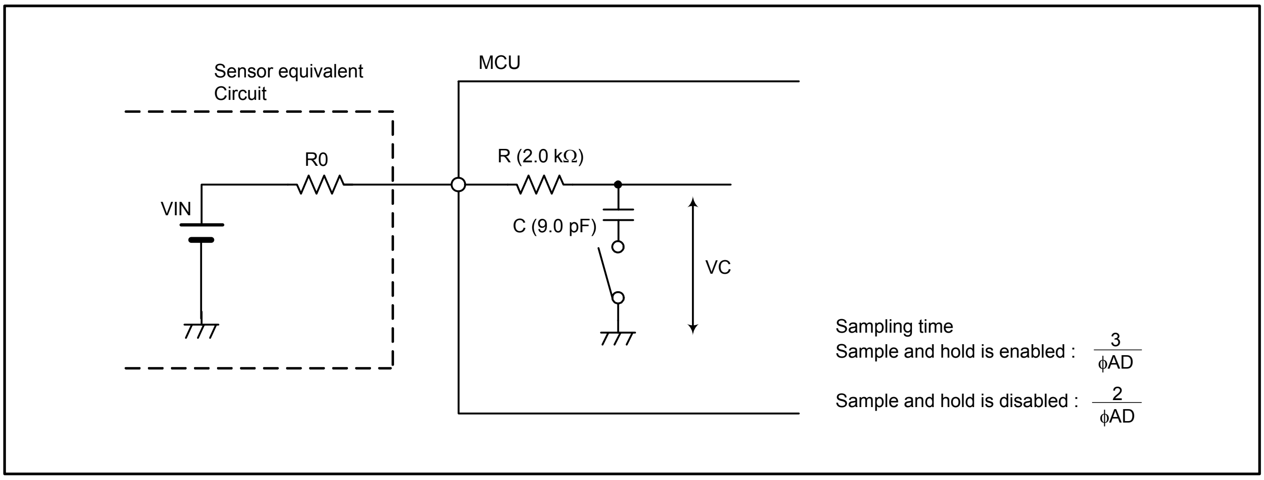


Figure 18.9 Analog Input Pin and External Sensor Equivalent Circuit

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), if bits DATi1 and DATi0 in the DACON1 register are set to 00b. Every time the selected timer underflows, a value in the DAi register is transferred to the DAi buffer and the D/A conversion is performed, if bits DATi1 and DATi0 are set to 01b, 10b, or 11b. The values in the DAi buffer is 00h after reset.
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), the corresponding port cannot be pulled up.
When the D/A converter is not used, set registers DAi and DACON1 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.

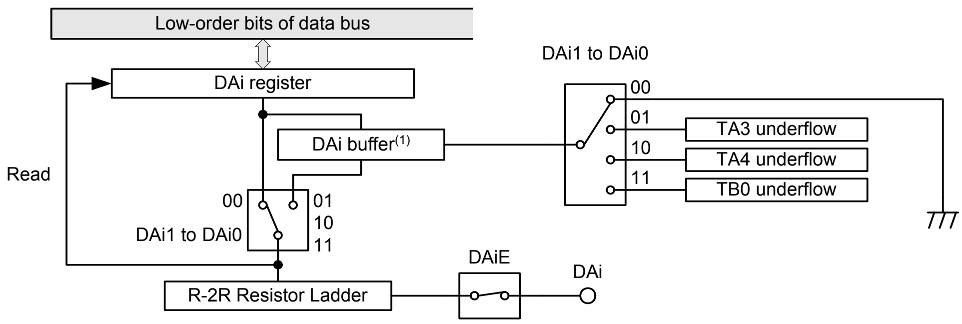
$$V = \frac{VREF \times n}{256} \quad (n = 0 \text{ to } 255)$$

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



i = 0, 1
DAiE: bit in the DACON register
DAi1, DAi0: bits in the DACON1 register
NOTE:
1. When bits DATi1 and DATi0 are set to 01b, 10b or 11b, a value in the DAi register is transferred to the DAi buffer every time the selected timer underflows. The value in the DAi buffer is 00h after reset.

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

b7 b6 b5 b4 b3 b2 b1 b0

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 Control Register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
DACON1	039Dh	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
DAT00	D/A0 conversion timing select bits[1][2]	b1 b0 0 0: When a value is written to D/A register 0 0 1: Timer A3 underflow 1 0: Timer A4 underflow 1 1: Timer B0 underflow	RW
DAT01			RW
DAT10	D/A1 conversion timing select bits[1][2]	b3 b2 0 0: When a value is written to D/A register 1 0 1: Timer A3 underflow 1 0: Timer A4 underflow 1 1: Timer B0 underflow	RW
DAT11			RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Set the selected timer for the conversion timing to timer mode.
2. Set bits DAi1 and DAi0 to 00b when the D/A converter is not used. (i = 0, 1)

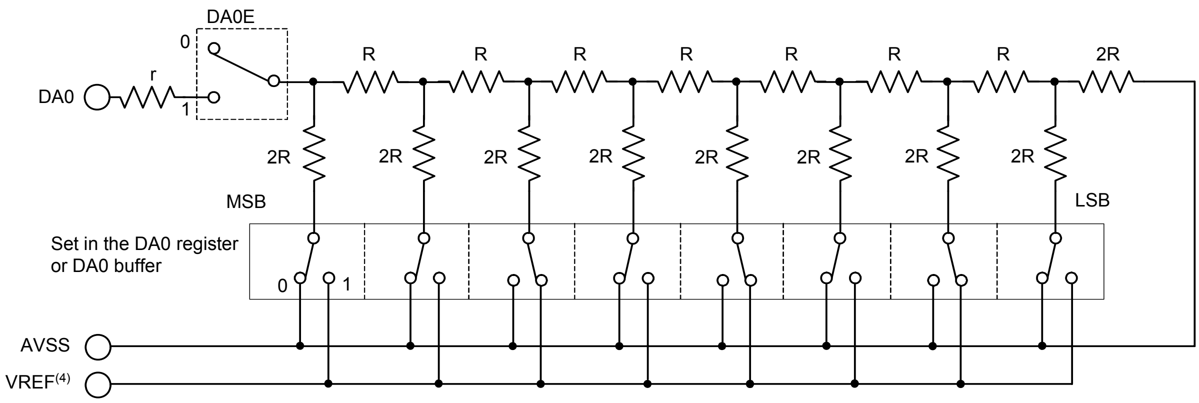
D/A Register i (i = 0, 1)

b7 b0

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, DACON1 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 registers DACON1 and DAi 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.

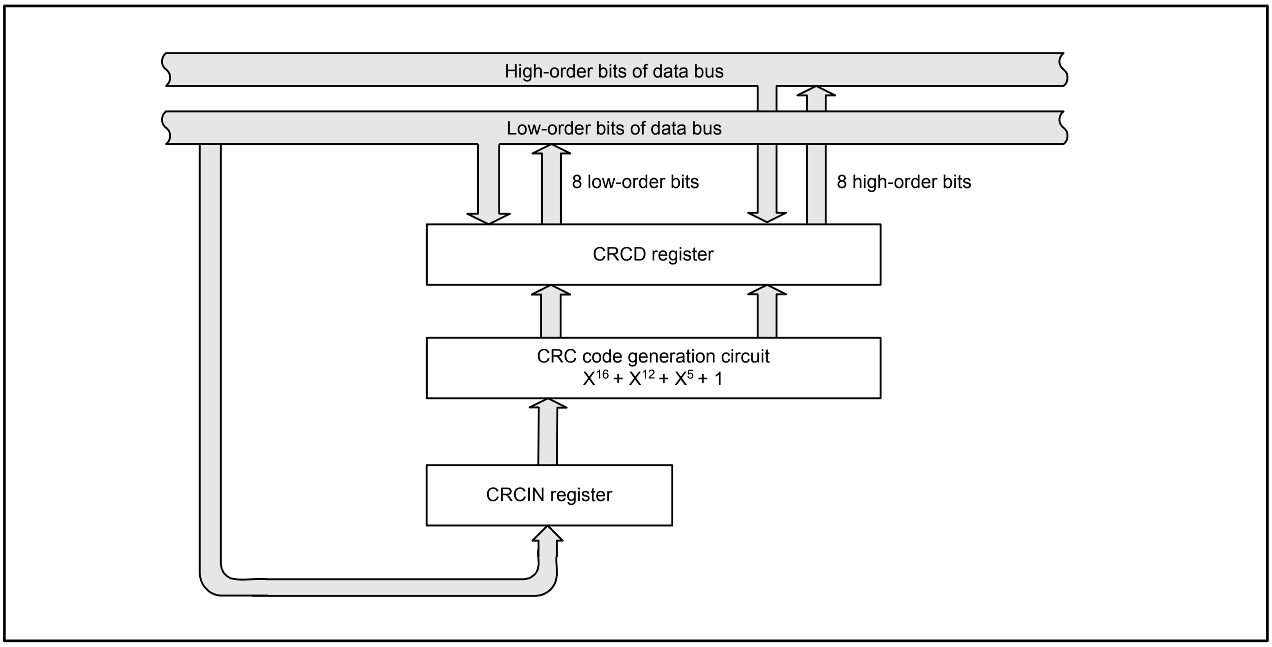


Figure 20.1 CRC Calculation Block Diagram

CRC Data Register

b15 b8 b7 b0

Symbol: CRCD
Address: 037Dh - 037Ch
After Reset: 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: CRCIN
Adddress: 037Eh
After Reset: 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

(2) Set 0000h (default value) → [b15 … b0] CRCD register

(3) Set 01h → [b7 … b0] CRCIN register

Bit position of the CRC code for 80h (9188h) is inverted to 1189h, which is stored into the CRCD register in the 3rd cycle.

[b15 1189h b0] CRCD register

(4) Set 23h → [b7 … b0] CRCIN register

Bit position of the CRC code for 80C4h (8250h) is inverted to 0A41h, which is stored into the CRCD register in the 3rd cycle.

[b15 0A41h b0] CRCD register

◯ 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.

```
                                              1000 1000
1 0001 0000 0010 0001 / 1000 0000 0000 0000 0000 0000   <- Data
      ^                 1000 1000 0001 0000 1
Generator               ---------------------
polynomial                   1000 0001 0000 1000 0
                             1000 1000 0001 0000 1
                             ---------------------
                          ->     1001 0001 1000 1000
                    CRC code
```

Modulo-2 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.

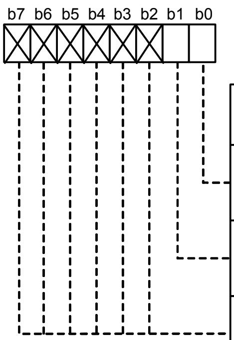
Figure 20.3 CRC Calculation

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.

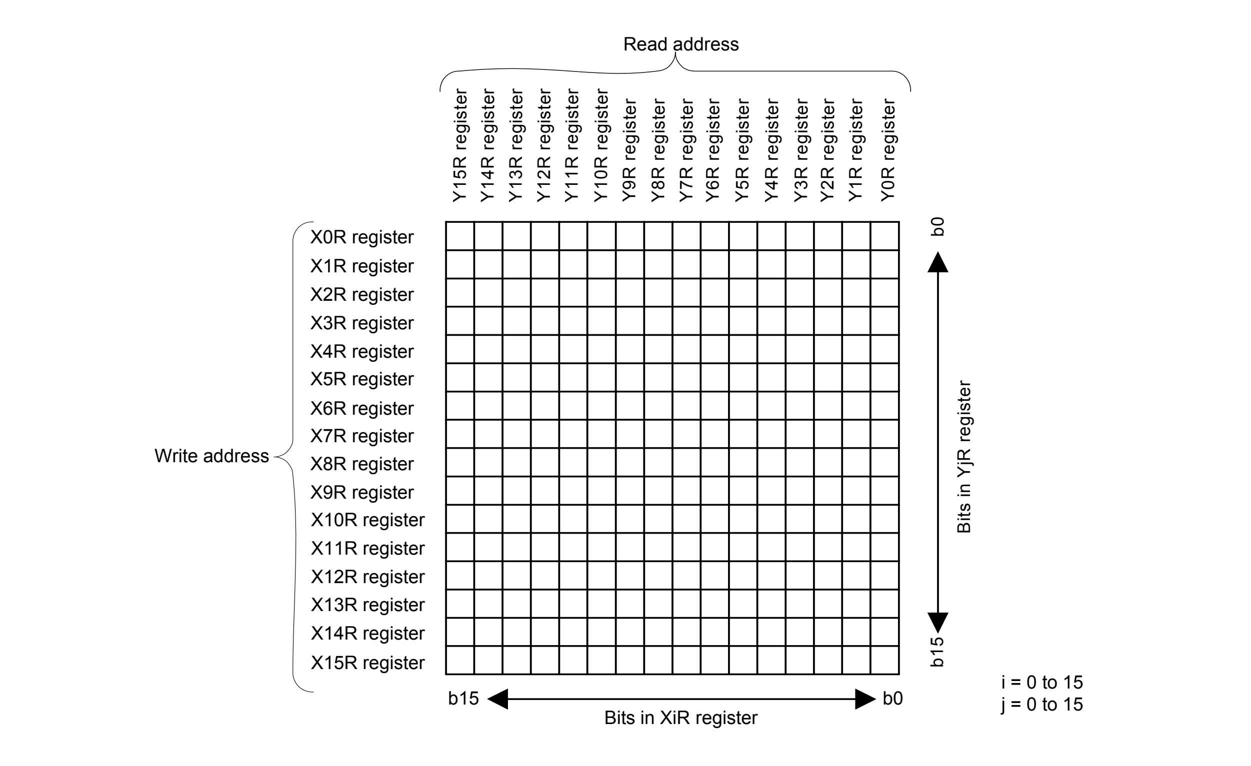


Figure 21.2 Conversion Table when the XYC0 Bit is Set to 0

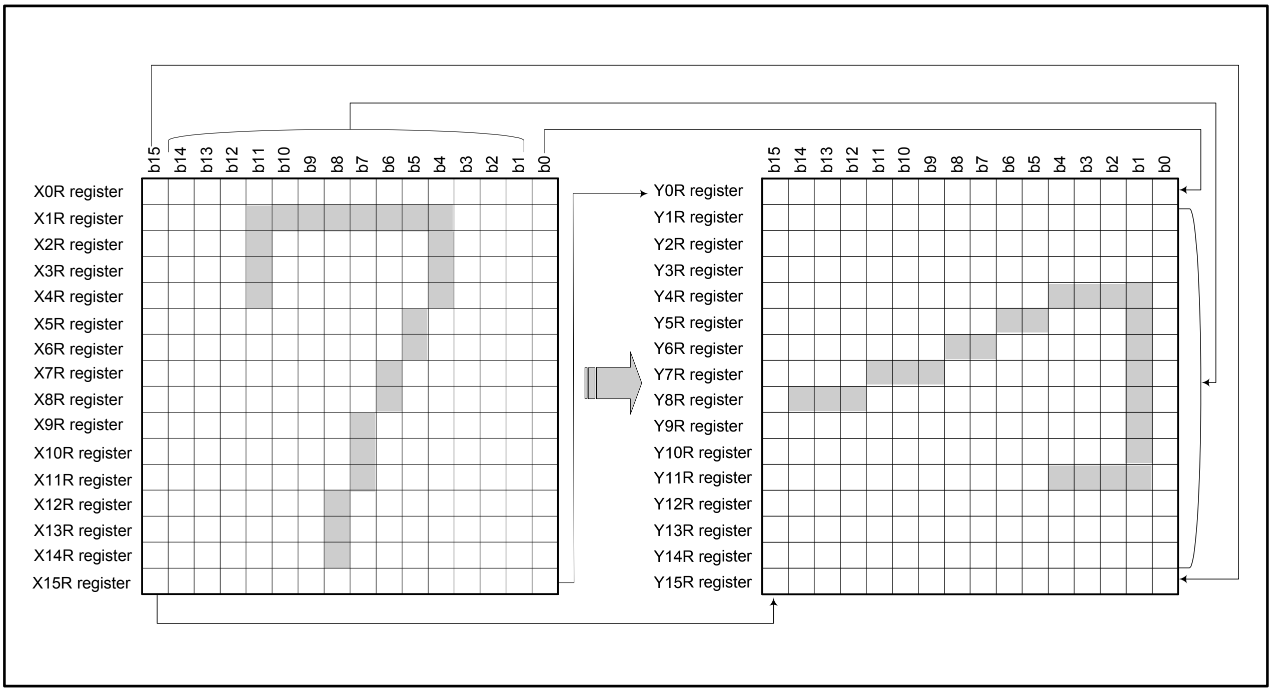


Figure 21.3 X/Y Conversion

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.

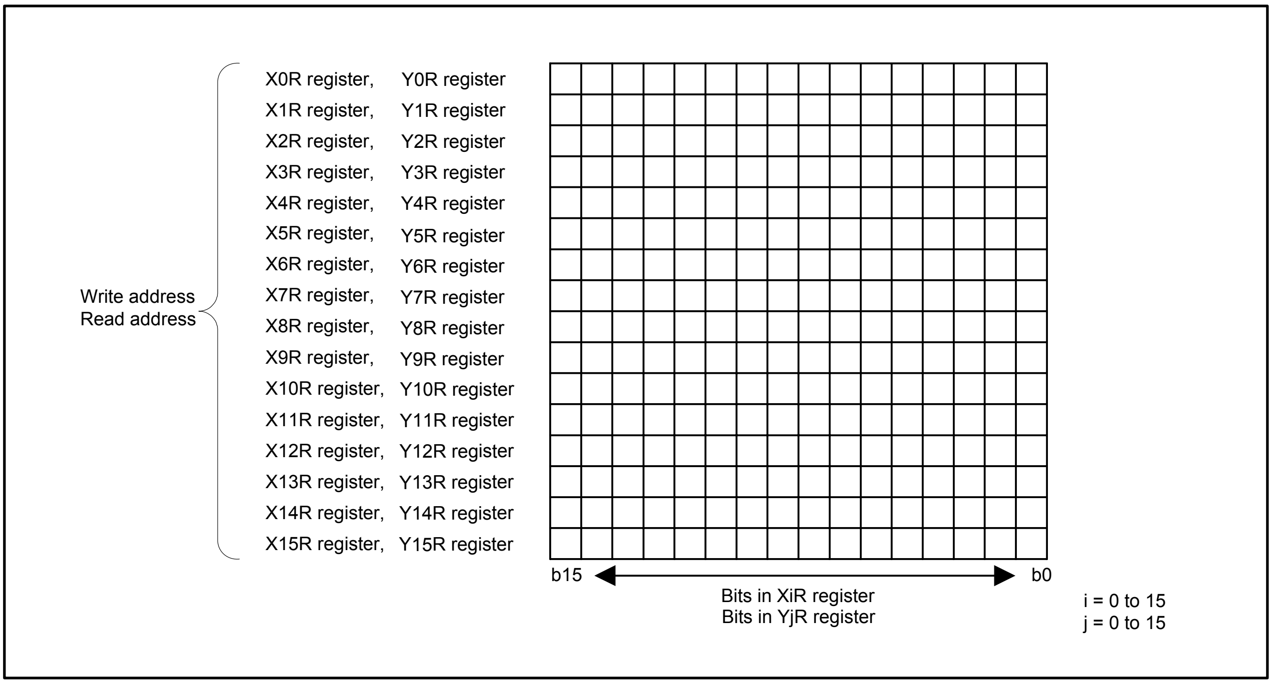


Figure 21.4 Conversion Table when the XYC0 Bit is Set to 1

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.

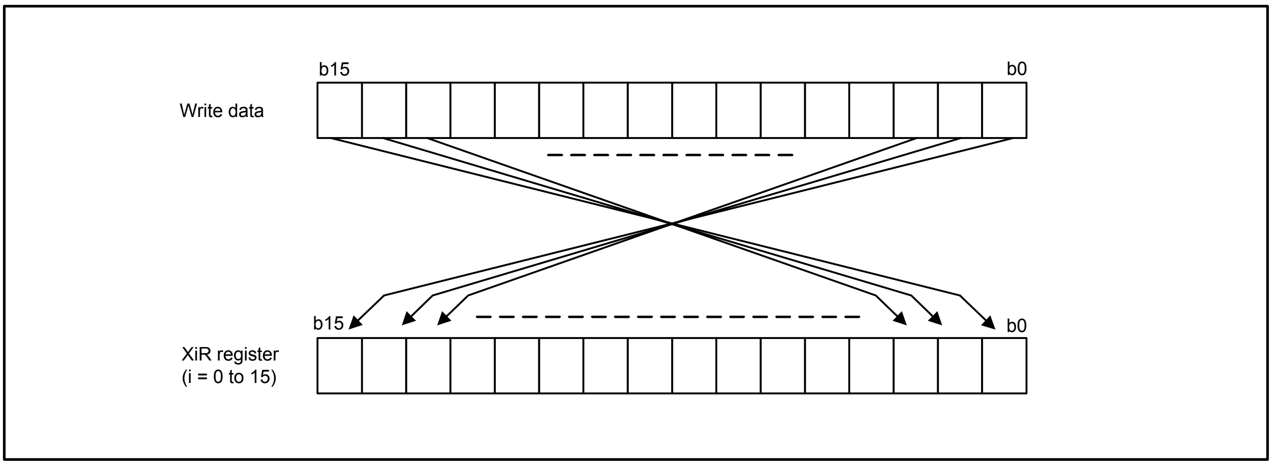


Figure 21.5 Conversion when the XYC1 Bit is Set to 1

22. Intelligent I/O

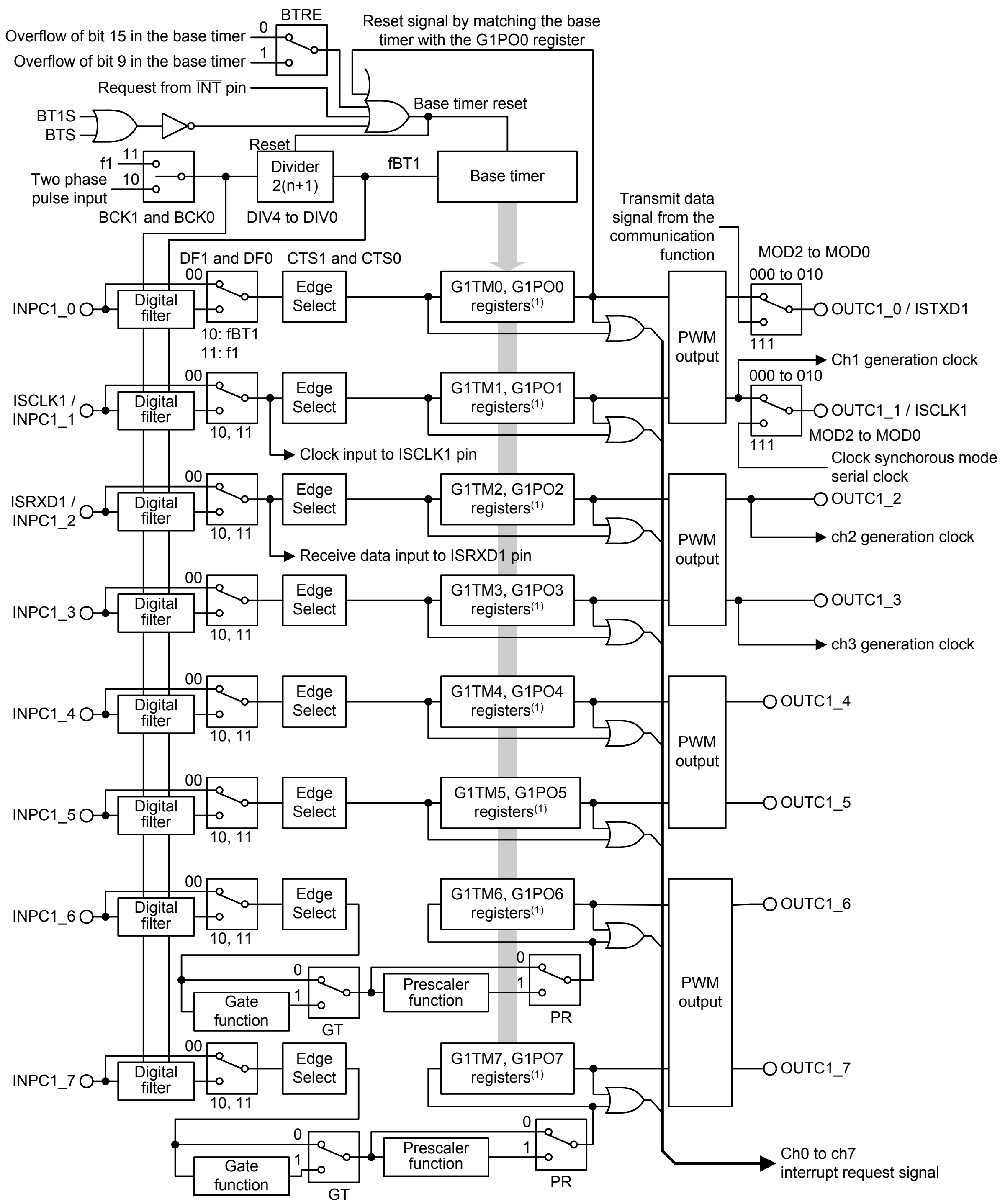
The intelligent I/O is multifunctional I/O ports, which can be used for time measurement function (input capture), waveform generation function (output compare), clock synchronous serial communication, clock asynchronous serial communication (UART), or HDLC data processing.
The intelligent I/O in M32C/87 Group (M32C/87, M32C/87A, M32C/87B) has three groups. Time measurement function or waveform generation function can be selected per channel.
Table 22.1 lists functions and channels of the intelligent I/O.

Table 22.1 Intelligent I/O Functions and Channels

<table>
<tr><th>Function</th><th>Group 0</th><th>Group 1(1)</th><th>Group 2</th></tr>
<tr><td>Base timer</td><td rowspan="2">Not Provided</td><td>1 base timer</td><td>1 base timer</td></tr>
<tr><td>Two-phase pulse signal processing mode</td><td>Provided</td><td>Not Provided</td></tr>
<tr><td>Time measurement function</td><td rowspan="3">Not Provided</td><td>8 channels</td><td rowspan="3">Not Provided</td></tr>
<tr><td>Prescaler function</td><td>2 channels</td></tr>
<tr><td>Gate function</td><td>2 channels</td></tr>
<tr><td>Waveform generation function</td><td rowspan="7">Not Provided</td><td>8 channels</td><td>8 channels(2)</td></tr>
<tr><td>Single-phase waveform output mode</td><td>Provided</td><td>Provided</td></tr>
<tr><td>Phase-delayed waveform output mode</td><td>Provided</td><td>Provided</td></tr>
<tr><td>Set-Reset (SR) waveform output mode</td><td>Provided</td><td>Provided</td></tr>
<tr><td>Bit modulation PWM output mode</td><td rowspan="3">Not Provided</td><td>Provided</td></tr>
<tr><td>Real-time port output mode</td><td>Provided</td></tr>
<tr><td>Parallel real-time output mode</td><td>Provided</td></tr>
<tr><td>Communication function</td><td>1 channel</td><td>1 channel</td><td>1 channel</td></tr>
<tr><td>Data length</td><td>8 bits</td><td>8 bits</td><td>Variable length</td></tr>
<tr><td>Clock synchronous mode</td><td>Provided</td><td>Provided</td><td>Provided</td></tr>
<tr><td>Clock asynchronous mode</td><td>Not Provided</td><td>Provided</td><td>Not Provided</td></tr>
<tr><td>HDLC data processing mode</td><td>Provided</td><td>Provided</td><td>Not Provided</td></tr>
<tr><td>IEBus mode (optional)(3)</td><td>Not Provided</td><td>Not Provided</td><td>Provided</td></tr>
</table>

NOTES:
1. The time measurement function and the waveform generation function can use a total of eight channels per group.
2. 8 channels are available in the 144-pin package. 3 channels are available in 100-pin package.
3. Please contact a Renesas sales office for optional features.

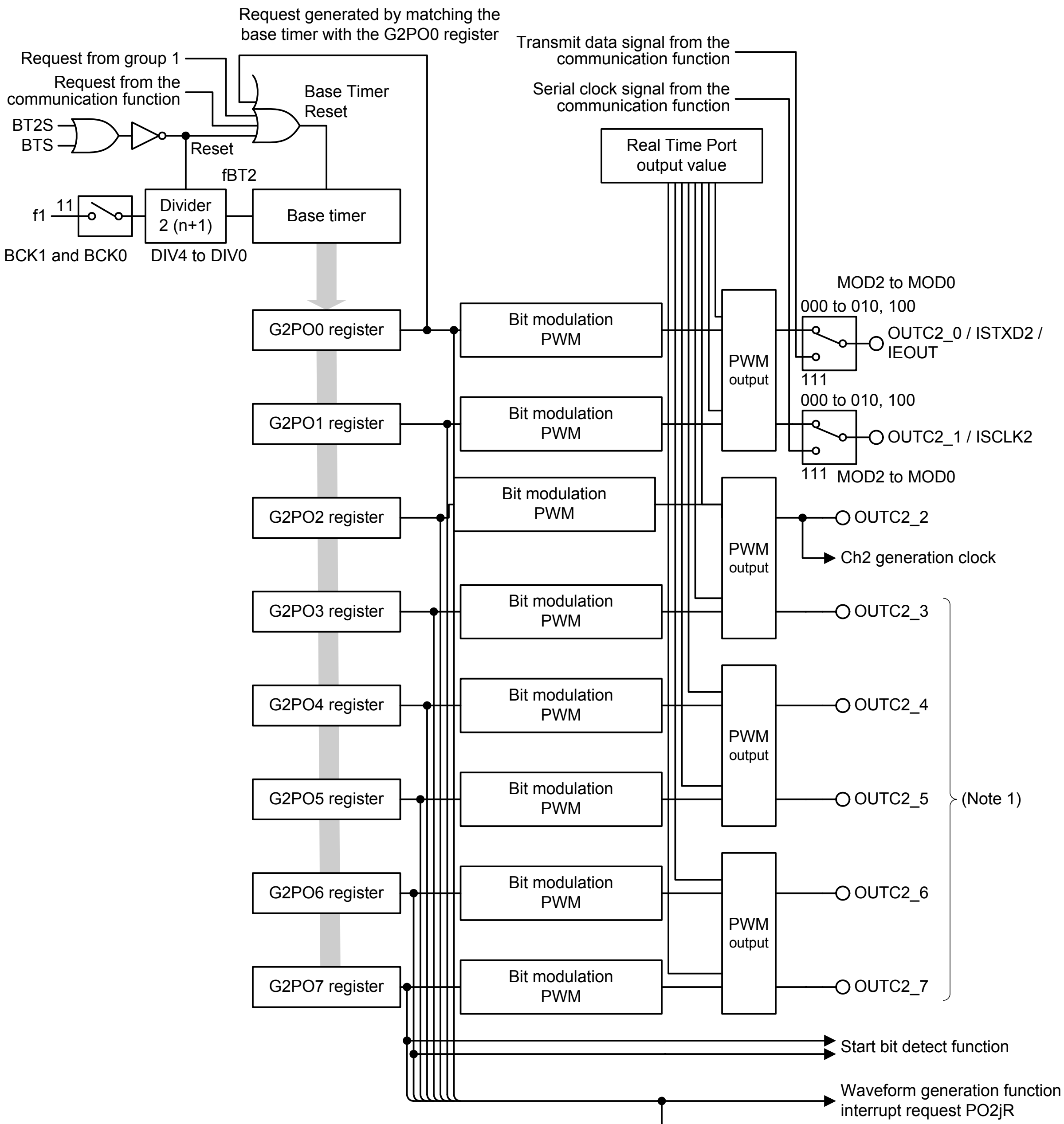
Figure 22.1 shows a block diagram of time measurement and waveform generation functions in group 1. Figure 22.2 shows a block diagram of waveform generation function in group 2.
Figures 22.3 to 22.14 show registers associated with the base timer, time measurement and waveform generation functions. (See figures 22.36, 22.37, and 22.55 for block diagrams of the communication function, and figures 22.38 to 22.46 and 22.56 to 22.60 for registers associated with the communication function.)



BTRE: Bit in the G1POCR0 register
BT1S: Bit in the BTSR register
BCK1 and BCK0, DIV4 to DIV0: Bits in the G1BCR0 register
BTS: Bit in the G1BCR1 register
CTS1 and CTS0, DF1 and DF0, GT, PR: Bits in the G1TMCRj register (j = 0 to 7)
MOD2 to MOD0: Bits in the G1POCRj register

NOTE:
1. After a clock, which is selected in the G1BCR0 register, is supplied to the registers, each register value becomes the after reset value.

Figure 22.1 Time Measurement/Waveform Generation Function in Group 1 Block Diagram



DIV4 to DIV0, BCK1 and BCK0: Bits in the G2BCR0 register
BTS: Bit in the G2BCR1 register
BT2S: Bit in the BTSR register
MOD2 to MOD0: Bits in the G2POCRj register (j = 0 to 7)
PO2jR: Bits in registers IIO3IR, IIO5IR to IIO11IR
BT2R:Bit in the IIO8IR register

NOTES:
1. In the100-pin package, these output function cannot be used.
2. After a clock, which is selected in the G2BCR0 register, is supplied to the registers, each register value becomes the after reset value.

Figure 22.2 Waveform Generation Function in Group 2 Block Diagram

Group 1 Base Timer Register(1)

b15 b8 b7 b0

Symbol	Address	After Reset
G1BT	0121h - 0120h	Undefined

Function	Setting Range	RW
· While the base timer is counting: When read, the base timer value is returned(2). When write, the count continues from the value written. · While the base timer is in reset state: When read, undefined value is returned. No value can be written.	0000h to FFFFh	RW

NOTES:
1. The base timer operates when its count source is selected using bits BCK1 and BCK0 in the G1BCR0 register. When both the BT1S bit in the BTSR register and the BTS bit in the G1BCR1 register are set to 0, the base timer is placed in a reset state and the count value remains 0000h. When either the BT1S bit or the BTS bit is set to 1, the count starts.
2. The G1BT register reflects the value of the base timer with a half fBT1 clock cycle delay.

Group 1 Base Timer Control Register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G1BCR0	0122h	00h

Bit Symbol	Bit Name	Function	RW
BCK0	Count source select bits	b1 b0 0 0: Clock stopped 0 1: Do not set to this value 1 0: Two-phase pulse signal input(1) 1 1: f1	RW
BCK1			RW
DIV0	Count source divide ratio select bits	If setting value is n (n = 0 to 31), the count source is divided by 2(n + 1). No division if n = 31. b6 b5 b4 b3 b2 (n = 0) 0 0 0 0 0: Divide-by-2 (n = 1) 0 0 0 0 1: Divide-by-4 (n = 2) 0 0 0 1 0: Divide-by-6 : (n = 30) 1 1 1 1 0: Divide-by-62 (n = 31) 1 1 1 1 1: No division	RW
DIV1			RW
DIV2			RW
DIV3			RW
DIV4			RW
IT	Base timer interrupt generation timing select bit	0: When bit 15 changed from 1 to 0 1: When bit 14 changed from 1 to 0	RW

NOTE:
1. To set bits BCK1 and BCK0 to 10b (two-phase pulse signal input), set bits UD1 and UD0 in the G1BCR1 register to 10b (two-phase pulse signal processing mode).

Figure 22.3 G1BT Register, G1BCR0 Register

Group 1 Base Timer Control Register 1

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

Symbol	Address	After Reset
G1BCR1	0123h	X000 000Xb

Bit Symbol	Bit Name	Function	RW
– (b0)	Unimplemented. Write 0. Read as undefined value.		–
RST1	Base timer reset source select bit 1	0: Base timer is not reset by matching the G1PO0 register 1: Base timer is reset by matching the G1PO0 register(1)	RW
RST2	Base timer reset source select bit 2	0: Base timer is not reset by applying "L" to the $\overline{INT0}$ or $\overline{INT1}$ pin 1: Base timer is reset by applying "L" to the $\overline{INT0}$ or $\overline{INT1}$ pin(2)	RW
– (b3)	Reserved bit	Set to 0	RW
BTS	Base timer start bit(3)	0: Base timer reset 1: Base timer count starts	RW
UD0	Counter increment/ decrement control bits	b6 b5 0 0: Counter increment mode 0 1: Counter increment/decrement mode 1 0: Two-phase pulse signal processing mode(4) 1 1: Do not set to this value	RW
UD1			RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. The base timer is reset at the second fBT1 clock cycle after the base timer matches the G1PO0 register.
2. The IPSA_0 bit in the IPSA register selects the input pin, either $\overline{INT0}$ or $\overline{INT1}$.
3. Use the BTSR register when multiple base timers start counting simultaneously. In this case, set the BTS bit to 0.
4. In two-phase pulse signal processing mode, the base timer is not reset if the counter is decremented at the second clock cycle after the base timer matches the G1PO0 register, even though the RST1 bit is set to 1.

Figure 22.4 G1BCR1 Register

Group 1 Time Measurement Control Register i (i = 0 to 7)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G1TMCR0 to G1TMCR3	0118h, 0119h, 011Ah, 011Bh	00h
G1TMCR4 to G1TMCR7	011Ch, 011Dh, 011Eh, 011Fh	00h

Bit Symbol	Bit Name	Function	RW
CTS0	Time measurement trigger select bits	b1 b0 0 0: No time measurement 0 1: Rising edge 1 0: Falling edge 1 1: Both edges	RW
CTS1			RW
DF0	Digital filter select bits	b3 b2 0 0: No digital filter 0 1: Do not set to this value 1 0: Use digital filter (use fBT1 as sampling clock) 1 1: Use digital filter (use f1 as sampling clock)	RW
DF1			RW
GT	Gate function select bit[1]	0: Gate function not used 1: Gate function used	RW
GOC	Gate release bit 1[1][2]	0: No trigger input is accepted by matching the base timer and the G1POk register (k = 4, 5) 1: One trigger input is accepted after matching the base timer and the G1POk register	RW
GSC	Gate release bit 2[1][2]	One trigger input is accepted after setting the GSC bit to 1	RW
PR	Prescaler function select bit[1]	0: Prescaler function not used 1: Prescaler function used	RW

NOTES:
1. The gate function (bits GT, GOC, and GSC) and the prescaler function (PR bit) are available in registers G1TMCR6 and G1TMCR7 only. Set each bit 4 to 7 in registers G1TMCR0 to G1TMCR5 to 0.
2. Bits GOC and GSC are enabled only when the GT bit is set to 1.

Group 1 Time Measurement Prescaler Register i (i = 6, 7)

b7 b0

Symbol	Address	After Reset
G1TPR6, G1TPR7	0124h, 0125h	00h

Function	Setting Range	RW
If the setting value is *n*, the time measurement is performed every time a trigger input is counted *n+1* times[1]	00h to FFh	RW

NOTE:
1. After the PR bit in the G1TMCRi register is changed from 0 (prescaler function not used) to 1 (prescaler function used), the first time measurement may be performed when a trigger input is counted n times.

Figure 22.5 G1TMCR0 to G1TMCR7 Registers, G1TPR6 and G1TPR7 Registers

Group 1 Time Measurement Register i (i = 0 to 7)

b15 b8 b7 b0

Symbol	Address	After Reset
G1TM0 to G1TM2	0101h - 0100h, 0103h - 0102h, 0105h - 0104h	Undefined
G1TM3 to G1TM5	0107h - 0106h, 0109h - 0108h, 010Bh - 010Ah	Undefined
G1TM6, G1TM7	010Dh - 010Ch, 010Fh - 010Eh	Undefined

Function	Setting Range	RW
The base timer value is stored every time measurement is performed	–	RO

Group 1 Waveform Generation Control Register i (i = 0 to 7)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G1POCR0	0110h	0000 X000b
G1POCR1 to G1POCR3	0111h, 0112h, 0113h	0X00 X000b
G1POCR4 to G1POCR7	0114h, 0115h, 0116h, 0117h	0X00 X000b

Bit Symbol	Bit Name	Function	RW
MOD0	Operating mode select bits	b2 b1 b0 0 0 0: Single waveform output mode 0 0 1: SR waveform output mode(1) 0 1 0: Phase-delayed waveform output mode 0 1 1: Do not set to this value 1 0 0: Do not set to this value 1 1 0: Do not set to this value(2) 1 1 1: Use communication function output(3)	RW
MOD1			RW
MOD2			RW
– (b3)	Unimplemented. Write 0. Read as undefined value.		–
IVL	Output level select bit(5)	0: "L" output 1: "H" output	RW
RLD	G1POi register value reload timing select bit	0: Reload when written 1: Reload when the base timer is reset	RW
BTRE	Base timer reset timing select bit(4)	0: Base timer is reset when the bit 15 overflows 1: Base timer is reset when the bit 9 overflows(6)	RW
INV	Inverted output function select bit(5)	0: Output not inverted 1: Output inverted	RW

NOTES:
1. SR waveform output mode is enabled only in even channels. In SR waveform output mode, the setting for the corresponding odd channel (the channel followed by the even channel) is ignored. SR waveform can be output from even channels, and not from odd channels.
2. To perform the UART receive operation in group 1, set the G1POCR2 register to 0000 0110b.
3. To use the ISTXD1 pin, set bits MOD2 to MOD0 in the G1POCR0 register to 111b. To use the ISCLK1 pin as output, set bits MOD2 to MOD0 in the G1POCR1 register to 111b. Do not set bits MOD2 to MOD0 in registers G1POCR2 to G1POCR7 to 111b.
4. The BTRE bit is available only in the G1POCR0 register. Set the bit 6 in registers G1POCR1 to G1POCR7 to 0.
5. If the INV or IVL bit is written while outputting waveform, the value written takes effect immediately on the output waveform.
6. When the BTRE bit is set to 1, set bits BCK1 and BCK0 in the G1BCR0 register to 11b (f1), and bits UD1 and UD0 in the G1BCR1 register to 00b (counter increment mode).

Figure 22.6 G1TM0 to G1TM7 Registers, G1POCR0 to G1POCR7 Registers

Group 1 Waveform Generation Register i (i = 0 to 7)

b15 b8 b7 b0

Symbol	Address	After Reset
G1PO0 to G1PO2	0101h - 0100h, 0103h - 0102h, 0105h - 0104h	Undefined
G1PO3 to G1PO5	0107h - 0106h, 0109h - 0108h, 010Bh - 010Ah	Undefined
G1PO6, G1PO7	010Dh - 010Ch, 010Fh - 010Eh	Undefined

Function	Setting Range	RW
When the G1POi register is read, the value written is returned. When a value is written to the G1POi register: - If the RLD bit in the G1POCRi register is set to 0, the value written is immediately reloaded into the internal register and reflected in such as output waveform. - If the RLD bit in the G1POCRi register is set to 1, the value written is reloaded into the internal register when the base timer is reset.	0000h to FFFFh	RW

Figure 22.7 G1PO0 to G1PO7 Registers

Group 1 Function Select Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G1FS	0127h	00h

Bit Symbol	Bit Name	Function	RW
FSC0	Channel 0 time measurement/waveform generation function select bit	0: Waveform generation function selected 1: Time measurement function selected	RW
FSC1	Channel 1 time measurement/waveform generation function select bit		RW
FSC2	Channel 2 time measurement/waveform generation function select bit		RW
FSC3	Channel 3 time measurement/waveform generation function select bit		RW
FSC4	Channel 4 time measurement/waveform generation function select bit		RW
FSC5	Channel 5 time measurement/waveform generation function select bit		RW
FSC6	Channel 6 time measurement/waveform generation function select bit		RW
FSC7	Channel 7 time measurement/waveform generation function select bit		RW

Group 1 Function Enable Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G1FE	0126h	00h

Bit Symbol	Bit Name	Function	RW
IFE0	Channel 0 function enable bit	0: Channel i's function disabled (i = 0 to 7) 1: Channel i's function enabled	RW
IFE1	Channel 1 function enable bit		RW
IFE2	Channel 2 function enable bit		RW
IFE3	Channel 3 function enable bit		RW
IFE4	Channel 4 function enable bit		RW
IFE5	Channel 5 function enable bit		RW
IFE6	Channel 6 function enable bit		RW
IFE7	Channel 7 function enable bit		RW

Figure 22.8 G1FS Register, G1FE Register

Group 2 Base Timer Register[1]

b15 b8 b7 b0

Symbol: G2BT
Address: 0161h - 0160h
After Reset: Undefined

Function	Setting Range	RW
· While the base timer is counting: When read, the base timer value is returned[2]. When write, the count continues from the value written. · While the base timer is in reset state: When read, undefined value is returned. No value can be written.	0000h to FFFFh	RW

NOTES:
1. The base timer operates when its count source is selected using bits BCK1 and BCK0 in the G2BCR0 register. When both the BT2S bit in the BTSR register and the BTS bit in the G2BCR1 register are set to 0, the base timer is placed in a reset state and the count value remains 0000h. When either the BT2S or the BTS bit is set to 1, the count starts.
2. The G2BT register reflects the value of the base timer with a half fBT2 clock cycle delay.

Group 2 Base Timer Control Register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: G2BCR0
Address: 0162h
After Reset: 00h

Bit Symbol	Bit Name	Function	RW
BCK0	Count source select bits	b1 b0 0 0: Clock stopped 0 1: Do not set to this value 1 0: Do not set to this value 1 1: f1	RW
BCK1			RW
DIV0	Count source divide ratio select bits	If setting value is n (n = 0 to 31), the count source is divided by 2(n + 1). No division if n = 31. b6 b5 b4 b3 b2 (n = 0) 0 0 0 0 0: Divide-by-2 (n = 1) 0 0 0 0 1: Divide-by-4 (n = 2) 0 0 0 1 0: Divide-by-6 : (n = 30) 1 1 1 1 0: Divide-by-62 (n = 31) 1 1 1 1 1: No division	RW
DIV1			RW
DIV2			RW
DIV3			RW
DIV4			RW
IT	Base timer interrupt generation timing select bit	0: When bit 15 is changed from 1 to 0 1: When bit 14 is changed from 1 to 0	RW

Figure 22.9 G2BT Register, G2BCR0 Register

Group 2 Base Timer Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
	0	0		0			

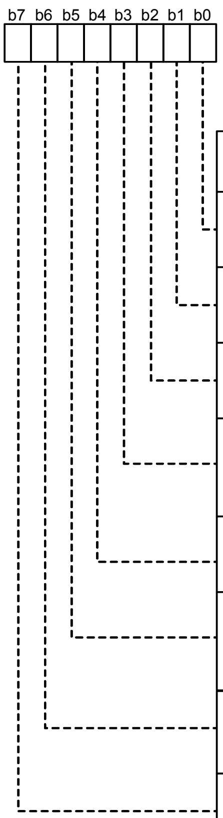
Symbol	Address	After Reset
G2BCR1	0163h	00h

Bit Symbol	Bit Name	Function	RW
RST0	Base timer reset source select bit 0	0: Base timer is not reset when the base timer in group 1 is reset. 1: Base timer is reset when the base timer in group 1 is reset.	RW
RST1	Base timer reset source select bit 1	0: Base timer is not reset by matching the G2PO0 register 1: Base timer is reset by matching the G2PO0 register[1]	RW
RST2	Base timer reset source select bit 2	0: Base timer is not reset by a reset request from the communication function 1: Base timer is reset by a reset request from the communication function	RW
– (b3)	Reserved bit	Set to 0	RW
BTS	Base timer start bit[3]	0: Base timer reset 1: Base timer count starts	RW
– (b6-b5)	Reserved bits	Set to 0	RW
PRP	Parallel real-time port function select bit[2]	0: Real-time port output mode 1: Parallel read-time port output mode	RW

NOTES:
1. The base timer is reset at the second fBT1 clock cycle after the base timer matches the G2PO0 register.
2. The PRP bit is enabled when the RTP bit in the G2POCRi register is set to 1 (real-time port function used).
3. Use the BTSR register when multiple base timers start counting simultaneously. In this case, set the BTS bit to 0.

Figure 22.10 G2BCR1 Register

Group 2 Waveform Generation Control Register i (i = 0 to 7)



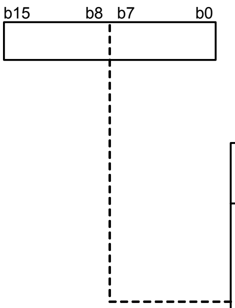
Symbol	Address	After Reset
G2POCR0 to G2POCR3	0150h, 0151h, 0152h, 0153h	00h
G2POCR4 to G2POCR7	0154h, 0155h, 0156h, 0157h	00h

Bit Symbol	Bit Name	Function	RW
MOD0	Operating mode select bits[3]	b2 b1 b0 0 0 0: Single waveform output mode 0 0 1: SR waveform output mode[1] 0 1 0: Phase-delayed waveform output mode 0 1 1: Do not set to this value 1 0 0: Bit modulation PWM output mode 1 0 1: Do not set to this value 1 1 0: Do not set to this value 1 1 1: Use communication function output[2]	RW
MOD1			RW
MOD2			RW
PRT	Parallel real-time port output trigger select bit[4]	0: Signal output when base timer matches the G2POi register is not used as a trigger 1: Signal output when base timer matches the G2POi register is used as a trigger	RW
IVL	Output level select bit[6]	0: "L" output 1: "H" output	RW
RLD	G2POi register value reload timing select bit	0: Reload when written 1: Reload when the base timer is reset	RW
RTP	Real-time port function select bit[3][4]	0: Not used 1: Used (real-time port output mode or parallel real-time port output mode)	RW
INV	Inverted output function select bit[5][6]	0: Output not inverted 1: Output inverted	RW

NOTES:
1. SR waveform output mode is enabled only in even channels. In SR waveform output mode, the setting for the corresponding odd channel (the channel followed by the even channel) is ignored. SR waveform can be output from even channels, and not from odd channels.
2. To use the ISTXD2 pin or IEOUT pin as output, set bits MOD2 to MOD0 in the G2POCR0 register to 111b. To use the ISCLK2 pin as output, set bits MOD2 to MOD0 in the G2POCR1 register to 111b. Do not set bits MOD2 to MOD0 in registers G2POCR2 to G2POCR7 to 111b.
3. When the RTP bit is set to 1, set bits MOD2 to MOD0 to 000b.
4. Real-time port output and parallel real-time port output cannot be used in the same group. To use parallel real-time port output, set the RTP bit to 1 and the PRT bit to 1 in the channel used for parallel real-time port output. Also, set the PRP bit in the G2BCR1 register to 1.
5. When the RTP bit is set to 1, the INV bit setting is disabled.
6. If the INV or IVL bit is written while outputting waveform, the value written takes effect immediately on the output waveform.

Figure 22.11 G2POCR0 to G2POCR7 Registers

Group 2 Waveform Generation Register i (i = 0 to 7)



Symbol	Address	After Reset
G2PO0 to G2PO2	0141h - 0140h, 0143h - 0142h, 0145h - 0144h	Undefined
G2PO3 to G2PO5	0147h - 0146h, 0149h - 0148h, 014Bh - 014Ah	Undefined
G2PO6, G2PO7	014Dh - 014Ch, 014Fh - 014Eh	Undefined

Function	Setting Range	RW
When the G2POi register is read, the value written is returned. When a value is written to the G2POi register: - If the RLD bit in the G2POCRi register is set to 0, the value written is immediately reloaded into the internal register and reflected in such as output waveform. - If the RLD bit in the G2POCRi register is set to 1, the value written is reloaded into the internal register when the base timer is reset.	0000h to FFFFh	RW

Figure 22.12 G2PO0 to G2PO7 Register

Group 2 Function Select Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G2RTP	0167h	00h

Bit Symbol	Bit Name	Function	RW
RTP0	Channel 0 RTP output buffer	0: “L” output 1: “H” output	RW
RTP1	Channel 1 RTP output buffer		RW
RTP2	Channel 2 RTP output buffer		RW
RTP3	Channel 3 RTP output buffer		RW
RTP4	Channel 4 RTP output buffer		RW
RTP5	Channel 5 RTP output buffer		RW
RTP6	Channel 6 RTP output buffer		RW
RTP7	Channel 7 RTP output buffer		RW

Group 2 Function Enable Register

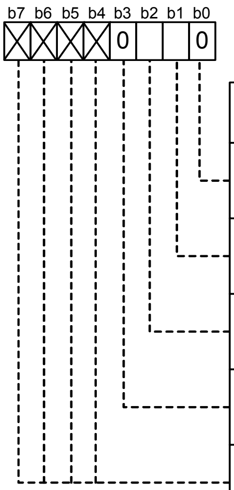
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G2FE	0166h	00h

Bit Symbol	Bit Name	Function	RW
IFE0	Channel 0 function enable bit	0: Channel i's function disabled (i = 0 to 7) 1: Channel i's function enabled	RW
IFE1	Channel 1 function enable bit		RW
IFE2	Channel 2 function enable bit		RW
IFE3	Channel 3 function enable bit		RW
IFE4	Channel 4 function enable bit		RW
IFE5	Channel 5 function enable bit		RW
IFE6	Channel 6 function enable bit		RW
IFE7	Channel 7 function enable bit		RW

Figure 22.13 G2RTP Register, G2FE Register

Base Timer Start Register(1)(2)(3)



Symbol	Address	After Reset
BTSR	0164h	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
– (b0)	Reserved bit	Set to 0	RW
BT1S	Group 1 base timer start bit	0: Base timer reset 1: Base timer count starts	RW
BT2S	Group 2 base timer start bit	0: Base timer reset 1: Base timer count starts	RW
– (b3)	Reserved bit	Set to 0	RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. To use the intelligent I/O, follow the procedure below in the initial configuration.
   (1) Set the G2BCR0 register to supply the clock to the group 2 base timer.
   (2) Set all the BTiS bits (i = 1, 2) to 0 (base timer reset).
   (3) Set the other registers associated with the intelligent I/O.
   The BTiS bits are used to start the base timers in group 1 and group 2 simultaneously. To start each base timer independently, set the BTiS bits to 0 and use the BTS bit in the GiBCR1 register.
2. To start the base timers in group 1 and group 2 simultaneously, set as follows.
   - Set bits BCK1 and BCK0, and bits DIV4 to DIV0 in the GiBCR0 register to the same value in group 1 and group 2.
   - If bits BCK1 and BCK0 or bits DIV4 to DIV0 are changed, set the BTiS bits to 1 twice using the following procedure.
     (1) Set the BTiS bits to 1 (base timer count starts).
     (2) Wait for one or more fBTi clock cycles, and then set the BTiS bits to 0 (base timer reset).
     (3) Wait another one or more fBTi clock cycles, and then set the BTiS bits to 1.
3. The BTSR register is enabled after setting the G2BCR0 register.

Figure 22.14 BTSR Register

22.1 Base Timer

The base timer, a 16-bit free running counter, is available in group 1 and group 2.
Registers in group 1 and group 2 are initialized and written using the base timer clock (fBT) selected in the GiBCR0 register (i = 1, 2). The BTSR register is initialized and written using the base timer clock in group 2. Ensure to select the base timer clock in the G2BCR0 register to initialize the BTSR register; otherwise the BTSR register value remains undefined and the base timer in group 1 may start counting unintentionally.
The base timer counts an internally generated count source continuously.
Tables 22.2 and 22.3 list specifications of the base timer. Figure 22.15 shows a block diagram of the base timer. Figure 22.16 shows a base timer operation example in counter increment mode. Figure 22.17 shows a base timer operation example in count increment/decrement mode.

Table 22.2 Base Timer Specifications (Group 1)

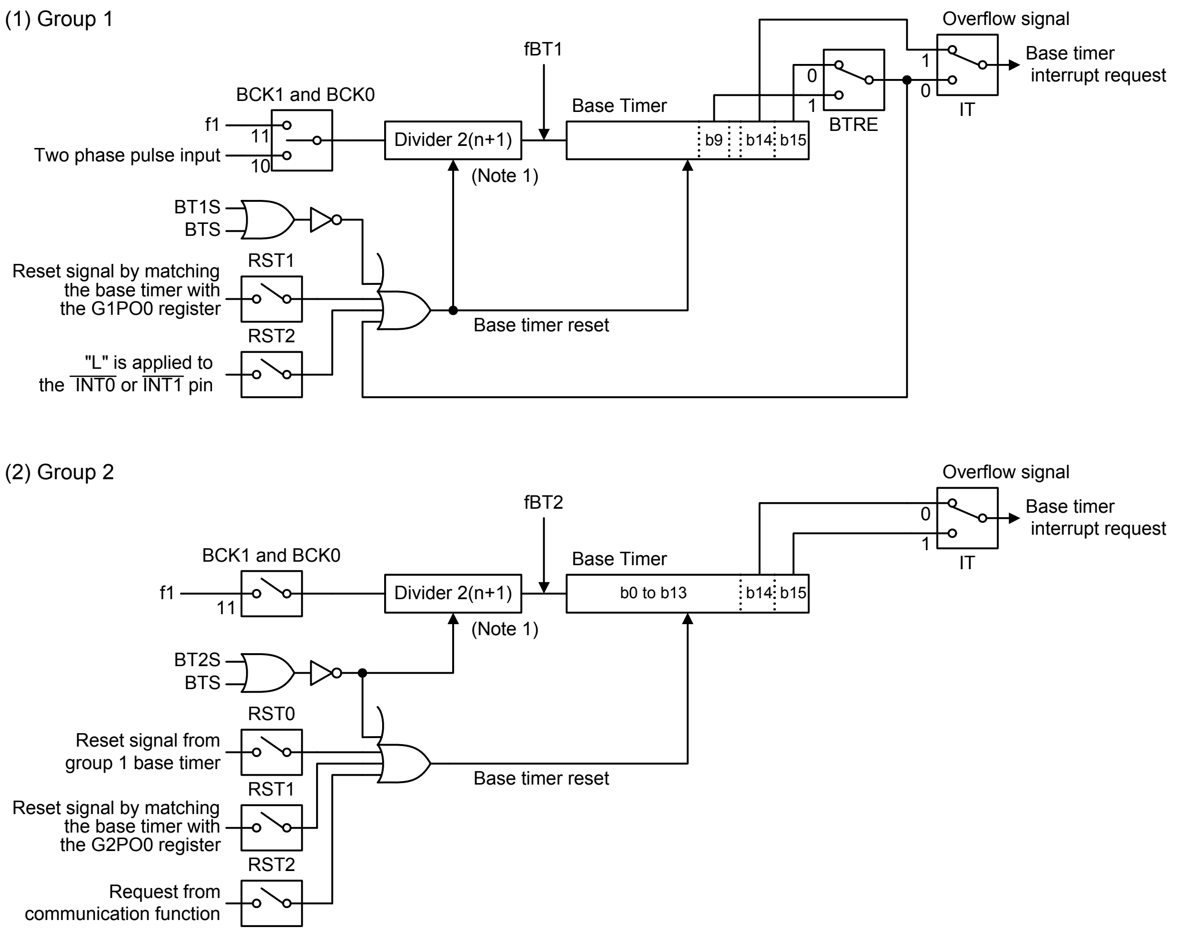
Item	Specification
Count source (fBT1)	• f1 divided by 2(n+1) • Two-phase pulse input divided by 2(n+1) n: determined by bits DIV4 to DIV0 in the G1BCR0 register (n = 0 to 31); no division when n = 31
Count operation	• Counter increments • Counter both increments and decrements • Two-phase pulse signal processing
Count start condition	• When the base timers in groups 1 and 2 start counting independently: Set the BTS bit in the G1BCR1 register to 1 (base timer count starts) •When the base timers in groups 1 and 2 start counting simultaneously: Set bits BT2S and BT1S in the BTSR register to 11b (base timer count starts)
Count stop condition	Base timer count stops when both of the following conditions are met: • The BT1S bit in the BTSR register is set to 0 (base timer reset) • The BTS bit in the G1BCR1 register to 0 (base timer reset)
Base timer reset condition	• The base timer value matches the G1PO0 register value(1) • Bit 15 of the base timer overflows • Bit 9 of the base timer overflows • A low-level (“L”) signal is input to the $\overline{INT0}$ or $\overline{INT1}$ pin
Value when the base timer is in reset state	0000h
Interrupt request generation timing	When bit 9, 14, or 15 of the base timer is changed from 1 to 0 The BT1R bit in the IIO4IR register becomes 1 (interrupt requested) when the interrupt request is generated.
Read from base timer	• Count value is returned when reading the G1BT register while the base timer is counting • Undefined value is returned when reading the G1BT register while the base timer is in reset
Write to base timer	• When a value is written while the base timer is counting, the count continues from the value written • No value can be written while base timer is in reset state
Selectable function	Counter increment/decrement mode • The base timer starts incrementing when the BTS bit is set to 1. When the count reaches FFFFh, the base timer decrements. • If the RST1 bit in the G1BCR1 register is set to 1 (base timer is reset by matching the G1PO0 register), the base timer decrements at the third clock cycle after the base timer value matches the G1PO0 register. Then, the base timer increments again when the count reaches 0000h. Two-phase pulse processing mode • Count two-phase pulse signals from pins P8_0 and P8_1, or pins P7_6 and P7_7. Pins are selectable using the IPSA_0 bit in the IPSA register.

NOTE:
1. When bits RST2 and RST1 in the G1BCR1 register are set to 01b (base timer is reset by matching the G1PO0 register), the setting range of the G1PO0 register must be 0001h to FFFDh.

Table 22.3 Base Timer Specifications (Group 2)

Item	Specification
Count source (fBT2)	• f1 divided by 2(n+1) n: determined by bits DIV4 to DIV0 in the G2BCR0 register (n = 0 to 31); no division when n = 31
Count operation	• Counter increments
Count start condition	• When the base timers in groups 1 and 2 start counting independently: Set the BTS bit in the G2BCR1 register to 1 (base timer count starts) • When the base timers in groups 1 and 2 start counting simultaneously: Set bits BT2S and BT1S in the BTSR register to 11b (base timer count starts)
Count stop condition	Base timer count stops when both of the following conditions are met: • The BT2S bit in the BTSR register is set to 0 (base timer reset) • The BTS bit in the G2BCR1 register to 0 (base timer reset)
Base timer reset condition	• The base timer value matches the G2PO0 register value(1) • Bit 15 of the base timer overflows • When the base timer in group 1 is reset • Reset request from the communication function
Value when the base timer is in reset state	0000h
Interrupt request generation timing	When bit 14 or 15 of the base timer is changed from 1 to 0 The BT2R bit in the IIO8IR register becomes 1 (interrupt requested) when the interrupt request is generated.
Read from base timer	• Count value is returned when reading the G2BT register while the base timer is counting • Undefined value is returned when reading the G2BT register while the base timer is in reset state
Write to base timer	• When a value is written while the base timer is counting, the count continues from the value written • No value can be written while base timer is in reset

NOTE:
1. When bits RST2 and RST1 in the G2BCR1 register are set to 01b (base timer is reset by matching the G2PO0 register), the setting range of the G2PO0 register must be 0001h to FFFDh.



i = 1, 2

BCK1 and BCK0, IT: Bits in the GiBCR0 register
RST1, RST2, BTS: Bits in the GiBCR1 register
RST0: Bit in the G2BCR1 register
BTRE: Bit in the G1POCR0 register
BT1S, BT2S: Bits in the BTSR register

NOTE:
1. The divider is reset when both the BTiS and BTS bits are set to 0.

Figure 22.15 Base Timer Block Diagram

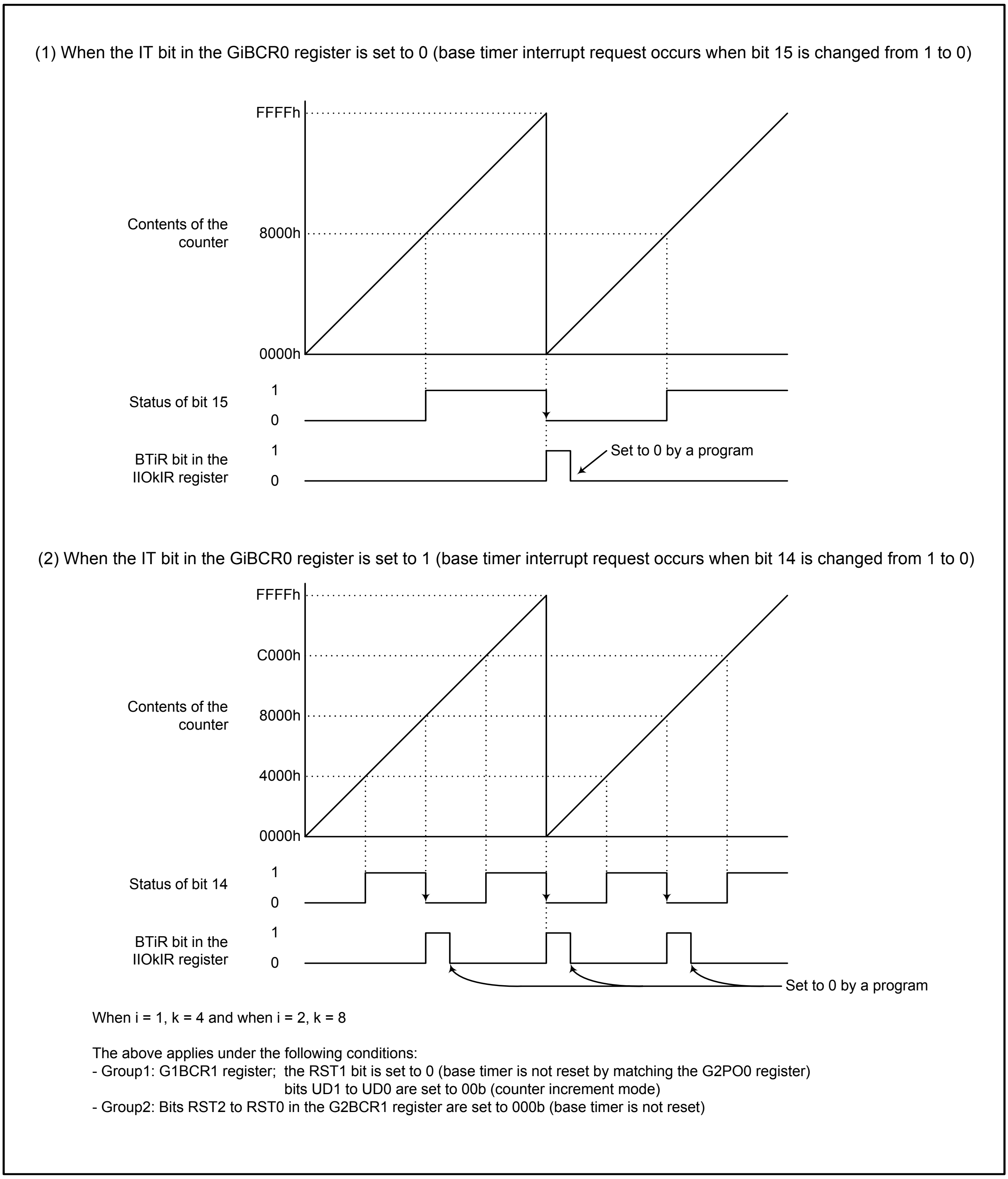
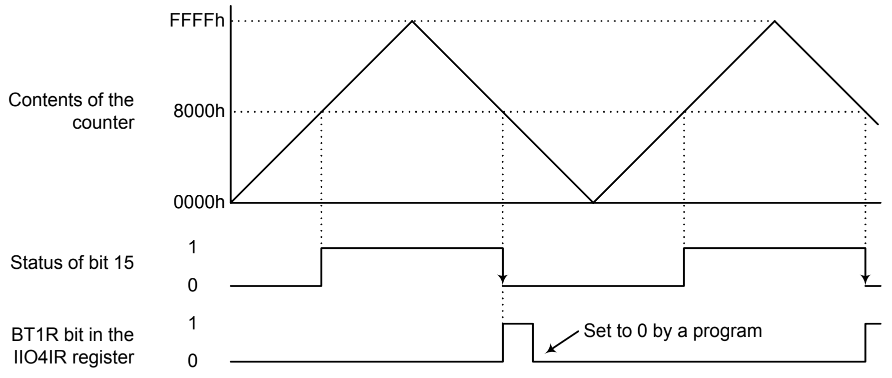


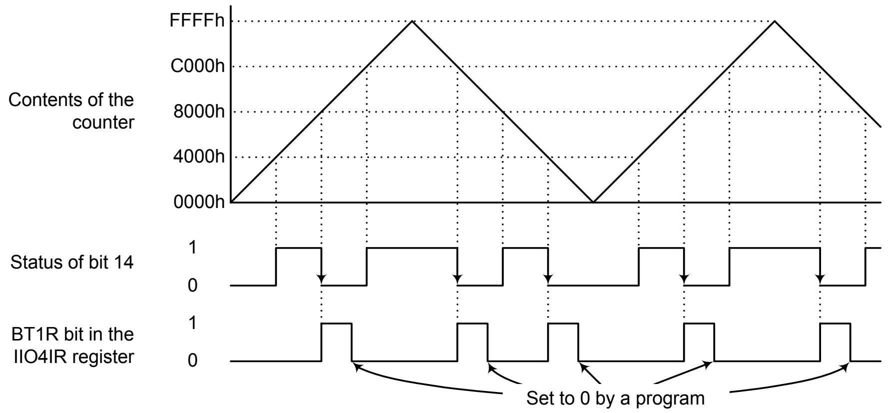
Figure 22.16 Base Timer Operation in Counter Increment Mode (Group 1 and 2)

(1) When the IT bit in the G1BCR0 register is set to 0 (base timer interrupt request occurs when bit 15 is changed from 1 to 0)



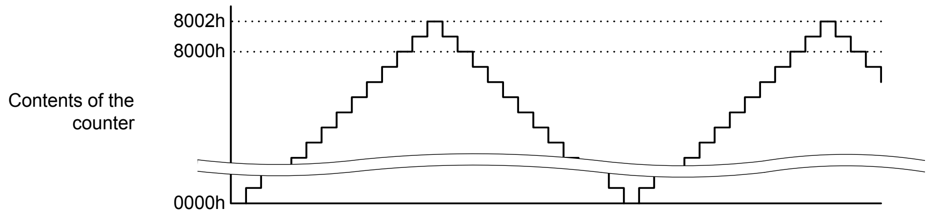
The above applies under the following conditions:
- G1BCR1 register: the RST1 bit is set to 0 (Base timer is not reset by matching the base timer and the G1PO0 register)
bits UD1 and UD0 are set to 01b (counter increment/decrement mode)

(2) When the IT bit in the G1BCR0 register is set to 1 (base timer interrupt request occurs when bit 14 is changed from 1 to 0)



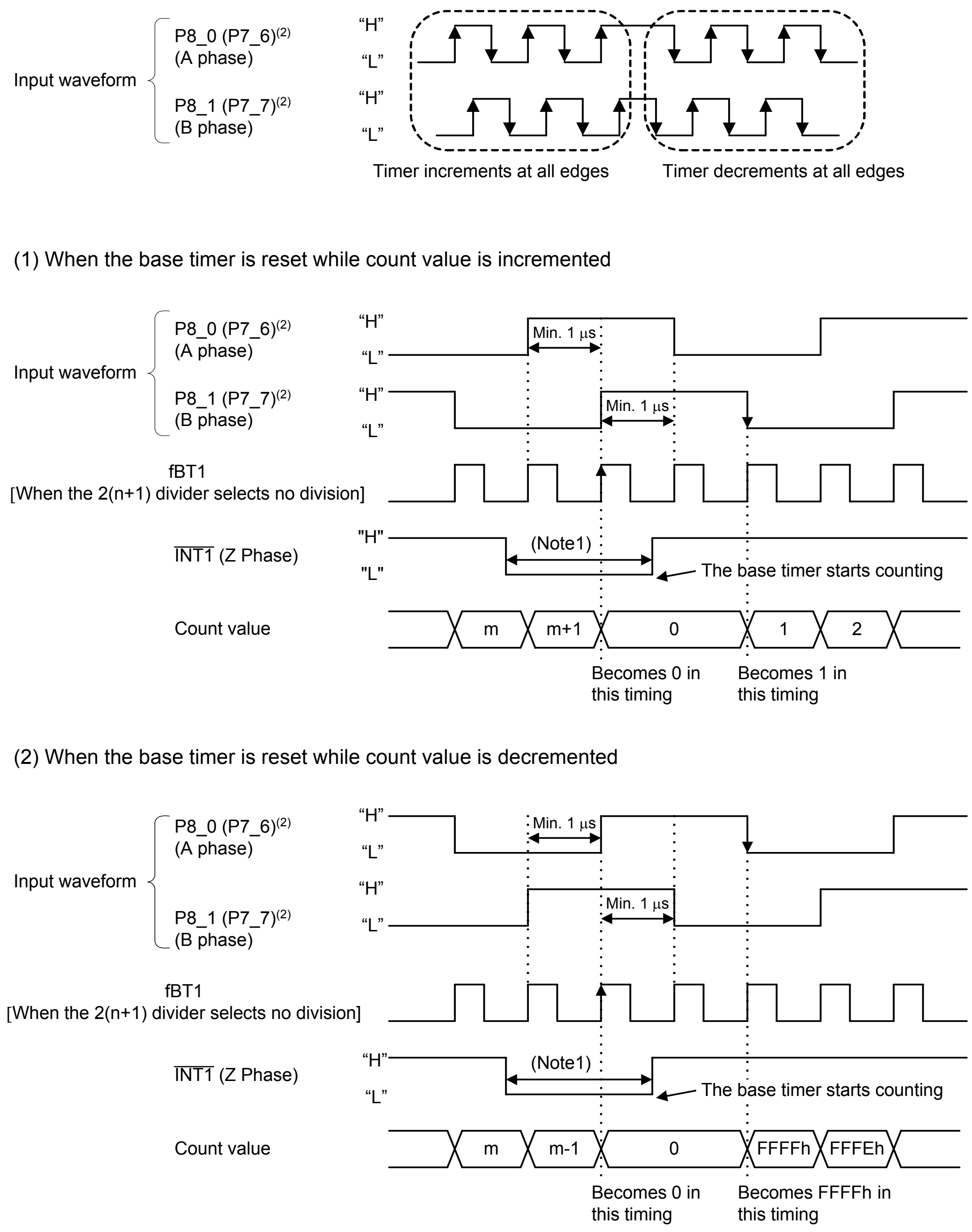
The above applies under the following conditions:
- G1BCR1 register: the RST1 bit is set to 0 (Base timer is not reset by matching the base timer and the G1PO0 register)
bits UD1 and UD0 are set to 01b (counter increment/decrement mode)

(3) When the RST1 bit in the G1BCR1 register is set to 1
(Base timer is reset by matching the base timer and the G1PO0 register)



The above applies under the following conditions:
- The G1PO0 register is set to 8000h
- Bits UD1 and UD0 in the G1BCR1 register are set to 01b (counter increment/decrement mode)

Figure 22.17 Base Timer Operation in Count Increment/Decrement Mode (Group 1)



NOTES:
1. The width requires 1.5 or more fBT1 clock cycles.
2. The IPSA_0 bit in the IPSA register determines two-phase pulse input pins, whether P8_0 and P8_1 or P7_6 and P7_7.

Figure 22.18 Base Timer Operation in Two-Phase Pulse Signal Processing Mode (Group 1)

22.2 Time Measurement Function (Input Capture)

When the external trigger is input, the base timer value is stored into the G1TMi register (i = 0 to 7). The time measurement function is available in group 1. Table 22.4 shows specifications of the time measurement function. Table 22.5 lists pin settings for the time measurement function. Figure 22.19 shows register settings. Figure 22.20 shows an example of time measurement function operation.

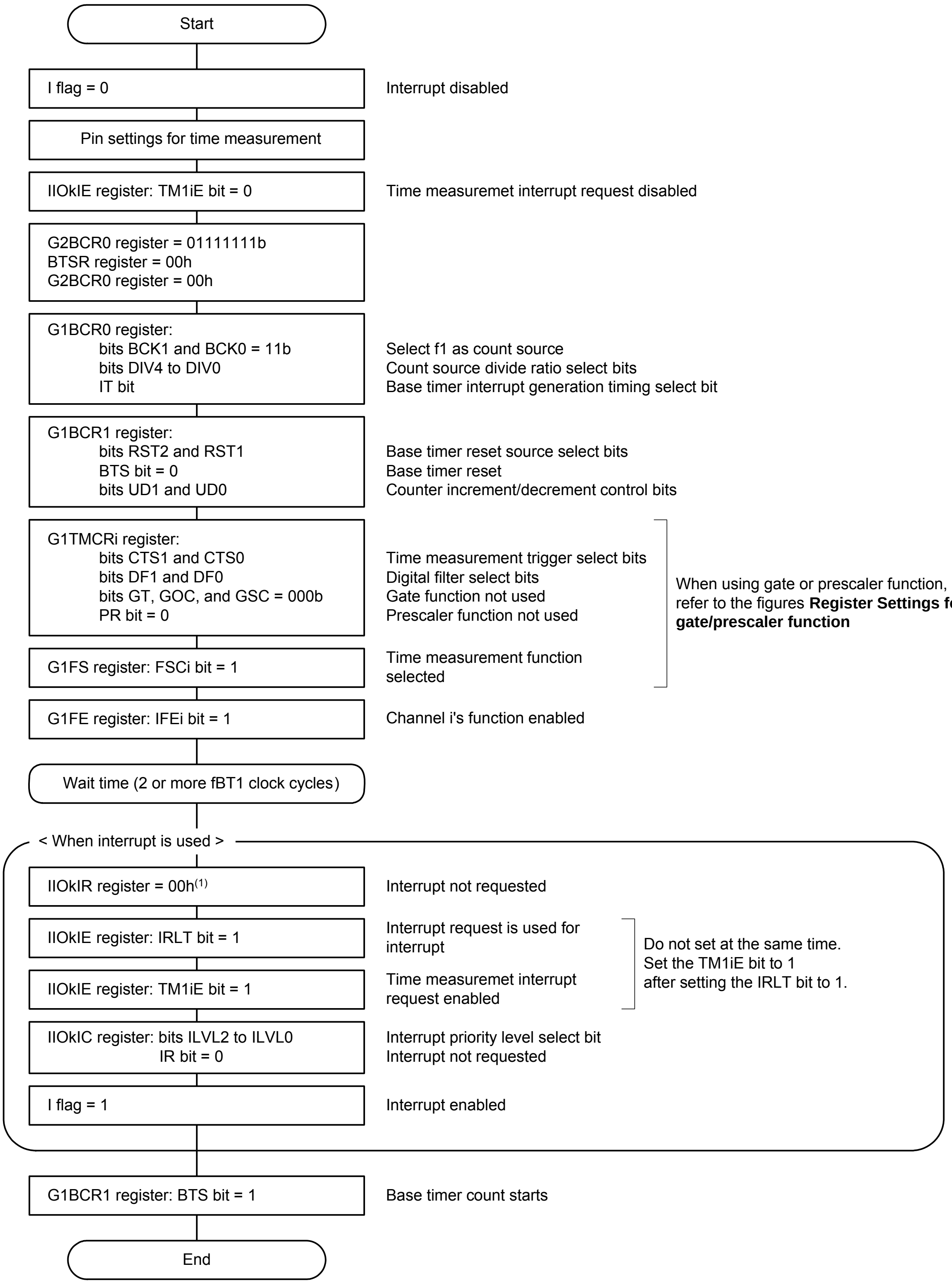
Table 22.4 Time Measurement Function Specifications

Item	Specification
Measurement channel	Group 1: Channels 0 to 7
INPC1_ i pin (i = 0 to 7)	Trigger input
Trigger input polarity	Selectable among rising edge, falling edge, or both edges
Measurement start condition	Time measurement starts when all of the following conditions are met: • Base timer count starts • Set the FSCi bit in the G1FS register to 1 (time measurement function selected) • Set the IFEi bit in the G1FE register to 1 (channel i's function enabled)
Measurement stop condition	Time measurement stops when any of the following conditions is met: • Set the IFEi bit to 0 (channel i's function disabled) • Base timer count stops (function in all channels disabled)
Time measurement timing	• Without prescaler: every time a valid edge is input • With prescaler (channels 6 and 7): every (*G1TPRj register value + 1*) times a valid edge is input (*j = 6, 7*)
Interrupt request generation timing	At the time measurement timing The TM1iR bit in the IIOkIR register (k = 0 to 4, 8 to 10) becomes 1 (interrupt requested) when an interrupt request is generated. (See **Figure 11.18** IIO0IR to IIO11IR Registers)
Selectable function	• Digital filter function The digital filter samples a trigger input signal level using f1 or fBT1 and passes the pulse that have matched its signal level three times • Prescaler function (channels 6 and 7) Time measurement is performed every (*G1TPRj register value + 1*) times a trigger is input • Gate function (channels 6 and 7) After a time measurement is performed by the first trigger input, the subsequent trigger inputs are all ignored. Thereafter, one trigger input is accepted when either of the following conditions is met: - Base timer value matches the G1POn register value (n = 4, 5) - Set the GSC bit in the G1TMCRj register to 1

Table 22.5 Pin Settings for Time Measurement Function

Port	Function	Bit Setting		
		IPS Register	PD7, PD8, PD11, PD14 Registers	PS1, PS2, PS5, PS8 Registers
P7_0	INPC1_6	IPS1 = 0	PD7_0 = 0	PS1_0 = 0
P7_1	INPC1_7		PD7_1 = 0	PS1_1 = 0
P7_3	INPC1_0		PD7_3 = 0	PS1_3 = 0
P7_4	INPC1_1		PD7_4 = 0	PS1_4 = 0
P7_5	INPC1_2		PD7_5 = 0	PS1_5 = 0
P7_6	INPC1_3		PD7_6 = 0	PS1_6 = 0
P7_7	INPC1_4		PD7_7 = 0	PS1_7 = 0
P8_1	INPC1_5		PD8_1 = 0	PS2_1 = 0
P11_0(1)	INPC1_0	IPS1 = 1	PD11_0 = 0	PS5_0 = 0
P11_1(1)	INPC1_1		PD11_1 = 0	PS5_1 = 0
P11_2(1)	INPC1_2		PD11_2 = 0	PS5_2 = 0
P11_3(1)	INPC1_3		PD11_3 = 0	PS5_3 = 0
P14_0(1)	INPC1_4		PD14_0 = 0	PS8_0 = 0
P14_1(1)	INPC1_5		PD14_1 = 0	PS8_1 = 0
P14_2(1)	INPC1_6		PD14_2 = 0	PS8_2 = 0
P14_3(1)	INPC1_7		PD14_3 = 0	PS8_3 = 0

NOTE:
1. This port is provided in the 144-pin package only.

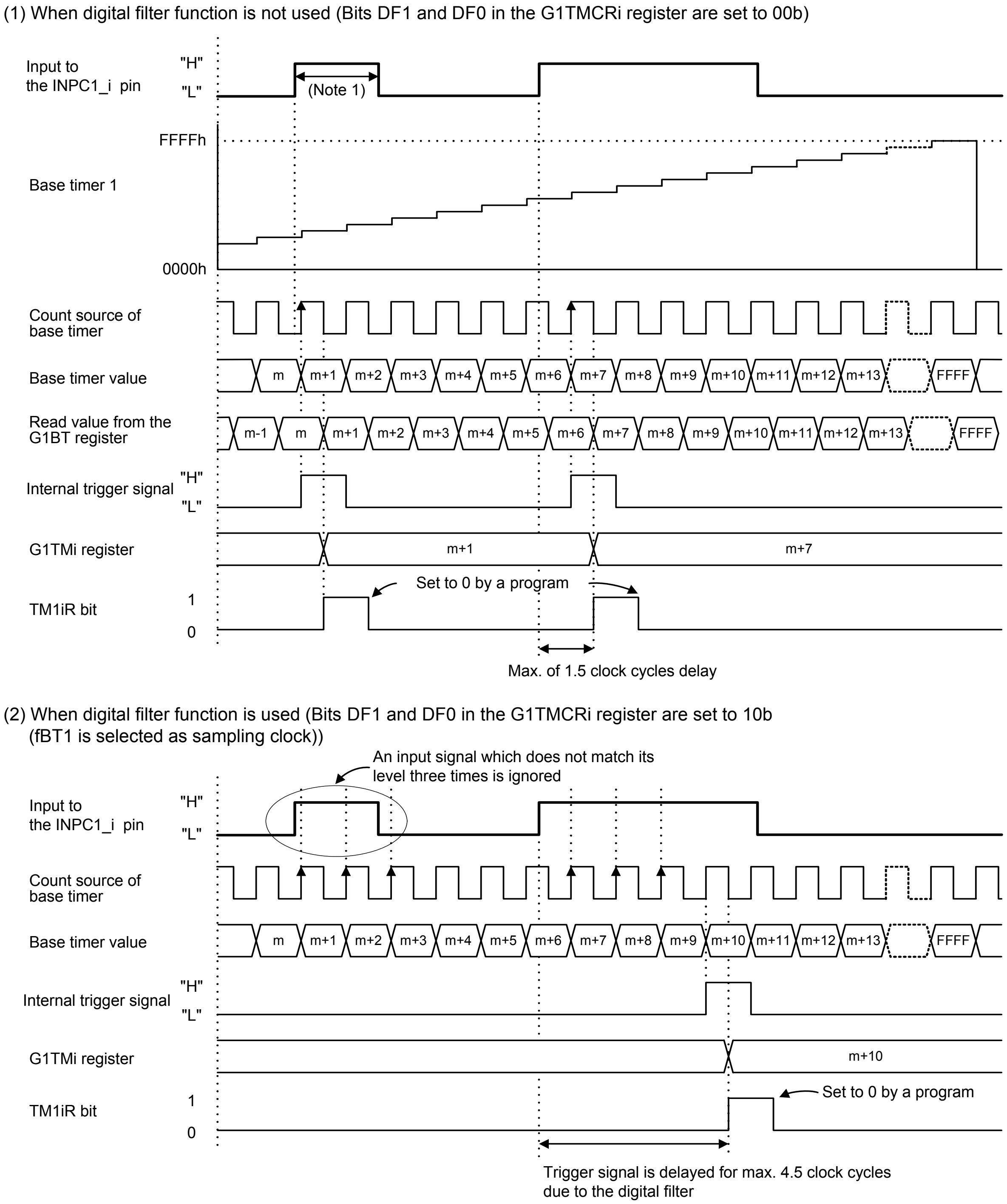


i = 0 to 7; k = 0 to 4, 8 to 10

NOTE:

1. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated in the same register (Interrupt does not occur).

Figure 22.19 Register Settings for Time Measurement Function



i = 0 to 7
The TM1iR bit: bit in registers IIO0IR to IIO4IR and IIO8IR to IIO10IR
The above applies under the following conditions:
- G1TMCRi register: bits CTS1 and CTS0 are set to 01b (rising edge is selected for time measurement trigger)
the PR bit is set to 0 (prescaler function is not used)
the GT bit is set to 0 (gate function is not used)
- G1BCR1 register: bits RST2 and RST1 are set to 00b (base timer is not reset)
bits UD1 and UD0 are set to 00b (counter increment mode)

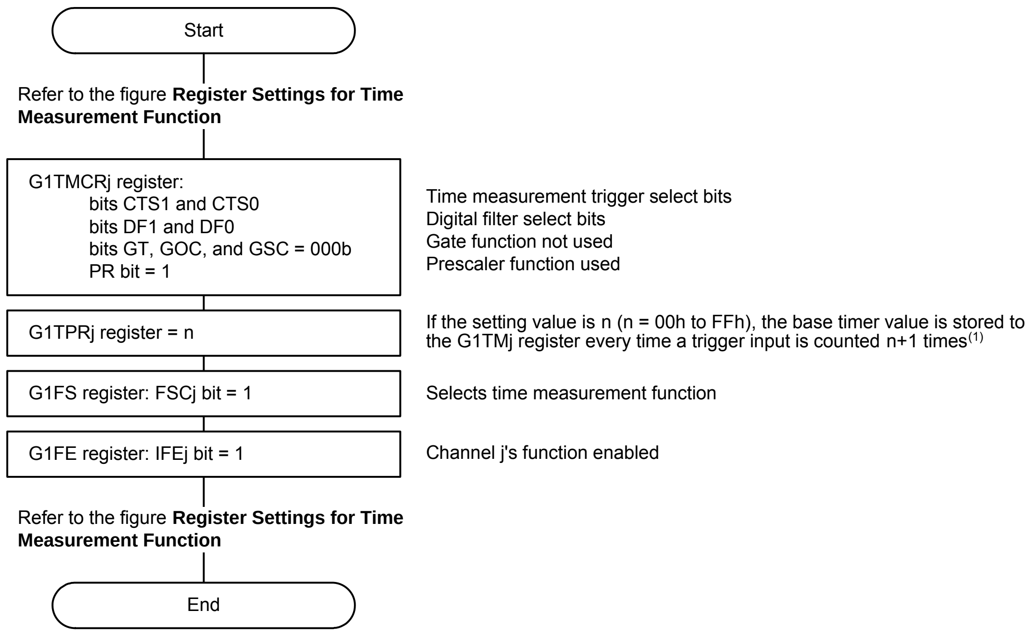
NOTE:
1. The width of pulse input to INPC1_i pin requires 1.5 or more fBT1 clock cycles.

Figure 22.20 Time Measurement Function Operation

22.2.1 Prescaler Function

With the prescaler function, a time measurement is performed every (G1TPRj register value + 1) times a trigger is input. The prescaler function is available in channel 6 and channel 7 in group 1.

Figure 22.21 shows register settings. Figure 22.22 shows an example of prescaler function operation.

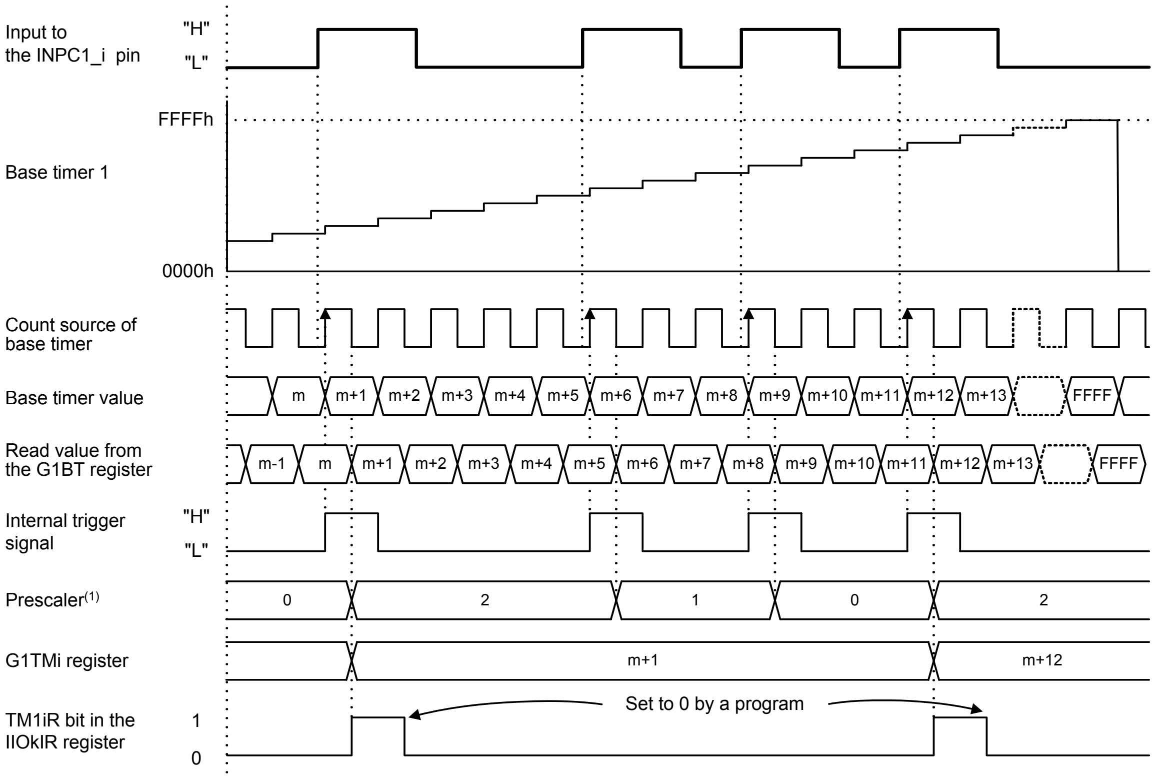


j = 6, 7

NOTE:

1. After the PR bit in the G1TMCRj register is changed from 0 (prescaler function not used) to 1 (prescaler function used), the first time measurement may be performed when a trigger input is counted n times.

Figure 22.21 Register Settings for Prescaler Function



When i = 6, k = 10 and when i = 7, k = 4

The above applies under the following conditions:
- The G1TPRi register is set to 02h
- G1TMCRi register; bits CTS1 and CTS0 are set to 01b (rising edge is selected for time measurement trigger) the PR bit is set to 1 (prescaler function used)

NOTE:
1. This applies to the 2nd or later prescaler cycle after the PR bit is set to 1.

Figure 22.22 Prescaler Function Operation

22.2.2 Gate Function

With the gate function, trigger inputs are ignored for a specific period of time. After a time measurement is performed by the first trigger input, the subsequent trigger inputs are all ignored. Thereafter, one trigger input is accepted every time either of the following conditions is met:

- Base timer value matches the G1POk register value (k = 4, 5) (Waveform generation function is used).
  The G1PO4 register is used to control the gate function in channel 6.
  The G1PO5 register is used to control the gate function in channel 7.
- Set the GSC bit in the G1TMCRj register to 1. (j = 6, 7)

The gate function is available in channel 6 and channel 7.
Figure 22.23 shows register settings. Figure 22.24 shows an example of gate function operation.

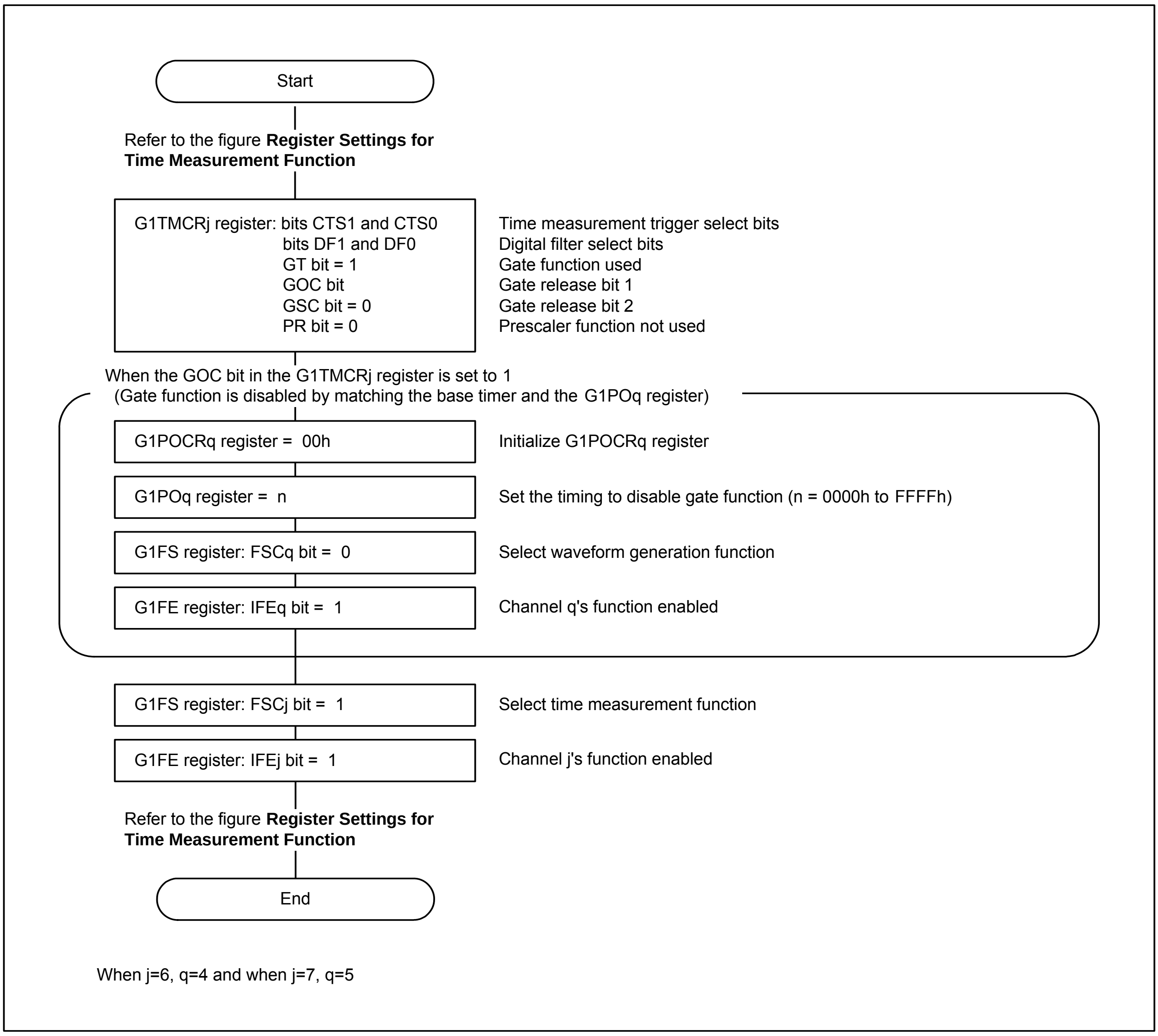
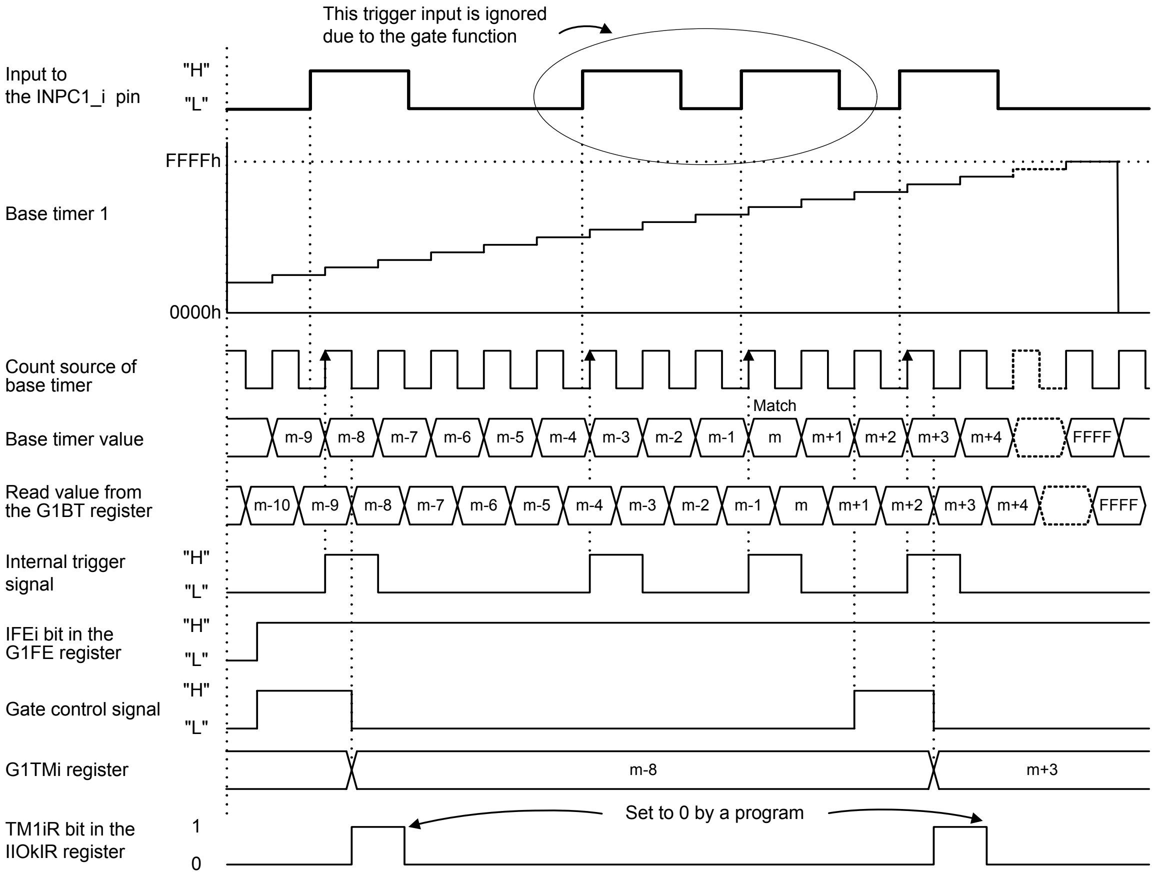


Figure 22.23 Register Settings for Gate Function



When i = 6, k = 10 and q = 4
When i = 7, k = 4 and q = 5

m: Setting value of the G1POq register (m = 0000h to FFFFh)

The above applies under the following conditions:
- G1TMCRi register: bits CTS1 and CTS0 are set to 01b (rising edge is selected for time measurement trigger)
  the GT bit is set to 1 (gate function used)
  the GOC bit is set to 1 (gate function is disabled by matching the base timer and the G1POq register)

Figure 22.24 Gate Function Operation

22.3 Waveform Generation Function (Output Compare)

Waveform generation function outputs a pulse when the base timer value matches the GiPOj register (i = 1, 2; j = 0 to 7). Group 1 and group 2 have waveform generation function.

The waveform generation function has the following six modes:

- Single-phase waveform output mode (Group 1 and group 2)
- Phase-delayed waveform output mode (Group 1 and group 2)
- Set/reset (SR) waveform output mode (Group 1 and group 2)
- Bit modulation PWM output mode (Group 2)
- Real-time port output mode (Group 2)
- Parallel real-time port output mode (Group 2)

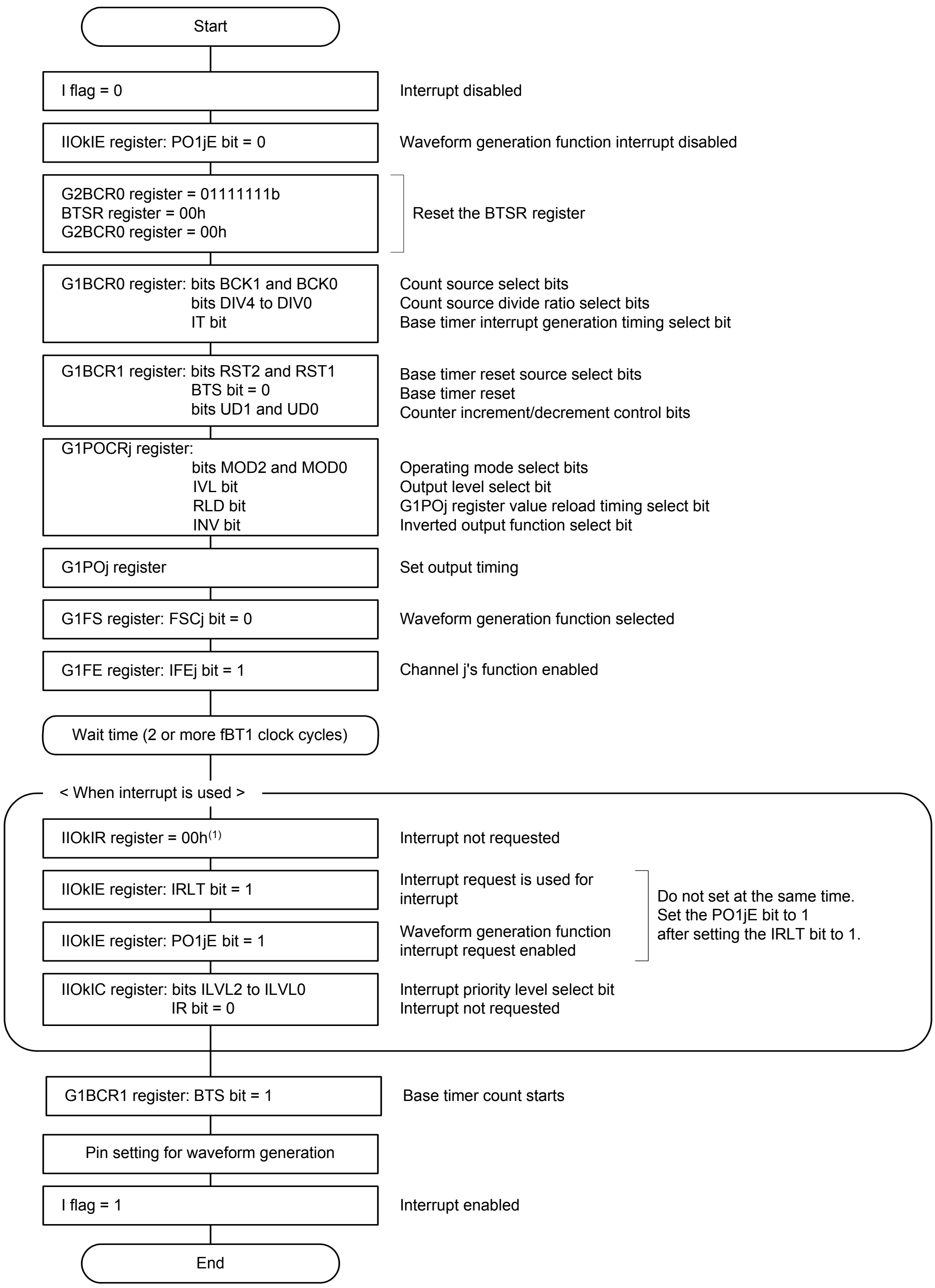
Table 22.6 lists pin settings for the waveform generating function. Figures 22.25 and 22.26 show register settings.

Table 22.6 Pin Settings for Waveform Generation Function

Port	Function	Bit Setting				
		PSE1 Register	PSD1 Register	PSC, PSC2 Registers	PSL0 to PSL3, PSL5, PSL7 Registers	PS0 to PS3, PS5, PS7, PS8 Registers(1)(4)
P6_4	OUTC2_1	–	–	–	PSL0_4 = 1	PS0_4 = 1
P7_0(3)	OUTC1_6	PSE1_0 = 0	PSD1_0 = 1	PSC_0 = 1	PSL1_0 = 0	PS1_0 = 1
P7_0(3)	OUTC2_0	–	PSD1_0 = 0	PSC_0 = 1	PSL1_0 = 0	PS1_0 = 1
P7_1(3)	OUTC1_7	PSE1_1 = 0	PSD1_1 = 1	PSC_1 = 1	PSL1_1 = 0	PS1_1 = 1
P7_1(3)	OUTC2_2	–	PSD1_1 = 0	PSC_1 = 1	PSL1_1 = 0	PS1_1 = 1
P7_3	OUTC1_0	–	–	PSC_3 = 1	PSL1_3 = 0	PS1_3 = 1
P7_4	OUTC1_1	–	PSD1_4 = 0	PSC_4 = 1	PSL1_4 = 0	PS1_4 = 1
P7_5	OUTC1_2	–	–	PSC_5 = 0	PSL1_5 = 1	PS1_5 = 1
P7_6	OUTC1_3	PSE1_6 = 0	PSD1_6 = 1	PSC_6 = 0	PSL1_6 = 0	PS1_6 = 1
P7_7	OUTC1_4	–	PSD1_7 = 0	–	PSL1_7 = 1	PS1_7 = 1
P8_1	OUTC1_5	–	PSD2_1 = 0	PSC2_1 = 1	PSL2_1 = 1	PS2_1 = 1
P9_2	OUTC2_0	–	–	–	PSL3_2 = 1	PS3_2 = 1
P11_0(2)	OUTC1_0	–	–	–	PSL5_0 = 0	PS5_0 = 1
P11_1(2)	OUTC1_1	–	–	–	PSL5_1 = 0	PS5_1 = 1
P11_2(2)	OUTC1_2	–	–	–	PSL5_2 = 0	PS5_2 = 1
P11_3(2)	OUTC1_3	–	–	–	PSL5_3 = 0	PS5_3 = 1
P13_0(2)	OUTC2_4	–	–	–	PSL7_0 = 0	PS7_0 = 1
P13_1(2)	OUTC2_5	–	–	–	PSL7_1 = 0	PS7_1 = 1
P13_2(2)	OUTC2_6	–	–	–	PSL7_2 = 0	PS7_2 = 1
P13_3(2)	OUTC2_3	–	–	–	PSL7_3 = 0	PS7_3 = 1
P13_4(2)	OUTC2_0	–	–	–	PSL7_4 = 0	PS7_4 = 1
P13_5(2)	OUTC2_2	–	–	–	PSL7_5 = 0	PS7_5 = 1
P13_6(2)	OUTC2_1	–	–	–	PSL7_6 = 0	PS7_6 = 1
P13_7(2)	OUTC2_7	–	–	–	PSL7_7 = 0	PS7_7 = 1
P14_0(2)	OUTC1_4	–	–	–	–	PS8_0 = 1
P14_1(2)	OUTC1_5	–	–	–	–	PS8_1 = 1
P14_2(2)	OUTC1_6	–	–	–	–	PS8_2 = 1
P14_3(2)	OUTC1_7	–	–	–	–	PS8_3 = 1

NOTES:

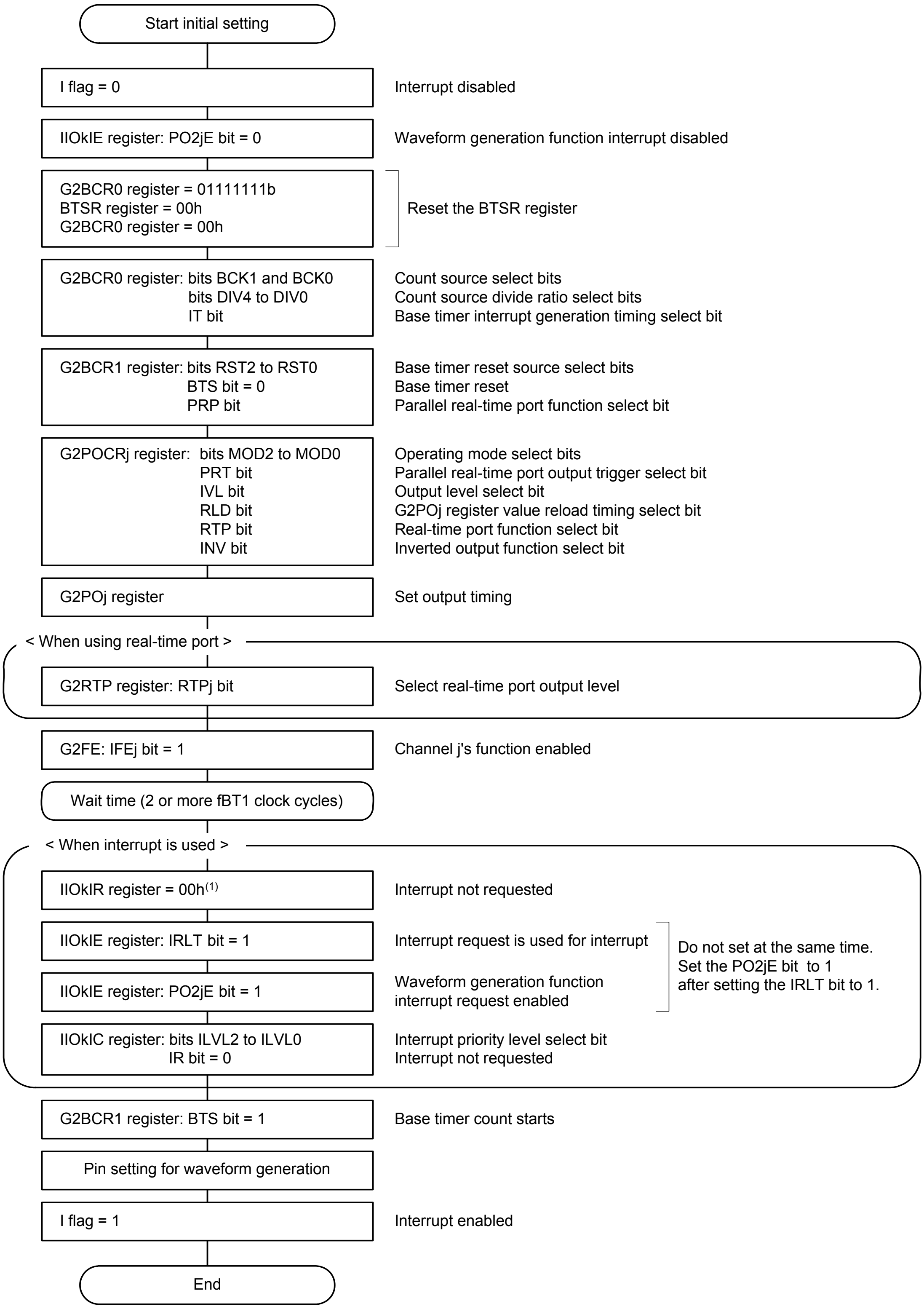
1. Set registers PS0 to PS3, PS5, PS7, and PS8 after setting the other registers.
2. This port is provided in the 144-pin package only.
3. P7_0 and P7_1 are N-channel open drain output ports.
4. Set the PS3 register immediately after the PRC2 bit in the PRCR register is set to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.



j = 0 to 7; k = 0 to 4, 8 to 10

NOTE:
1. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated in the same register (Interrupt does not occur).

Figure 22.25 Register Settings for Waveform Generation Function (Group 1)



j = 0 to 7; k = 3, 5 to 11

NOTE:
1. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated in the same register (Interrupt does not occur).

Figure 22.26 Register Settings for Waveform Generation Function (Group 2)

22.3.1 Single-Phase Waveform Output Mode (Group 1 and Group 2)

The OUTCi_j pin outputs “H” when the base timer value matches the GiPOj register value (i = 1, 2; j = 0 to 7), and outputs “L” when the base timer is reset.

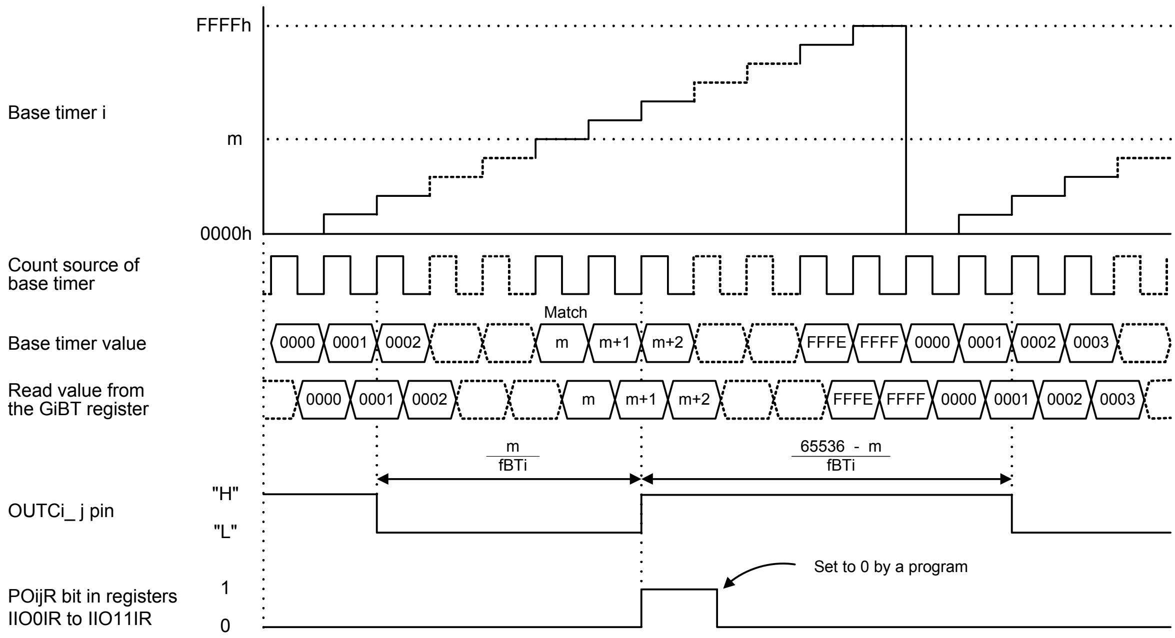
Table 22.7 lists specifications of single-phase waveform output mode. Figure 22.27 shows an example of single-phase waveform output mode operation.

Table 22.7 Single-Phase Waveform Output Mode Specifications

Item	Specification
Waveform generation channel	Group 1 and 2: channels 0 to 7
OUTCi_ j pin	Pulse output
Output waveform[1]	• Base timer is not reset: -The INV bit in the GiPOCRj register is set to 0 (output not inverted) -Bits UD1 and UD0 in G1BCR1 register are set to 00b (counter increment mode) Cycle: $\frac{65536}{fBTi}$ “L” width: $\frac{m}{fBTi}$ “H” width: $\frac{65536 - m}{fBTi}$ m: setting value of the GiPOj register: 0000h to FFFFh • Base timer is reset when base timer value matches the GiPO0 register value: -The INV bit in the GiPOCRj register is set to 0 (output not inverted) -Bits UD1 and UD0 in G1BCR1 register are set to 00b (counter increment mode) Cycle: $\frac{p + 2}{fBTi}$ “L” width: $\frac{m}{fBTi}$ “H” width: $\frac{p + 2 - m}{fBTi}$ m: setting value of the GiPOj register (0000h to FFFFh) p: setting value of the GiPO0 register (0001h to FFFDh) If $m \geq p + 2$, the output level is fixed to “L”
Waveform output start condition	Set both the BTS bit in the GiBCR1 register and the IFEj bit in the GiFE register to 1
Waveform output stop condition	Set either the BTS or IFEj bit to 0
Interrupt request generation timing	An interrupt request is generated at the second clock cycle after the base timer value matches the GiPOj register value. The POijR bit in the IIOkIR register (k = 0 to 11) becomes 1 (interrupt requested) when an interrupt request is generated. (See **Figure 11.18** IIO0IR to IIO11IR Registers)
Selectable function	• Initial value set function: Set the initial output level when waveform output is started (determined by the IVL bit in the GiPOCRj register) • Inverted output function: Output the inverted waveform level (determined by the INV bit in the GiPOCRj register)

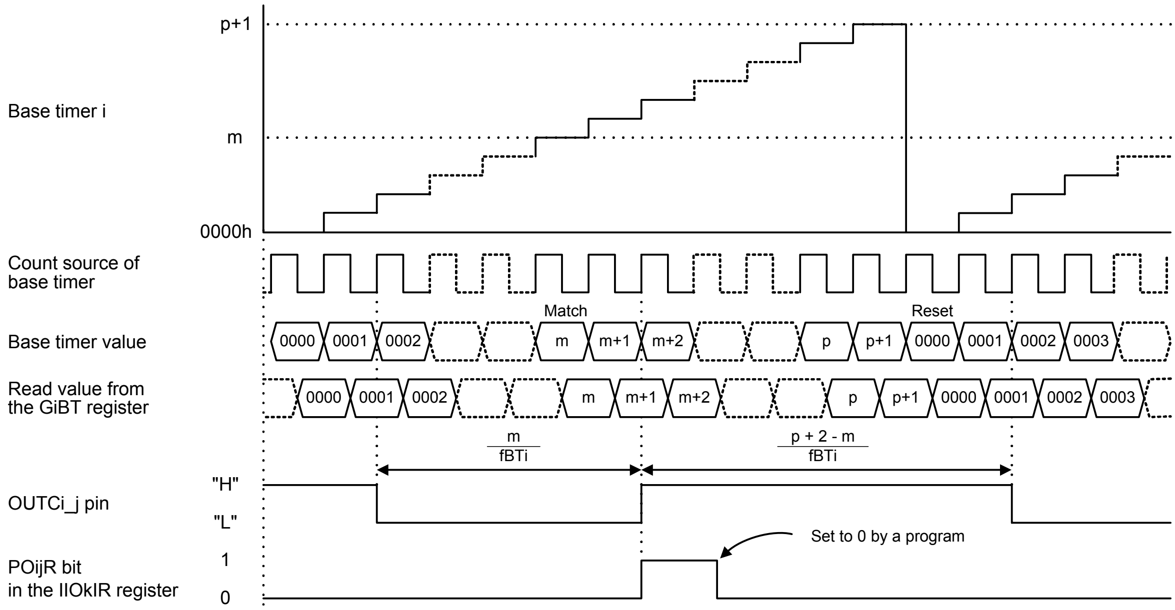
NOTE:
1. When the INV bit in the GiPOCRj register is set to 1 (output inverted), the “L” width and the “H” width are inversed.

(1) When the base timer is not reset



i = 1, 2; j = 0 to 7
m: Setting value of the GiPOj register (0000h to FFFFh)
The above applies under the following conditions:
- Group 1: In the G1BCR1 register, bits RST2 and RST1 are set to 00b and bits UD1 and UD0 are set to 00b (counter increment mode)
- Group 2: In the G2BCR1 register, bits RST2 to RST0 are set to 000b (Base timer is not reset by matching the G2PO0 register)
- In the GiPOCRj register, the IVL bit is set to 0 ("L" output) and the INV bit is set to 0 (output not inverted)

(2) When the base timer is reset by matching the GiPO0 register



i = 1, 2; j = 1 to 7; k = 0 to 5, 7 to 11
m: Setting value of the GiPOj register (0000h to FFFFh); p: Setting value of the GiPO0 register (0001h to FFFDh)
The above applies under the following conditions:
- Group 1: In the G1BCR1 register, bits RST2 and RST1 are set to 01b, and bits UD1 and UD0 are set to 00b (counter increment mode)
- Group 2: In the G2BCR1 register, bits RST2 to RST0 are set to 010b (Base timer is reset by matching the G2PO0 register)
- In the GiPOCRj register, the IVL bit is set to 0 ("L" output) and the INV bit is set to 0 (output not inverted)
- m < p + 2

Figure 22.27 Single-Phase Waveform Output Mode Operation

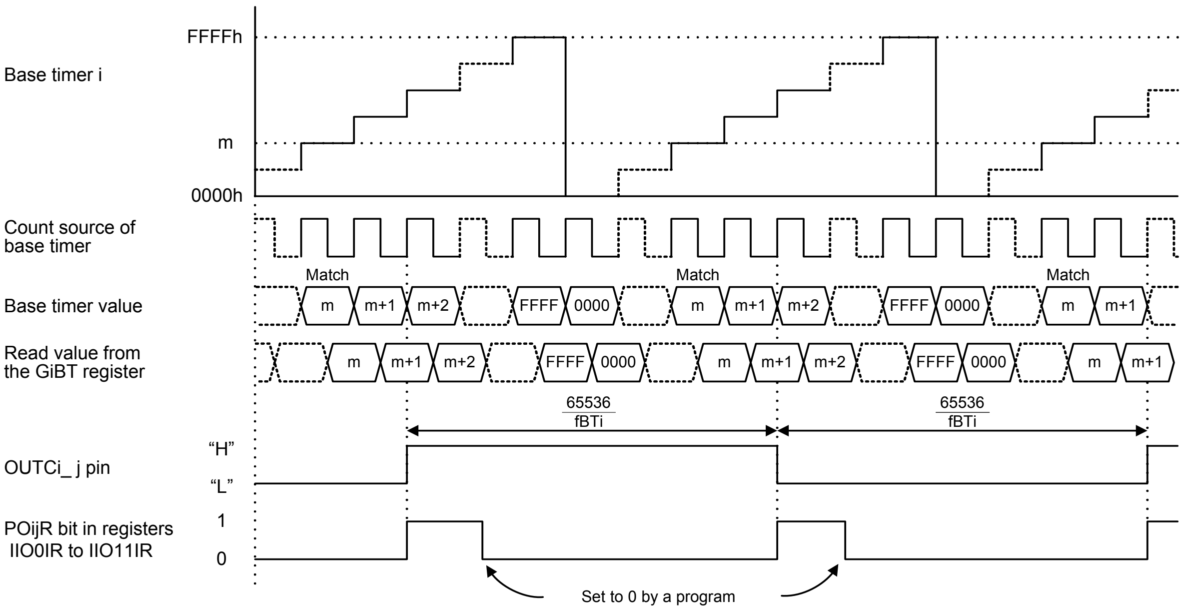
22.3.2 Phase-Delayed Waveform Output Mode (Group 1 and Group 2)

Output level from the OUTCi_j pin is inverted every time the base timer value matches the GiPOj register value (i = 1, 2; j = 0 to 7).
Table 22.8 lists specifications of phase-delayed waveform output mode. Figure 22.28 shows an example of phase-delayed waveform output mode operation.

Table 22.8 Phase-Delayed Waveform Output Mode Specifications

Item	Specification
Waveform generation channel	Group 1 and 2: channels 0 to 7
OUTCi_ j pin	Pulse output
Output waveform	• Base timer is not reset: -Bits UD1 and UD0 in G1BCR1 register are set to 00b (counter increment mode) Cycle: $\frac{65536 \times 2}{fBTi}$ "H" and "L" widths: $\frac{65536}{fBTi}$ • Base timer is reset when base timer value matches the GiPO0 register value: -Bits UD1 and UD0 in G1BCR1 register are set to 00b (counter increment mode) Cycle: $\frac{2 (p + 2)}{fBTi}$ "H" and "L" widths: $\frac{p + 2}{fBTi}$ p: setting value of the GiPO0 register (0001h to FFFDh) If GiPOq register value (q = 1 to 7) (0000h to FFFFh) ≥ p + 2, the output level is not inverted
Waveform output start condition	Set both the BTS bit in the GiBCR1 register and the IFEj bit in the GiFE register to 1
Waveform output stop condition	Set either the BTS or IFEj bit to 0
Interrupt request generation timing	An interrupt request is generated at the second clock cycle after the base timer value matches the GiPOj register value. The POijR bit in the IIOkIR register (k = 0 to 11) becomes 1 (interrupt requested) when an interrupt request is generated. (See **Figure 11.18** IIO0IR to IIO11IR Registers)
Selectable function	• Initial value set function: Set the initial output level when waveform output is started (determined by the IVL bit in the GiPOCRj register) • Inverted output function: Output the inverted waveform level (determined by the INV bit in the GiPOCRj register)

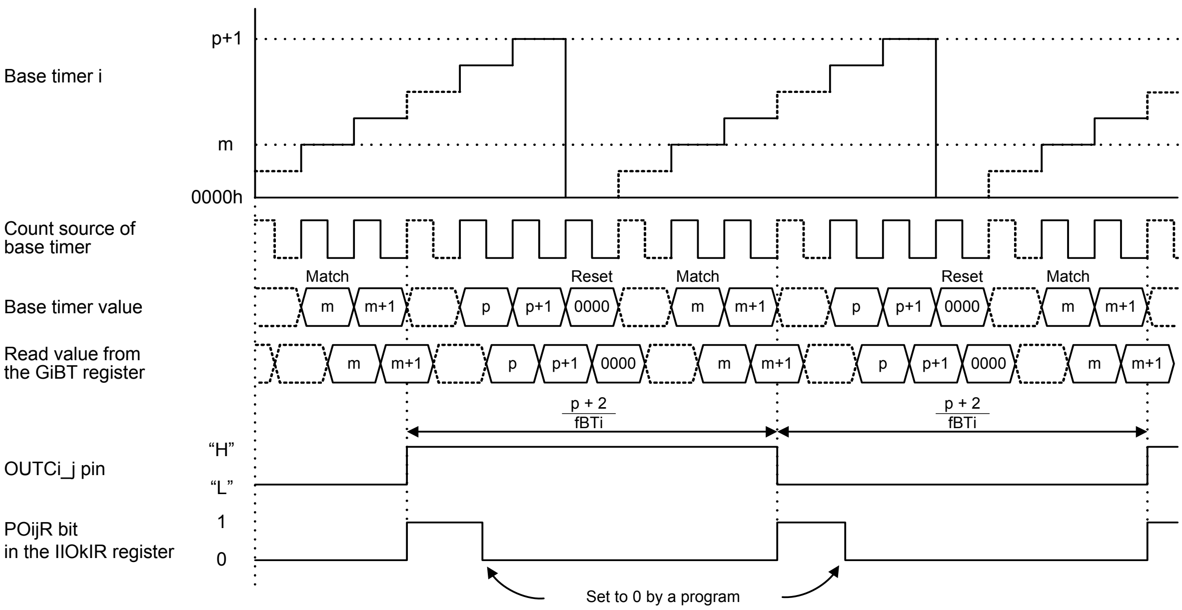
(1) When the base timer is not reset



i = 1, 2; j = 0 to 7 m: Setting value of the GiPOj register (0000h to FFFFh)

The above applies under the following conditions:
- Group 1: G1BCR1 register; bits RST2 and RST1 are set to 00b (Base timer is not reset by matching the G1PO0 register)
  bits UD1 and UD0 are set to 00b (counter increment mode)
- Group 2: Bits RST2 to RST0 in the G2BCR1 register are set to 000b (Base timer is not reset by matching the G2PO0 register)
- In the GiPOCRj register, the IVL bit is set to 0 ("L" output) and the INV bit is set to 0 (output not inverted)

(2) When the base timer is reset by matching the GiPO0 register



i = 1, 2; j = 1 to 7; k = 0 to 5, 7 to 11
m: Setting value of the GiPOj register (0000h to FFFFh); p: Setting value of the GiPO0 register (0001h to FFFDh)

The above applies under the following conditions:
- Group 1: G1BCR1 register; bits RST2 and RST1 are set to 01b (Base timer is reset by matching the G1PO0 register)
  bits UD1 and UD0 are set to 00b (counter increment mode)
- Group 2: Bits RST2 to RST0 in the G2BCR1 register are set to 010b (Base timer is reset by matching the G2PO0 register)
- In the GiPOCRj register, the IVL bit is set to 0 ("L" output) and the INV bit is set to 0 (output not inverted)
- m < p + 2

Figure 22.28 Phase-Delayed Waveform Output Mode Operation

22.3.3 Set/Reset (SR) Waveform Output Mode (Group 1 and Group 2)

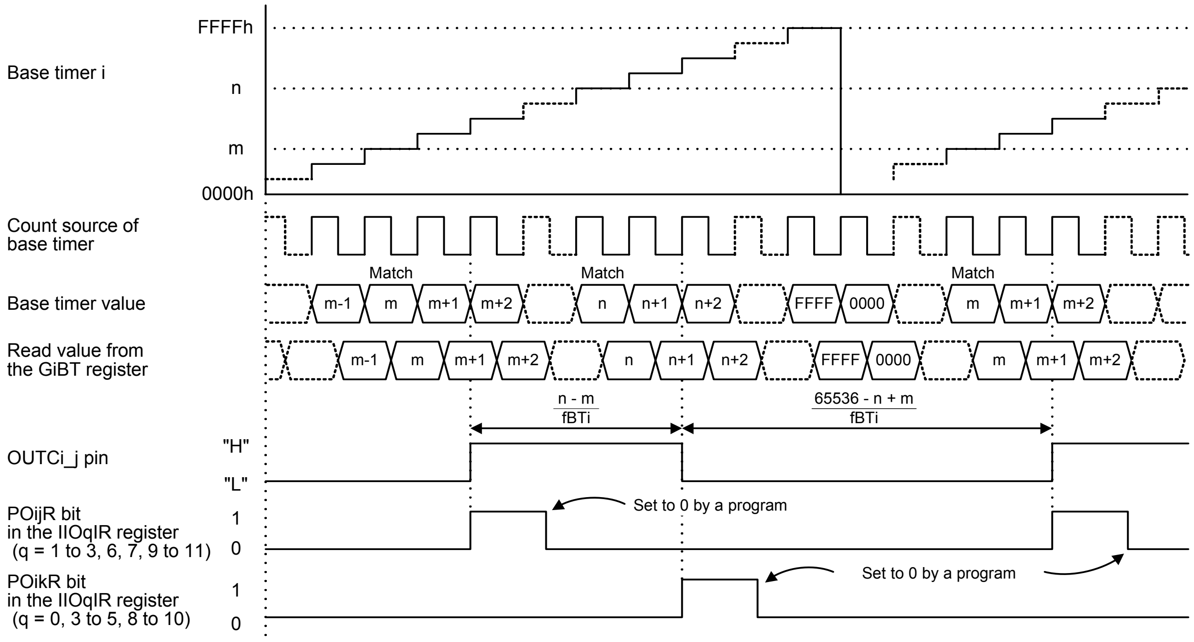
The OUTCi_j pin outputs "H" when the base timer value matches the GiPOj register value (i = 1, 2; j = 0, 2, 4, 6), and outputs "L" when the base timer value matches the GiPOk register value (k = j + 1) or when the base timer is reset. Table 22.9 lists specifications of SR waveform output mode. Figure 22.29 shows an example of SR waveform output mode operation.

Table 22.9 SR Waveform Output Mode Specifications

Item	Specification
Waveform generation channel[(1)]	Group 1 and 2: channels 0, 2, 4, 6
OUTCi_j pin	Pulse output
Output waveform[(1)(2)]	• Base timer is not reset: -The INV bit in the GiPOCRj register is set to 0 (output not inverted) -Bits UD1 and UD0 in G1BCR1 register are set to 00b (counter increment mode) (1) $m < n$ "H" width: $\frac{n - m}{fBTi}$ "L" width : $\frac{65536 - n + m}{fBTi}$ (2) $m \geq n$ "H" width: $\frac{65536 - m}{fBTi}$ "L" width : $\frac{m}{fBTi}$ m: setting value of the GiPOj register (0000h to FFFFh) n: setting value of the GiPOk register (0000h to FFFFh) • Base timer is reset when base timer value matches the GiPO0 register value[(1)]: -The INV bit in the GiPOCRj register is set to 0 (output not inverted) -Bits UD1 and UD0 in G1BCR1 register are set to 00b (counter increment mode) (1) $m < n < p + 2$ "H" width: $\frac{n - m}{fBTi}$ "L" width : $\frac{p + 2 - n + m}{fBTi}$ (2) $m < p + 2 \leq n$ "H" width: $\frac{p + 2 - m}{fBTi}$ "L" width : $\frac{m}{fBTi}$ (3) $m \geq p + 2$, the output level is fixed to "L" m: setting value of the GiPOq register (q = 2, 4, 6) (0000h to FFFFh) n: setting value of the GiPOk register (0000h to FFFFh) p: setting value of the GiPO0 register (0001h to FFFDh)
Waveform output start condition	Set both the BTS bit in the GiBCR1 register and the IFEj bit in the GiFE register to 1
Waveform output stop condition	Set either the BTS or IFEj bit to 0
Interrupt request generation timing	An interrupt request is generated at the second clock cycle after the base timer value matches the GiPOj register value. The POirR bit in the IIOsIR register becomes 1 (interrupt requested) when an interrupt request is generated. (r = 0 to 7; s = 0 to 11) (See **Figure 11.18** IIO0IR to IIO11IR Registers)
Selectable function	• Initial value set function: Set the initial output level when waveform output is started (determined by the IVL bit in the GiPOCRj register) • Inverted output function: Output the inverted waveform level (determined by the INV bit in the GiPOCRj register)

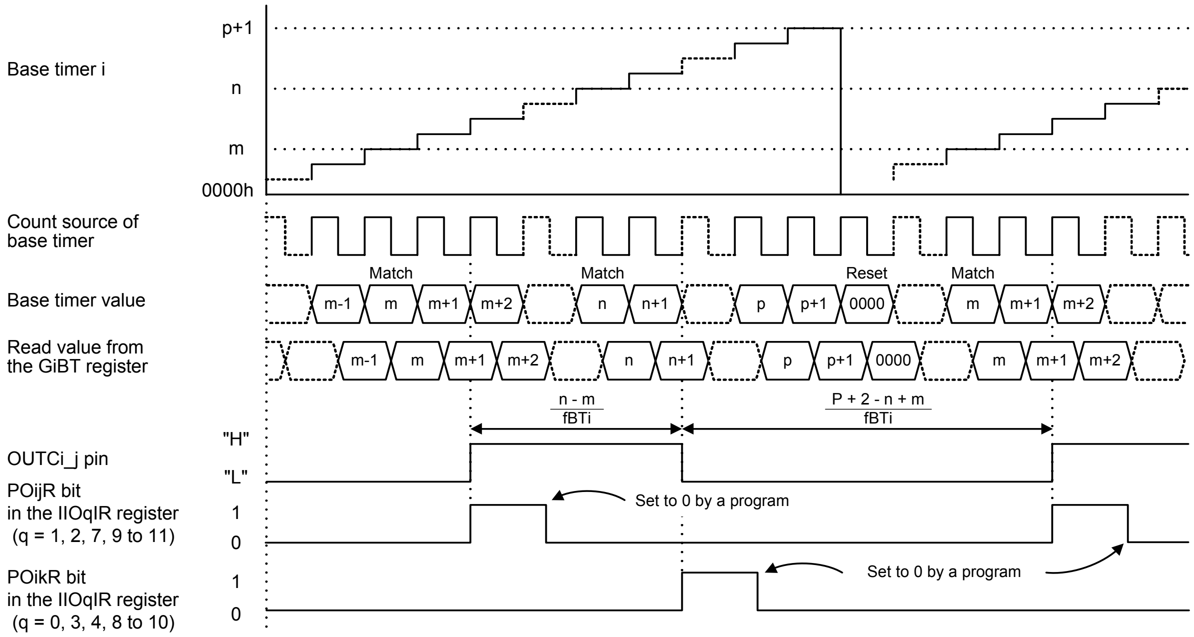
NOTES:
1. If the base timer is reset when the base timer value matches the GiPO0 register, the SR waveform generation function in the channel 0 can not be used.
2. When the INV bit in the GiPOCRj register is set to 1 (output inverted), the "L" width and the "H" width are inversed.

(1) When the base timer is not reset



i = 1, 2; j = 0, 2, 4, 6; k = j + 1
m: Setting value of the GiPOj register (0000h to FFFFh), n: Setting value of the GiPOk register (0000h to FFFFh)
The above applies under the following conditions:
- Group 1: In the G1BCR1 register, bits RST2 and RST1 bit are set to 00b and bits UD1 and UD0 are set to 00b (counter increment mode)
- Group 2: Bits RST2 to RST0 in the G2BCR1 register are set to 000b (base timer is not reset by matching the G2PO0 register)
- In the GiPOCRj register, the IVL bit is set to 0 ("L" output) and the INV bit is set to 0 (output not inverted)
- m < n

(2) When the base timer is reset by matching the GiPO0 register



i = 1, 2; j = 2, 4, 6; k = j + 1
m: Setting value of the GiPOj register (0000h to FFFFh), n: Setting value of the GiPOk register (0000h to FFFFh)
p: Setting value of the GiPO0 register (0001h to FFFDh)
The above applies under the following conditions:
- Group 1: In the G1BCR1 register, bits RST2 and RST1 are set to 01b and bits UD1 and UD0 are set to 00b (counter increment mode)
- Group 2: Bits RST2 to RST0 in the G2BCR1 register are set to 010b (base timer is reset by matching the G2PO0 register)
- In the GiPOCRj register, the IVL bit is set to 0 ("L" output) and the INV bit is set to 0 (output not inverted)
- m < n < p + 2

Figure 22.29 SR Waveform Output Mode Operation

22.3.4 Bit Modulation PWM Output Mode (Group 2)

In bit modulation PWM output mode, 16-bit PWM duty ratio can be achieved with a collection of 6-bit PWM pulses. A series of 1024 pulses whose "L" widths are specified with 6-bit PWM, is repeatedly output. The six high-order bits in the G2POi register (i = 0 to 7) determine the base "L" width. The ten low-order bits determine the number of pulses (modulated pulses) whose "L" widths are extended by one fBT2 clock cycle.
Table 22.10 lists specifications of bit modulation PWM output mode. Table 22.11 lists the number of modulated pulses and their locations. Figure 22.30 shows an example of bit modulation PWM output mode operation.

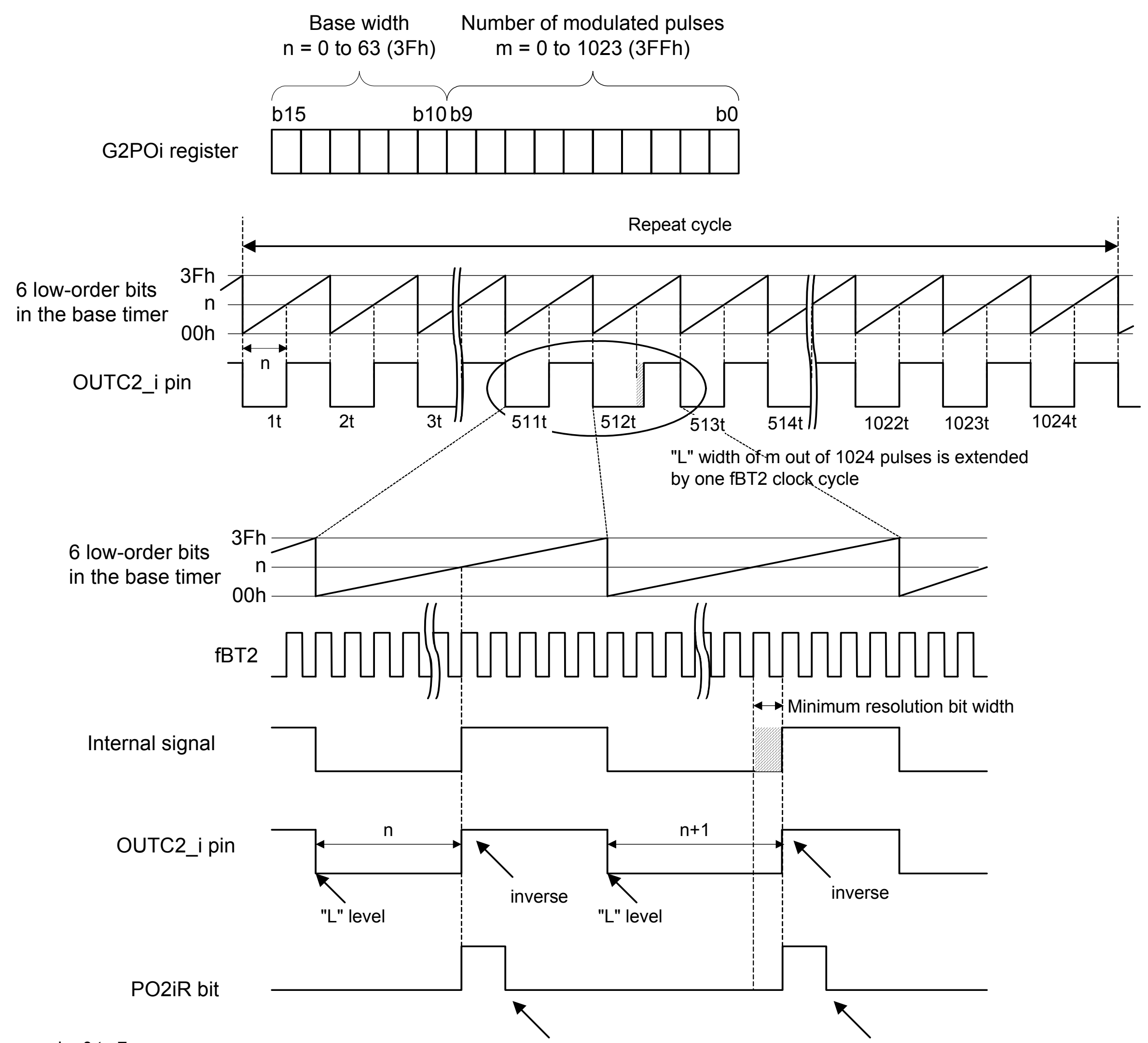
Table 22.10 Specifications of Bit Modulation PWM Output Mode

Item	Specification
Waveform generation channels	Group 2: channels 0 to 7(1)
OUTC2_i pin	Pulse output
Output waveform(2)(3)	PWM cycle: $\frac{64}{fBT2}$ (= t) Repeat cycle: $\frac{65536}{fBT2}$ (= $\frac{64}{fBT2}$ × 1024) "L" width: $\frac{n+1}{fBT2}$: for m pulses, $\frac{n}{fBT2}$: for (1024 - m) pulses Average "L" width: $\frac{1}{fBT2} \times (n + \frac{m}{1024})$ n: setting value of the six high-order bits in the G2POi register (00h to 3Fh) m: setting value of the ten low-order bits in the G2POi register (000h to 3FFh)
Waveform output start condition	Set both the BTS bit in the G2BCR1 register and the IFEi bit in the G2FE register to 1
Waveform output stop condition	Set either the BTS or IFEi bit to 0
Interrupt request generation timing	An interrupt request is generated at the second clock cycle after the base timer value matches the G2POi register value. The PO2iR bit in the IIOkIR register (k = 3, 5 to 11) becomes 1 (interrupt requested) when an interrupt request is generated. (See **Figure 11.18** IIO0IR to IIO11IR Registers)
Selectable function	• Initial value set function: Set the initial output level when waveform output is started (determined by the IVL bit in the G2POCRi register) • Inverted output function: Output the inverted waveform level (determined by the INV bit in the G2POCRi register)

NOTES:
1. Channels 0 to 7 are provided in the 144-pin package. Channels 0 to 2 are provided in the 100-pin package.
2. Set bits RST2 to RST0 in the G2BCR1 register to 000b to use bit modulation PWM mode.
3. When the INV bit in the G2POCRi register is set to 1 (output inverted), the "L" width and the "H" width are inversed.

Table 22.11 Number of Modulated Pulses and Locations

Ten low-order bits in the G2POi register	Number of Pulses	Location
00 0000 0000b	0	none
00 0000 0001b	1	512t
00 0000 0010b	2	256t, 768t
00 0000 0100b	4	128t, 384t, 640t, 896t
00 0000 1000b	8	64t, 192t, 320t, 448t, 576t, 704t, 832t, 960t
...	...	...
10 0000 0000b	512	1t, 3t, 5t, 7t, ... 1019t, 1021t, 1023t



i = 0 to 7

PO2iR bit: Bit in the register IIO3IR to IIO11IR

The above applies under the following conditions:
- In the G2POCRj register, the IVL bit is set to 0 ("L" output) and the INV bit is set to 0 (output not inverted)
- m = 1

Figure 22.30 Bit Modulation PWM Output Mode Operation

22.3.5 Real-Time Port Output Mode (Group 2)

The OUTC2_i pin (i = 0 to 7) outputs the value of the RTPi bit in the G2RTP register when the base timer value matches the G2POi register. To use real-time output mode, set the RTP bit in the G2POCRi register to 1 and the PRT bit to 0 in the channel used for this mode. Also, set the PRP bit in the G2BCR1 register to 0.
Table 22.12 lists specifications of real-time port output mode. Figure 22.31 shows a block diagram. Figure 22.32 shows an example of real-time port output mode operation.

Table 22.12 Specifications of Real-Time Port Output Mode

Item	Specification
Waveform generation channels	Group 2: channels 0 to 7(1)
OUTC2_i pin	Real-time port output
Waveform output start condition	Set both the BTS bit in the G2BCR1 register and the IFEi bit in the G2FE register to 1
Waveform output stop condition	Set either the BTS or IFEi bit to 0
Interrupt request generation timing	An interrupt request is generated at the second clock cycle after the base timer value matches the G2POi register value. The PO2iR bit in the IIOkIR register (k = 3, 5 to 11) becomes 1 (interrupt requested) when an interrupt request is generated. (See **Figure 11.18** IIO0IR to IIO11IR Registers)
Selectable function	• Initial value set function: Set the initial output level when waveform output is started (determined by the IVL bit in the G2POCRi register)

NOTE:
1. Channels 0 to 7 are provided in the 144-pin package. Channels 0 to 2 are provided in the 100-pin package.

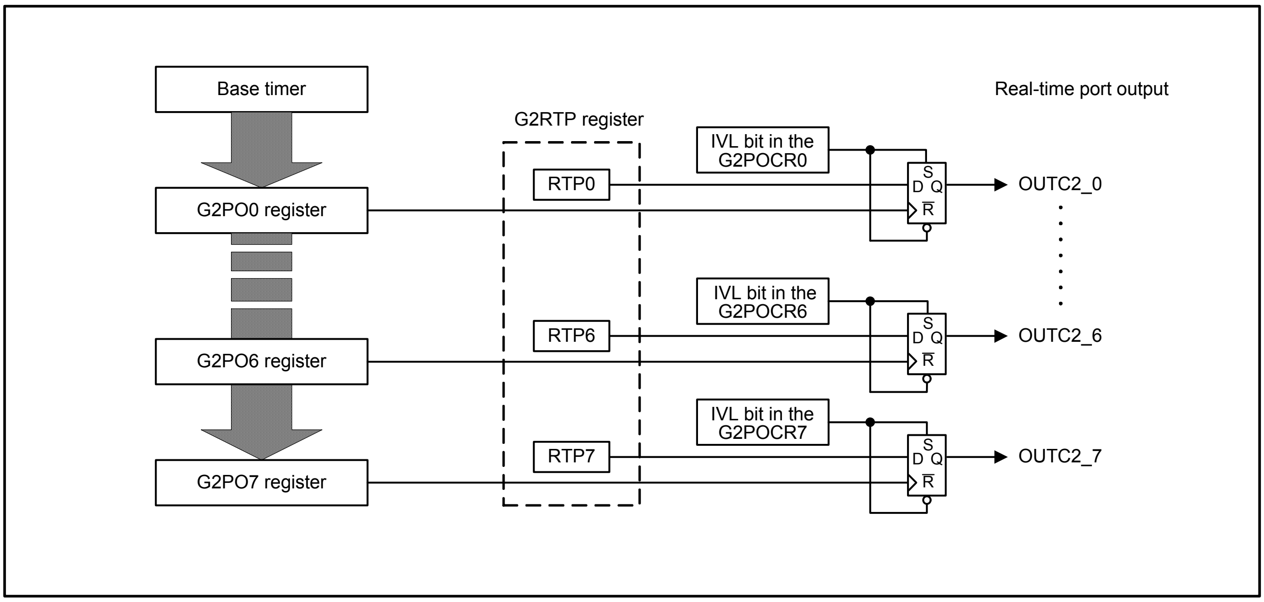
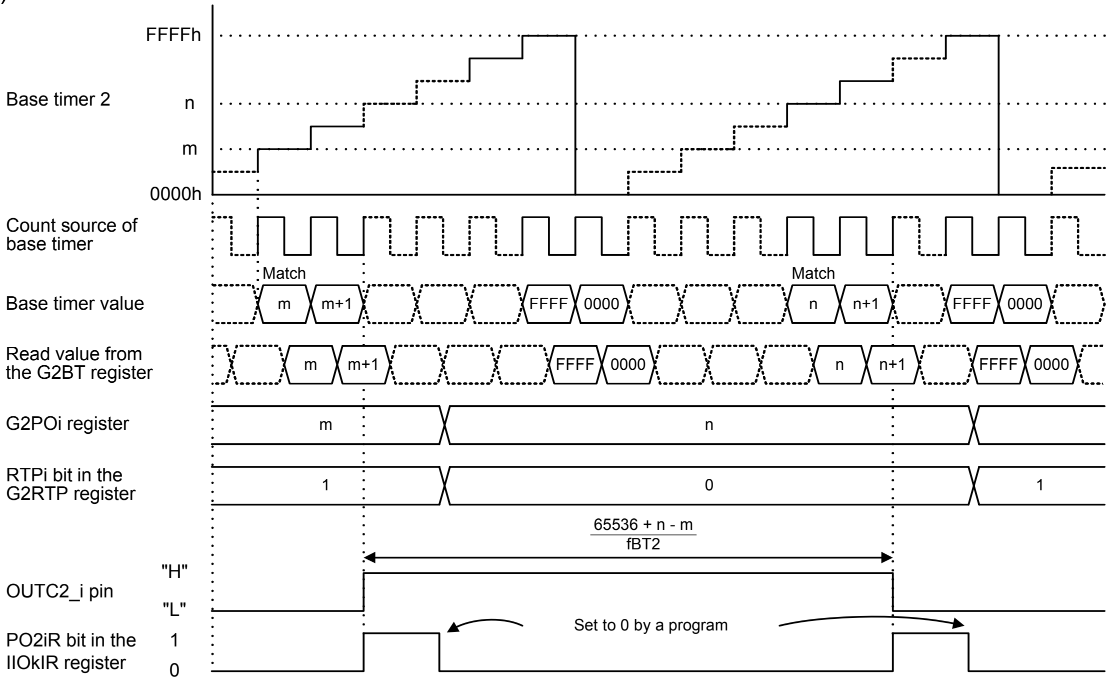


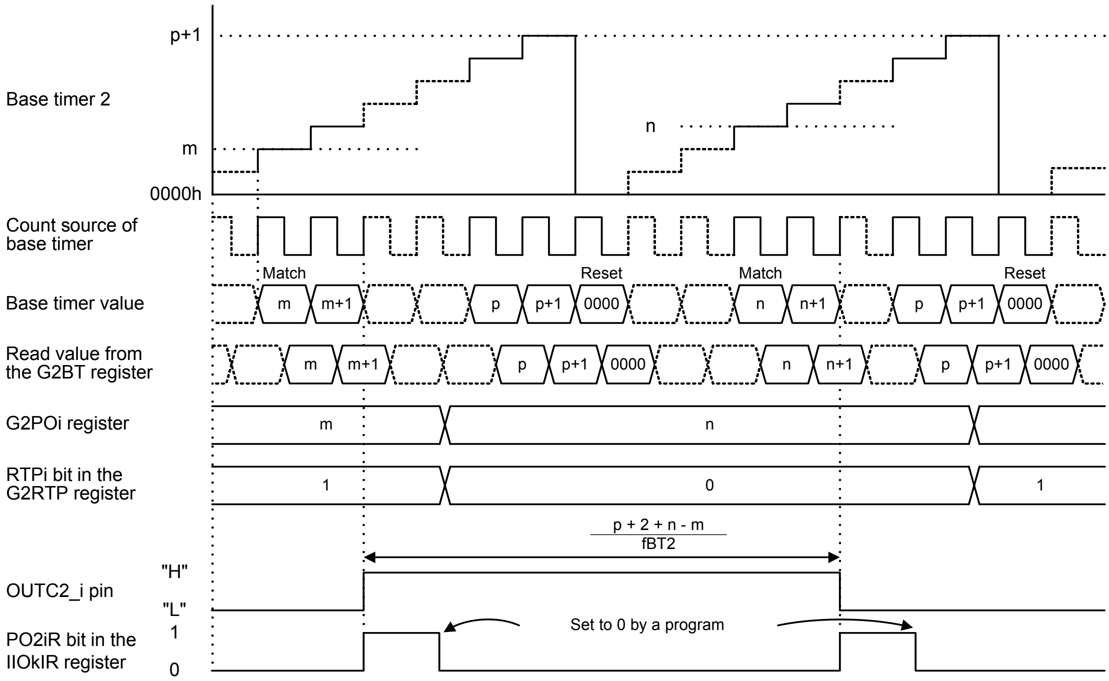
Figure 22.31 Real-Time Port Output Function Block Diagram

(1) When the base timer is not reset



i = 0 to 7; k = 3, 5 to 11 m, n: Setting value of the G2POj register (0000h to FFFFh)
The above applies under the following conditions:
- Bits RST2 to RST0 in the G2BCR1 register are set to 000b (base timer is not reset by matching the G2PO0 register)
- In the G2POCRj register, the IVL bit is set to 0 ("L" output) and the RLD bit is set to 1 (reload when the base timer is reset)

(2) When the base timer is reset by matching the G2PO0 register



i = 1 to 7; k = 3, 5, 7 to 11
m, n: Setting value of the G2POi register (0000h to FFFFh); p: Setting value of the G2PO0 register (0001h to FFFDh)
The above applies under the following conditions:
- Bits RST2 to RST0 in the G2BCR1 register are set to 010b (base timer is reset by matching the G2PO0 register)
- In the G2POCRi register, the IVL bit is set to 0 ("L" output) and the RLD bit is set to 1 (reload when the base timer is reset)
- m < n < p + 2

Figure 22.32 Real-Time Port Output Mode Operation

22.3.6 Parallel Real-Time Port Output Mode (Group 2)

In parallel real-time port output mode, all of the channels which the RTP bit in the G2POCRi register (i = 0 to 7) is set to 1, perform the parallel real-time port output. The value set in the G2RTP register is output from the OUTC2_i pin in these channels, when the base timer value matches any of the G2POi register which the RTP bit is set to 1. Real-time port output and parallel real-time port output cannot be used in the same group. To use parallel real-time port output, set the RTP bit to 1 and the PRT bit to 1 in the channel used for parallel real-time port output. Also, set the PRP bit in the G2BCR1 register to 1.

Table 22.13 lists specifications of parallel real-time port output mode. Figure 22.33 shows a block diagram. Figure 22.34 shows an example of parallel real-time port output mode operation.

Table 22.13 Specifications of parallel real-time port output mode

Item	Specification
Waveform generation channels	Group 2: channels 0 to 7(1)
OUTC2_i pin	Real-time port output
Waveform output start condition	Set both the BTS bit in the G2BCR1 register and the IFEi bit in the G2FE register to 1
Waveform output stop condition	Set either the BTS or IFEi bit to 0
Interrupt request generation timing	An interrupt request is generated at the second clock cycle after the base timer value matches the G2POi register value. The PO2iR bit in the IIOkIR register (k = 3, 5 to 11) becomes 1 (interrupt requested) when an interrupt request is generated. (See **Figure 11.18** IIO0IR to IIO11IR Registers)
Selectable function	• Initial value set function: Set the initial output level when waveform output is started (determined by the IVL bit in the G2POCRi register)

NOTE:
1. Channels 0 to 7 are provided in the 144-pin package. Channels 0 to 2 are provided in the 100-pin package.

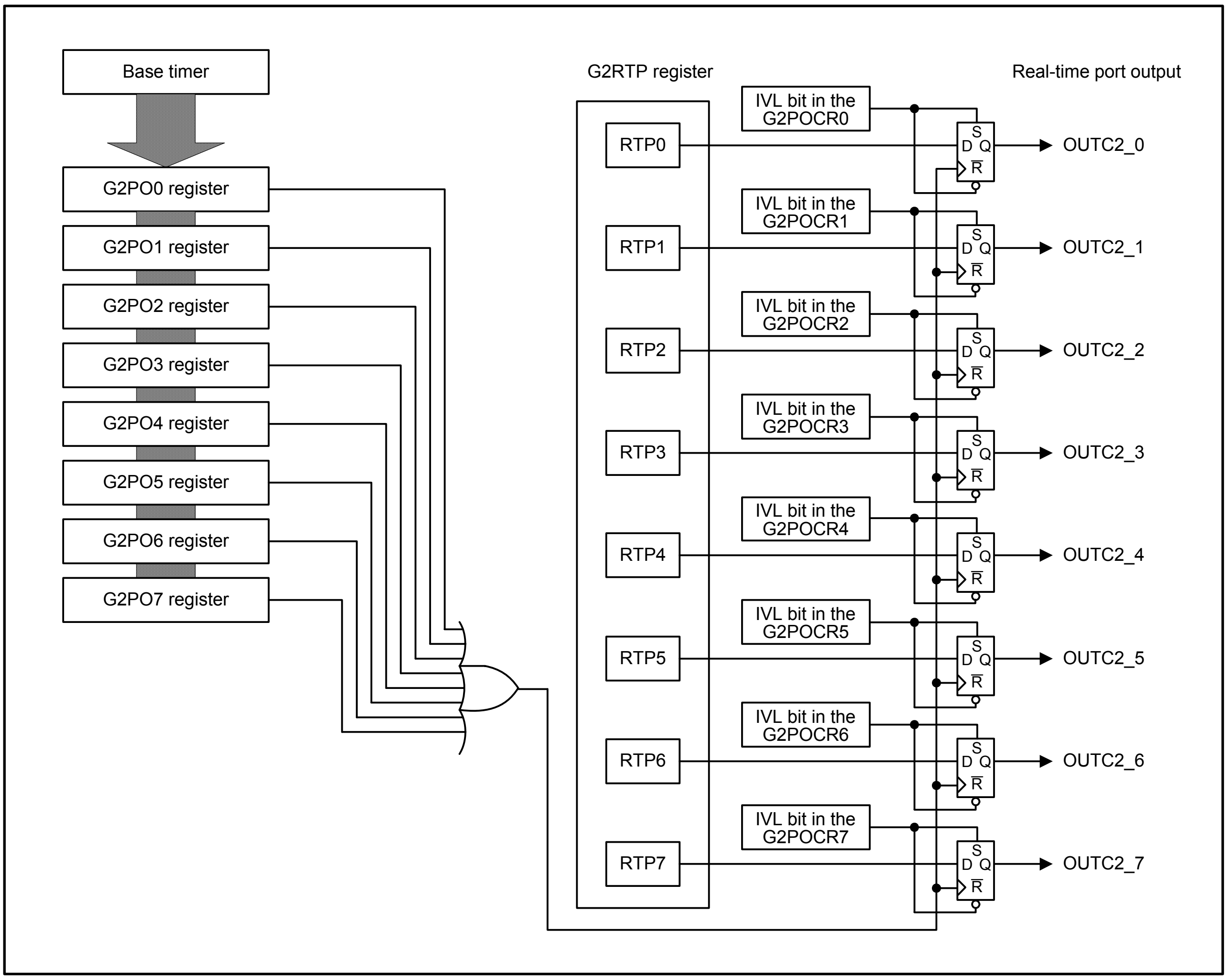
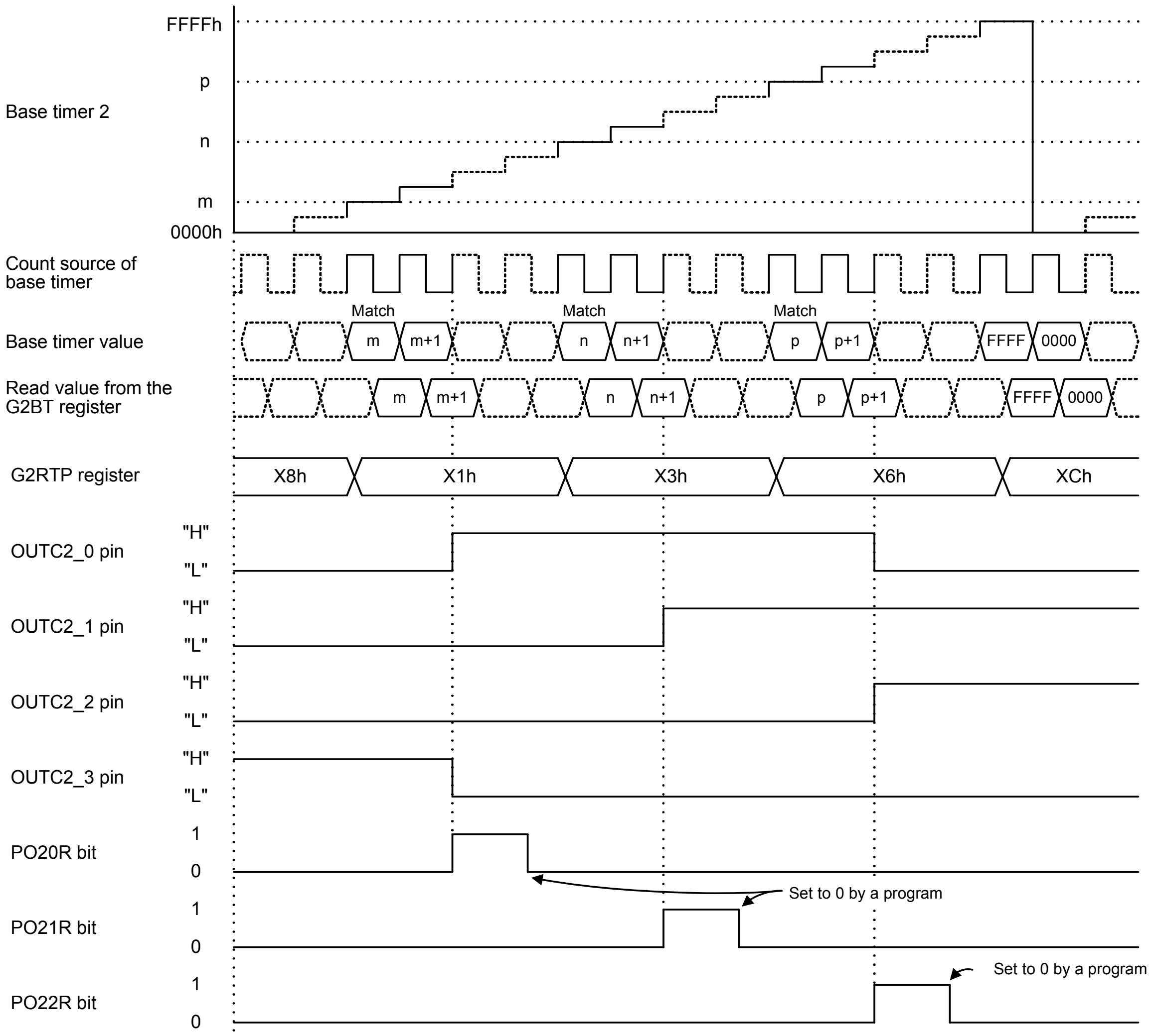


Figure 22.33 Parallel Real-Time Port Output Function Block Diagram



m: Setting value of the G2PO0 register (0000h to FFFFh)
n: Setting value of the G2PO1 register (0000h to FFFFh)
p: Setting value of the G2PO2 register (0000h to FFFFh)

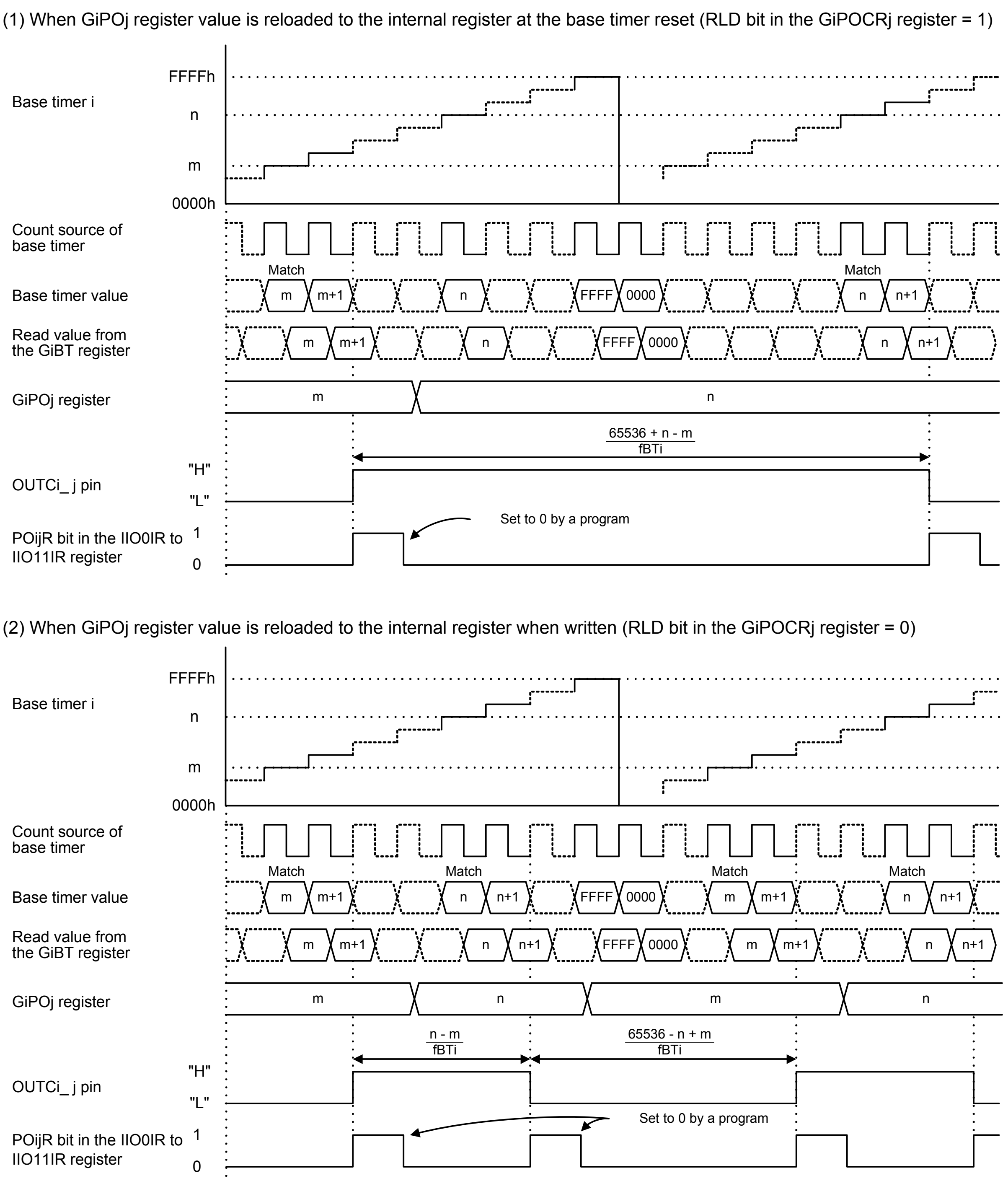
PO20R, PO21R, and PO22R: Bits in registers IIO5IR to IIO7IR

The above applies under the following conditions:
- The IVL bit in the G2POCRi (i = 0 to 3) register is set to 0 ("L" output)
- Bits RST2 to RST0 in the G2BCR1 register are set to 000b (base timer is not reset by matching the G2PO0 register)
- m < n < p

Figure 22.34 Parallel Real-Time Port Output Mode Operation

22.3.7 GiPOj Register Value Reload Timing Select Function (i = 1, 2; j = 0 to 7)

The RLD bit in the GiPOCRj register determines whether the GiPOj register value is reloaded to the internal register when the value is written, or when the base timer is reset. Figure 22.35 shows an operation example.



i = 1, 2; j = 0 to 7; m: Setting value of the GiPOj register (0000h to FFFFh); n: Setting value of the GiPO0 register (0000h to FFFFh)
The above applies under the following conditions:
- Group 1: G1BCR1 register; bits RST2 and RST1 are set to 00b (base timer is not reset by matching the G1PO0 register)
bits UD1 and UD0 are set to 00b (counter increment mode)
- Group 2: Bits RST2 and RST0 in the G2BCR1 register are set to 000b (base timer is not reset by matching the G2PO0 register)
- GiPOCRj register: bits MOD2 to MOD0 are set to 010b (phase-delayed waveform output mode),
the IVL bit is set to 0 ("L" output), and the INV bit is set to 0 (output not inverted)
- m < n

Figure 22.35 GiPOj Register Value Reload Timing Select Function Operation

22.4 Group 0 and Group 1 Communication Function

In the group 0 communication function, clock synchronous mode or HDLC data processing mode is available. In the group 1 communication function, clock synchronous mode, clock asynchronous (UART) mode, or HDLC data processing mode is available. Figure 22.36 shows a block diagram of group 0 communication function. Figure 22.37 shows a block diagram of group 1 communication function. Figures 22.38 to 22.46 show registers associated with the communication function.

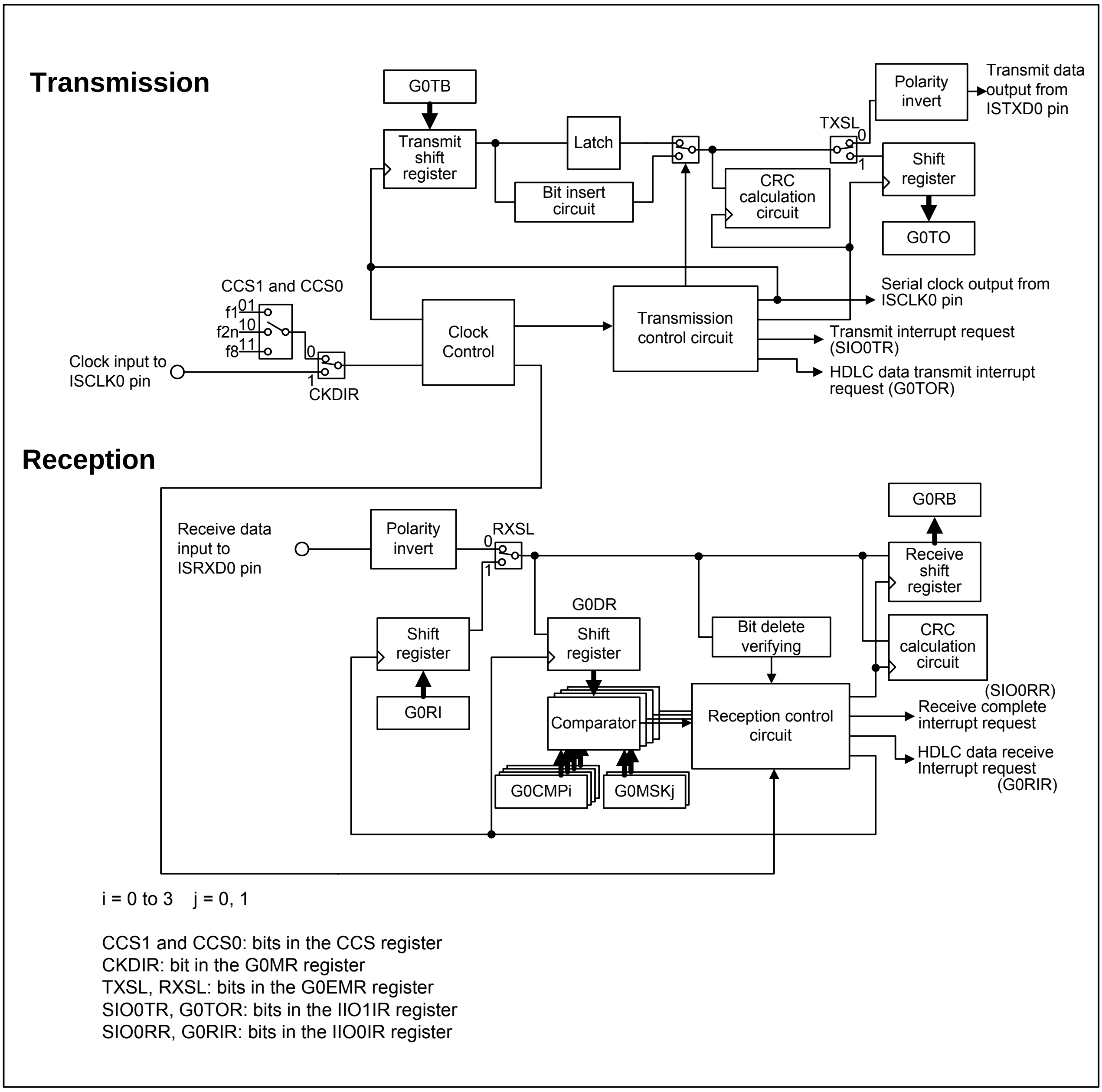
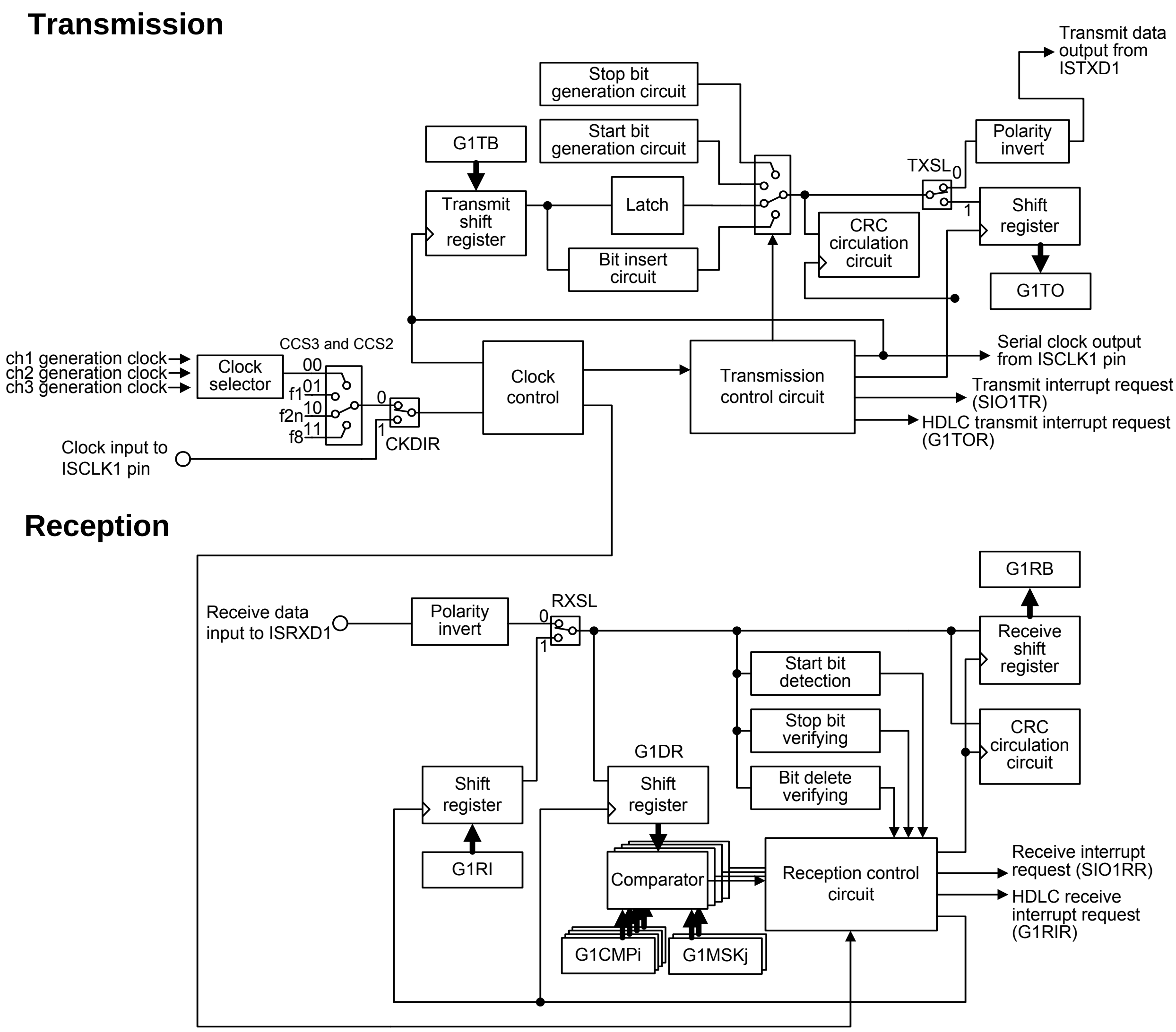


Figure 22.36 Group 0 Communication Function Block Diagram



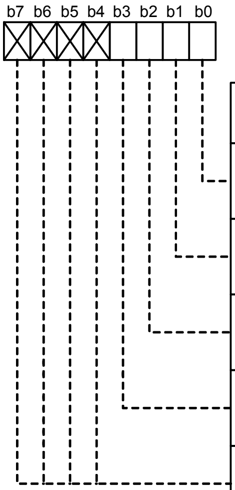
i = 0 to 3 j = 0,1

CCS3 and CCS2: bits in the CCS register
CKDIR: bit in the G1MR register
TXSL, RXSL: bits in the G1EMR register
SIO1TR, G1TOR: bits in the IIO3IR register
SIO1RR, G1RIR: bits in the IIO2IR register

NOTE:
1. After a clock, which is selected in the G1BCR0 register, is supplied to the registers, each register value becomes the after reset value.

Figure 22.37 Group 1 Communication Function Block Diagram

Communication Clock Select Register



Symbol	Address	After Reset
CCS	00F6h	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
CCS0	Group 0 clock select bits(1)	b1 b0 0 0 : Do not set to this value 0 1 : f1 1 0 : f2n(2) 1 1 : f8	RW
CCS1			RW
CCS2	Group 1 clock select bits(1)	b3 b2 0 0 : Clock generated with a waveform generation function 0 1 : f1 1 0 : f2n(2) 1 1 : f8	RW
CCS3			RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Set the selected clock frequency to 5MHz or lower in the clock synchronous mode.
2. Bits CNT3 to CNT0 in the TCSPR register select no division (n=0) or divide-by-2n (n=1 to 15).

Figure 22.38 CCS Register

Group 0 SI/O Communication Mode Register

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

Symbol: G0MR | Address: 00EDh | After Reset: 00h

Bit Symbol	Bit Name	Function	RW
GMD0	Communication mode select bits	b1 b0 0 1 : Clock synchronous mode 1 1 : HDLC data processing mode Do not set to values other than the above.	RW
GMD1			RW
CKDIR	Clock select bit	0 : Internal clock 1 : External clock	RW
– (b5-b3)	Reserved bits	Set to 0	RW
UFORM	Bit order select bit	0 : LSB first 1 : MSB first	RW
IRS	Transmit interrupt source select bit	0 : No data in the G0TB register (TI=1) 1 : Transmit operation is completed (TXEPT=1)	RW

Group 1 SI/O Communication Mode Register

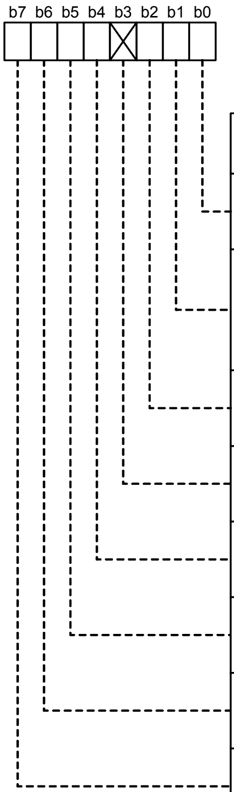
b7 b6 b5 b4 b3 b2 b1 b0

Symbol: G1MR | Address: 012Dh | After Reset: 00h

Bit Symbol	Bit Name	Function	RW
GMD0	Communication mode select bits	b1 b0 0 0 : UART mode 0 1 : Clock synchronous mode 1 0 : Do not set to this value 1 1 : HDLC data processing mode	RW
GMD1			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	0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
UFORM	Bit order select bit	0 : LSB first 1 : MSB first	RW
IRS	Transmit interrupt source select bit	0 : No data in the G0TB register (TI=1) 1 : Transmit operation is completed (TXEPT=1)	RW

Figure 22.39 G0MR and G1MR Registers

Group i SI/O Communication Control Register (i=0,1)



Symbol	Address	After Reset
G0CR	00EFh	0000 X011b
G1CR	012Fh	0000 X011b

Bit Symbol	Bit Name	Function	RW
TI	GiTB register empty flag	0 : Data in the GiTB register 1 : No data in the GiTB register	RO
TXEPT	Transmit shift register empty flag	0 : Data in the transmit shift register (during transmission) 1 : No data in the transmit shift register (transmit completed)	RO
RI	Receive complete flag	0 : No data in the GiRB register 1 : Data in the GiRB register	RO
– (b3)	Unimplemented. Write 0. Read as undefined value.		–
TE	Transmit enable bit	0 : Transmit operation disabled 1 : Transmit operation enabled	RW
RE	Receive enable bit	0 : Receive operation disabled 1 : Receive operation enabled	RW
IPOL	ISRXD input polarity invert bit	0 : Not inverted 1 : Inverted(1)	RW
OPOL	ISTXD output polarity invert bit	0 : Not inverted 1 : Inverted(1)	RW

NOTE:
1. Set these bits to 1 when using UART mode.

Figure 22.40 G0CR, G1CR Registers

Group i SI/O Expansion Mode Register (i=0, 1)[1]

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

Symbol	Address	After Reset
G0EMR	00FCh	00h
G1EMR	013Ch	00h

Bit Symbol	Bit Name	Function	RW
– (b0)	Reserved bit	Set to 0	RW
CRCV	CRC initial value select bit	0 : Set to 0000h 1 : Set to FFFFh	RW
ACRC	CRC initialize bit	0 : CRC is not initialized 1 : CRC is initialized[2]	RW
– (b3)	Reserved bit	Set to 0	RW
RXSL	Receive source select bit	0 : ISRXD0 pin 1 : GiRI register	RW
TXSL	Transmit destination select bit	0 : ISTXD0 pin 1 : GiTO register	RW
CRC0	CRC generation polynomial select bits	b7 b6 0 0 : X^8+X^4+X+1 0 1 : Do not set to this value 1 0 : $X^{16}+X^{15}+X^2+1$ 1 1 : $X^{16}+X^{12}+X^5+1$	RW
CRC1			RW

NOTES:
1. Set to 00h except HDLC data processing mode.
2. CRC is initialized when the GiDR register matches the GiCMP3 register.

Group i SI/O Expansion Transmit Control Register (i=0, 1)[1]

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

Symbol	Address	After Reset
G0ETC	00FFh	0000 0XXXb
G1ETC	013Fh	0000 0XXXb

Bit Symbol	Bit Name	Function	RW
– (b3-b0)	Reserved bits	Set to 0	RW
TCRCE	Transmit CRC enable bit	0 : Not used 1 : Used	RW
– (b6-b5)	Reserved bits	Set to 0	RW
TBSF1	Transmit bit stuffing "0" insert select bit	0 : "0" is not inserted 1 : "0" is inserted	RW

NOTE:
1. Set to 00h except HDLC data processing mode.

Figure 22.41 G0EMR, G1EMR, G0ETC, G1ETC Registers

Group i SI/O Expansion Receive Control Register (i=0,1)(1)

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol	Address	After Reset
G0ERC	00FDh	00h
G1ERC	013Dh	00h

Bit Symbol	Bit Name	Function	RW
CMP0E	Data compare function 0 select bit	0 : The GiDR register (receive data register) is not compared with the GiCMP0 register 1 : The GiDR register is compared with the GiCMP0 register	RW
CMP1E	Data compare function 1 select bit	0 : The GiDR register (receive data register) is not compared with the GiCMP1 register 1 : The GiDR register is compared with the GiCMP1 register	RW
CMP2E	Data compare function 2 select bit	0 : The GiDR register (receive data register) is not compared with the GiCMP2 register 1 : The GiDR register is compared with the GiCMP2 register	RW
CMP3E	Data compare function 3 select bit	0 : The GiDR register (receive data register) is not compared with the GiCMP3 register 1 : The GiDR register is compared with the GiCMP3 register(2)	RW
RCRCE	Receive CRC enable bit	0 : Not used 1 : Used	RW
RSHTE	Receive shift operation enable bit	0 : Receive shift operation disabled 1 : Receive shift operation enabled	RW
– (b6)	Reserved bit	Set to 0	RW
RBSF1	Receive bit stuffing "0" delete select bit	0 : "0" is not deleted 1 : "0" is deleted	RW

NOTES:
1. The GiERC register is used in HDLC data processing mode.
   Set the GiERC register to 0010 0000b in clock synchronous mode, set it to 00h in UART mode.
2. When the ACRC bit in the GiEMR register is set to 1 (CRC initialized), set the CMP3E bit to 1.

Figure 22.42 G0ERC, G1ERC Registers

Group i SI/O Special Communication Interrupt Determination Register (i=0,1)(1)(2)

b7 b6 b5 b4 b3 b2 b1 b0
(b3–b0: 0 0 0 0)

Symbol	Address	After Reset
G0IRF	00FEh	0000 XXXXb
G1IRF	013Eh	0000 XXXXb

Bit Symbol	Bit Name	Function	RW
– (b3-b0)	Reserved bits	Set to 0	RW
IRF0	Interrupt source determination flag 0	0 : The GiDR register (receive data register) does not match the GiCMP0 register 1 : The GiDR register matches the GiCMP0 register	RW
IRF1	Interrupt source determination flag 1	0 : The GiDR register (receive data register) does not match the G0CMP1 register 1 : The GiDR register matches the GiCMP1 register	RW
IRF2	Interrupt source determination flag 2	0 : The GiDR register (receive data register) does not match the GiCMP2 register 1 : The GiDR register matches the GiCMP2 register	RW
IRF3	Interrupt source determination flag 3	0 : The GiDR register (receive data register) does not match the GiCMP3 register 1 : The GiDR register matches the GiCMP3 register	RW

NOTES:
1. Set to 00b except in HDLC data processing mode.
2. The SRTiR bit in the IIO4IR register is set to 1, if any of bits IRF3 to IRF0 is set to 1.

Figure 22.43 G0IRF and G1IRF Registers

Group i Data Comparison Register j (i=0,1; j=0 to 3)(1)

b7 b0

Symbol	Address	After Reset
G0CMP0 to G0CMP3	00F0h, 00F1h, 00F2h, 00F3h	Undefined
G1CMP0 to G1CMP3	0130h, 0131h, 0132h, 0133h	Undefined

Function	Setting Range	RW
Data to be compared	00h to FFh	RW

NOTE:
1. Set the GiMSK0 register to use the GiCMP0 register.
   Set the GiMSK1 register to use the GiCMP1 register.

Group i Data Mask Register j (i=0,1; j=0,1)

b7 b0

Symbol	Address	After Reset
G0MSK0, G0MSK1	00F4h, 00F5h	Undefined
G1MSK0, G1MSK1	0134h, 0135h	Undefined

Function	Setting Range	RW
Masked data for received data Write 1 to the bit which is not compared	00h to FFh	RW

Group i Transmit CRC Code Register (i=0,1)

b15 b8 b7 b0

Symbol	Address	After Reset
G0TCRC, G1TCRC	00FBh-00FAh, 013Bh-013Ah	0000h

Function	RW
Result of the transmit CRC calculation(1)(2)	RO

NOTES:
1. This register becomes the initial value selected by the CRCV bit in the GiEMR register when the TE bit in the GiCR register is set to 0 (transmit operation disabled).
2. Transmit CRC calculation is performed when each one bit of data is transmitted while the TCRCE bit in the GiETC register is set to 1 (used).

Group i Receive CRC Code Register (i=0,1)

b15 b8 b7 b0

Symbol	Address	After Reset
G0RCRC, G1RCRC	00F9h-00F8h, 0139h-0138h	Undefined

Function	RW
Result of the receive CRC calculation(1)(2)(3)	RO

NOTES:
1. This register becomes the initial value selected by the CRCV bit in the GiEMR register when the RCRCE bit in the GiERC register is set to 0 (not used). If the ACRC bit in the GiEMRj (j = 0 to 3) register is set to 1 (initialized), this register is initialized when the received data is matched the data in the GiCMPj register.
2. This register is initialized before receive operation starts.
3. Receive CRC calculation is performed when each one bit of data is received while the RCRCE bit in the GiERC register is set to 1 (used).

Figure 22.44 G0CMP0 to G0CMP3, G1CMP0 to G1CMP3 Registers, G0MSK0 and G0MSK1 Registers, G1MSK0 and G1MSK1 Registers, G0TCRC and G1TCRC Registers, G0RCRC and G1RCRC Registers

Group i Transmit Buffer (Receive Data) Register (i=0,1)

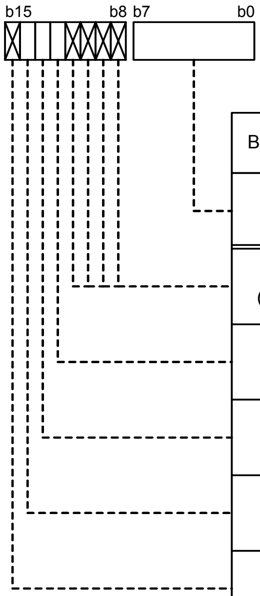
b7 b0

Symbol	Address	After Reset
G0TB, G0DR	00EAh	Undefined
G1TB, G1DR	012Ah	Undefined

Function	RW
GiTB The transmit data is set in the GiTB register by writing to this address. Set data to be transmitted. GiDR The receive data in the GiDR register is returned by reading this address. In HDLC data processing mode, the value set in the GiRI register is shifted to the GiDR register by bits.	RW

Figure 22.45 G0TB, G1TB Registers, G0DR, G1DR Registers

Group i SI/O Receive Buffer Register i (i=0, 1)

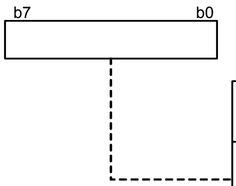


Symbol	Address	After Reset
G0RB	00E9h - 00E8h	XXX0 XXXX XXXX XXXXb
G1RB	0129h - 0128h	X000 XXXX XXXX XXXXb

Bit Symbol	Bit Name	Function	RW
– (b7-b0)	–	Received data	RW
– (b11-b8)	Unimplemented. Write 0. Read as undefined value.		–
OER	Overrun error flag(2)	0 : No overrun error 1 : Overrun error detected	RO
FER	Framing error flag(1)(2)	0 : No framing error 1 : Framing error detected	RO
PER	Parity error flag(1)(2)	0 : No parity error 1 : Parity error detected	RO
– (b15)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Nothing is implemented in bits FER and PER in the G0RB register. A read from these bits returns undefined value.
2. Each error flag is updated when the data is transferred from the receive shift register to the GiRB register every time a receive operation is completed.

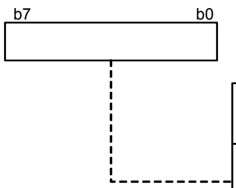
Group i Receive Input Register (i=0,1)



Symbol	Address	After Reset
G0RI, G1RI	00ECh, 012Ch	Undefined

Function	Setting Range	RW
Write data to be set to a receive data generation circuit	00h to FFh	WO

Group i Transmit Output Register (i=0,1)



Symbol	Address	After Reset
G0TO, G1TO	00EEh, 012Eh	Undefined

Function	RW
Read data output from a transmit data generation circuit	RO

Figure 22.46 G0RB, G1RB Registers, G0RI, G1RI Registers, G0TO, G1TO Registers

22.4.1 Clock Synchronous Mode (Groups 0 and 1)

Full-duplex clock synchronous serial communication is allowed in this mode. f8, f2n, or external clock can be selected as the group 0 serial clock. f8, f2n, the clock generated in channel 3, or external clock can be selected as the group 1 serial clock. Table 22.14 lists specifications of groups 0 and 1 clock synchronous mode. Table 22.15 and 22.16 list clock settings. Table 22.17 lists pin settings. Figures 22.47 to 22.49 show register setting. Figure 22.50 shows an example of a transmit and receive operation.

Table 22.14 Clock Synchronous Mode Specifications (Groups 0 and 1)

Item	Specification
Data format	Data length: 8 bits long
Serial clock	Refer to the Tables 22.15 and 22.16
Transmit and receive start condition	Select serial clock and set registers GiMR and GiERC (i = 0, 1). Then wait for one or more serial clock cycles before all of the following conditions are met to start the transmit/receive operation. •The TE bit in the GiCR register is set to 1 (transmit operation enabled) •The TI bit in the GiCR register is set to 0 (data in the GiTB register) •The RE bit in the GiCR register is set to 1 (receive operation enabled) If transmit-only operation is performed, the RE bit setting is not required.
Interrupt request generation timing	Transmit interrupt (The IRS bit in the GiMR register selects one of the following) •When IRS is set to 0 (no data in the GiTB register): When data is transferred from the GiTB register to the transmit shift register (transmit operation started) •When IRS is set to 1 (transmit operation completed): When data transmit operation from the transmit shift register is completed The SIOiTR bit in IIO1IR or IIO3IR register becomes 1 (interrupt requested) when a transmit interrupt request is generated (Refer to Figure 11.18). Receive interrupt •When data is transferred from the receive shift register to the GiRB register (receive operation completed) The SIOiRR bit in IIO1IR or IIO2IR register becomes 1 (interrupt requested) when a receive interrupt request is generated (Refer to Figure 11.18).
Error detection	•Overrun error Overrun error occurs when the 7th bit of the next data is received before reading the GiRB register. If an overrun error occurs, a read from the GiRB register returns an undefined value. The OER bit is updated when the data is transferred from the receive shift register to the GiRB register every time a receive operation is completed.
Selectable function	• LSB first or MSB first Data is transmitted and received from either bit 0 or bit 7. • ISTXDi and ISRXDi I/O polarity invert The level output from the ISTXDi pin and the level applied to the ISRXDi pin are inverted.

Table 22.15 Clock Settings (Group 0)

Serial Clock	G0MR Register	CCS Register
	CKDIR Bit	Bits CCS1 and CCS0
f8	0	11b
f2(1)	0	10b
Input to ISCLK0 pin	1	–

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

Table 22.16 Clock Settings (Group 1)

Serial Clock[3]	G1MR Register CKDIR Bit	CCS Register Bits CCS3 and CCS2
$\frac{fBT1}{2(n+2)}$ (NOTE 1)	0	00b
f8	0	11b
f2n[2]	0	10b
Input to ISCLK1 pin	1	–

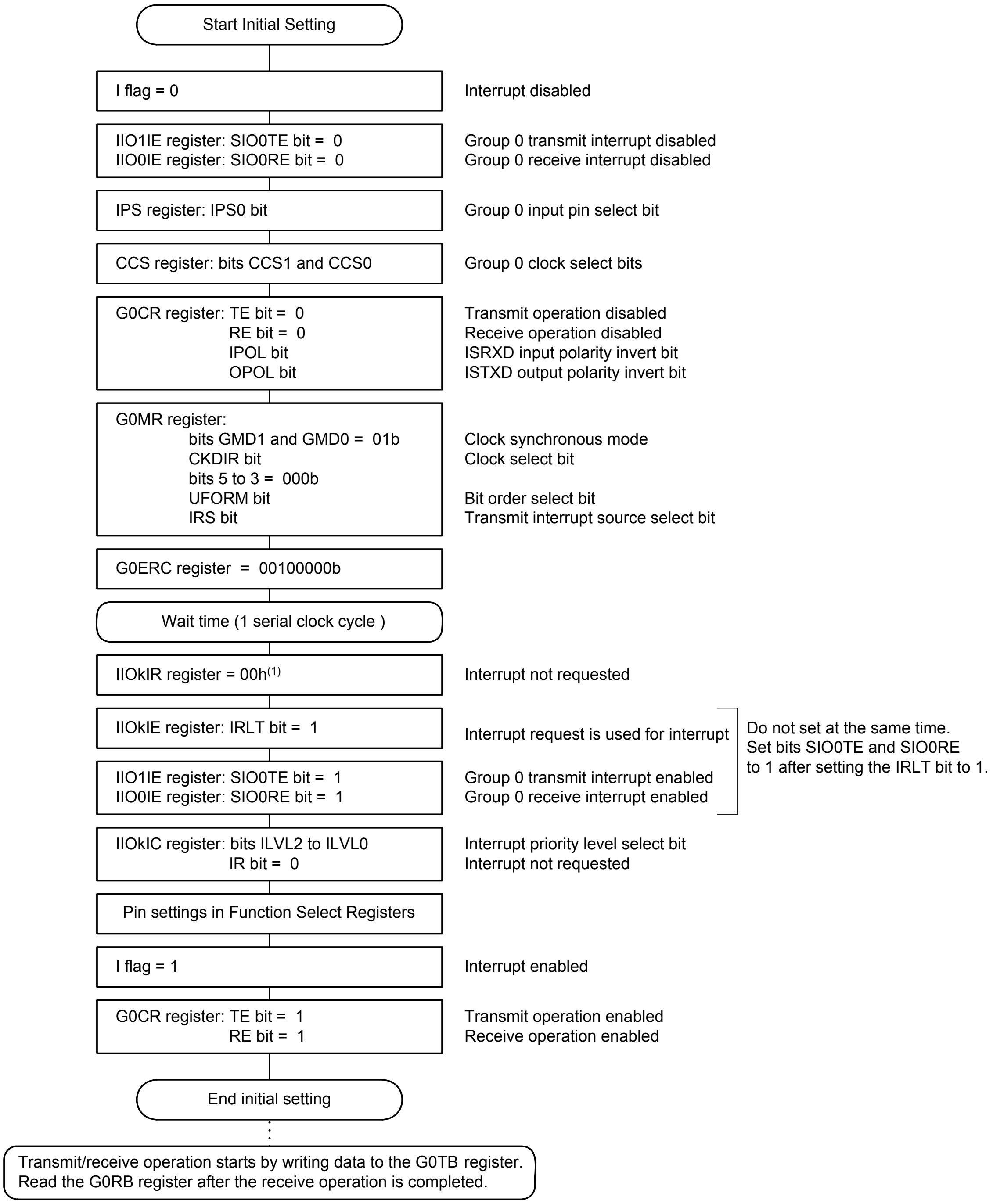
n: Setting value of the G1PO0 register (0001h to FFFDh)

NOTES:
1. The serial clock is generated in phase-delayed waveform output mode of the channel 3. The baud rate is set using the function, which is to reset a base timer when the value in the G1PO0 register matches the value of a base timer.
2. Bits CNT3 to CNT0 in the TCSPR register select no division (n=0) or divide-by-2n (n=1 to 15).
3. The serial clock is set to fBT1 divided by six or lower frequency. Additionally, meet the timing requirements, which are shown on **Tables 27.25 and 27.48 Intelligent I/O communication function (Groups 0 and 1)** in the chapter **27. Electrical Characteristics**.

Table 22.17 Pin Settings in Clock Synchronous Mode (Groups 0 and 1)

Port	Function	G1POCR0 G1POCR1 Registers[2]	Bit Setting					
			IPS Register	PD7, PD8, PD11,PD15 Registers	PSD1 Register	PSC Register	PSL1, PSL5, PSL9 Registers	PS1, PS2, PS5, PS9 Registers[1]
P7_3	ISTXD1 Output[3]	G1POCR0	–	–	–	PSC_3=1	PSL1_3=0	PS1_3=1
P7_4	ISCLK1 Input	–	IPS1=0	PD7_4=0	–	–	–	PS1_4=0
	ISCLK1 Output	G1POCR1	–	–	PSD1_4=0	PSC_4=1	PSL1_4=0	PS1_4=1
P7_5	ISRXD1 Input	–	IPS1=0	PD7_5=0	–	–	–	PS1_5=0
P7_6	ISTXD0 Output[3]	–	–	–	PSD1_6=0	PSC_6=0	PSL1_6=0	PS1_6=1
P7_7	ISCLK0 Input	–	IPS0=0	PD7_7=0	–	–	–	PS1_7=0
	ISCLK0 Output	–	–	–	–	–	PSL1_7=0	PS1_7=1
P8_0	ISRXD0 Input	–	IPS0=0	PD8_0=0	–	–	–	PS2_0=0
P11_0	ISTXD1 Output[3]	G1POCR0	–	–	–	–	PSL5_0=0	PS5_0=1
P11_1	ISCLK1 Input	–	IPS1=1	PD11_1=0	–	–	–	PS5_1=0
	ISCLK1 Output	G1POCR1	–	–	–	–	PSL5_1=0	PS5_1=1
P11_2	ISRXD1 Input	–	IPS1=1	PD11_2=0	–	–	–	PS5_2=0
P15_0	ISTXD0 Output[3]	–	–	–	–	–	PSL9_0=0	PS9_0=1
P15_1	ISCLK0 Input	–	IPS0=1	PD15_1=0	–	–	–	PS9_1=0
	ISCLK0 Output	–	–	–	–	–	PSL9_1=0	PS9_1=1
P15_2	ISRXD0 Input	–	IPS0=1	PD15_2=0	–	–	–	–

NOTES:
1. Set registers PS1, PS2, PS5, and PS9 after setting the other registers.
2. Set bits MOD2 to MOD0 in the corresponding register to 111b (use communication function output).
3. After an operating mode is selected in the GiMR register and the pin function is set in the Function Select Registers, the ISTXDi pin outputs an "H" signal when the OPOL bit is set to 0 (No ISTXD output polarity invert) or the ISTXDi pin outputs an "L" signal when the OPOL bit is set to 1 (ISTXD output polarity invert) until a transmit operation starts.



k = 0,1

NOTE:
1. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated in the same register (Interrupt does not occur).

Figure 22.47 Register Settings in Group 0 Clock Synchronous Mode

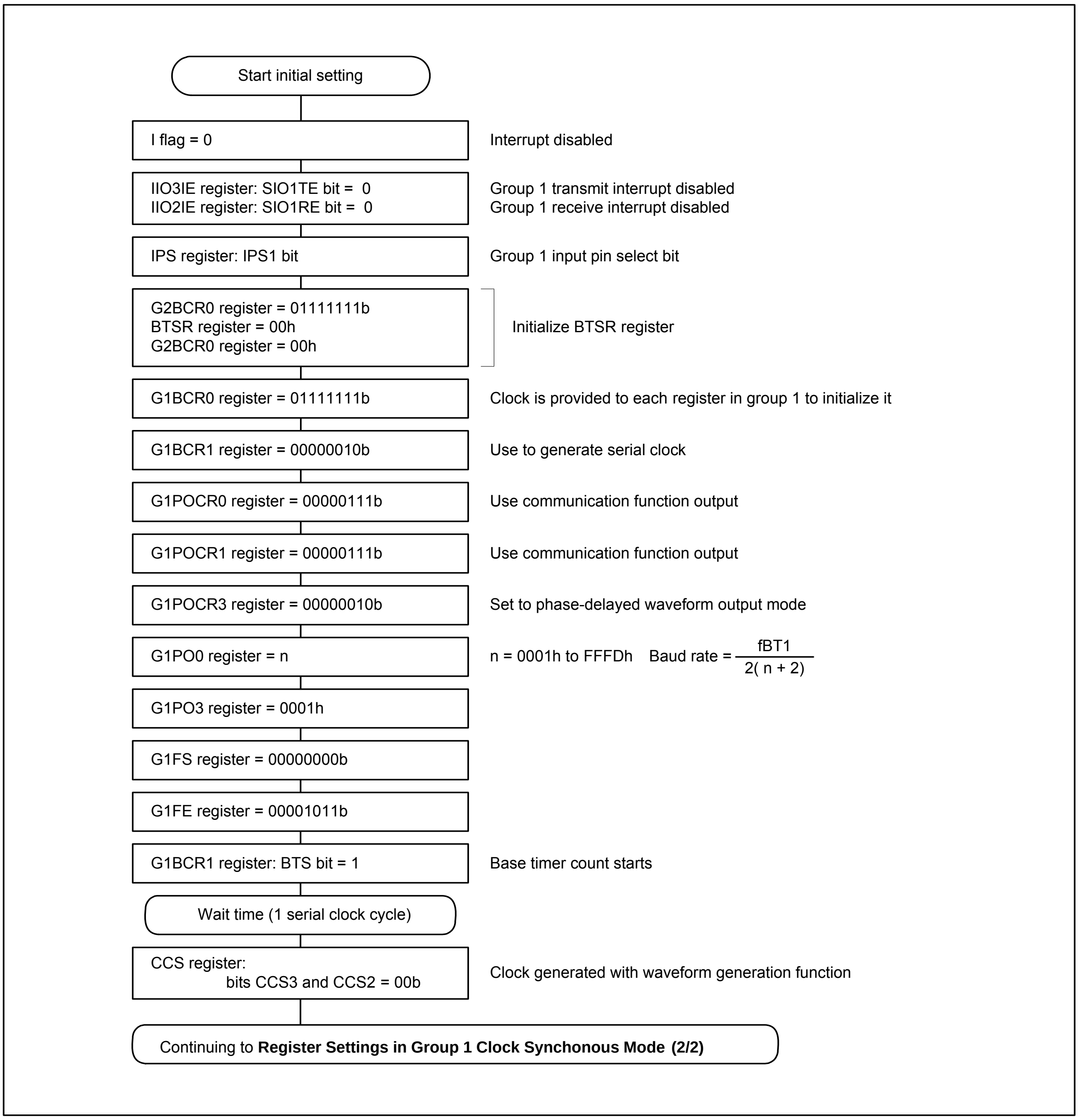
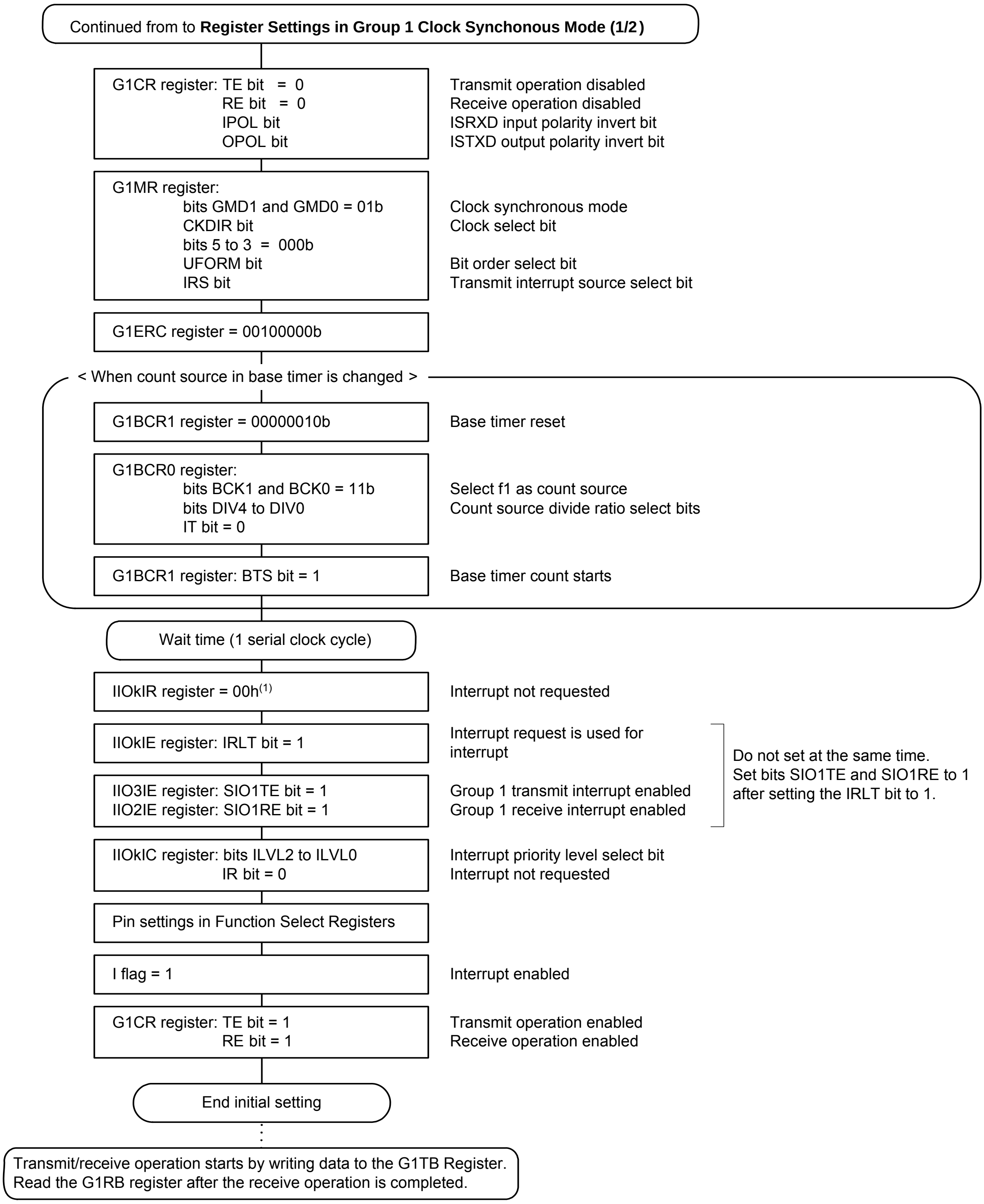


Figure 22.48 Register Settings in Group 1 Clock Synchronous Mode (1/2)

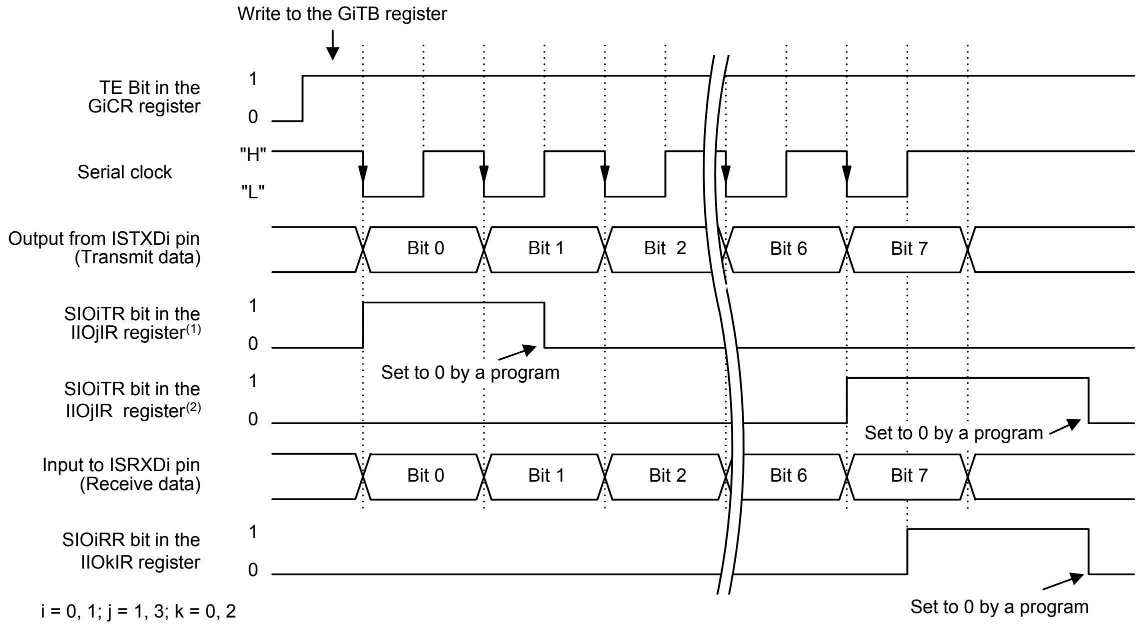


k = 2, 3

NOTE:
1. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated in the same register (Interrupt does not occur).

Figure 22.49 Register Settings in Group 1 Clock Synchronous Mode (2/2)

(1) When f8, f2n or External Clock is Selected as the Serial Clock (Groups 0 and 1)

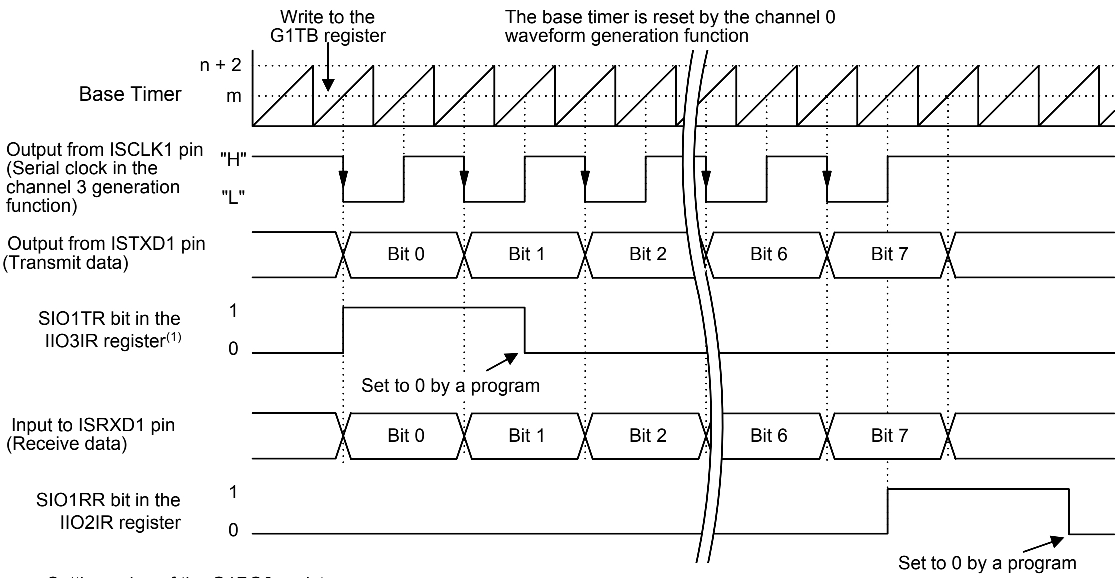


i = 0, 1; j = 1, 3; k = 0, 2

The above applies under the following conditions:
- Bits CCS1 and CCS0 or bits CCS3 and CCS2 in the CCS register are set to 10b or 11b
- The UFORM bit in the GiMR register is set to 0 (LSB first)
- Bits IPOL and OPOL in the GiCR register are set to 0 (not inverted)

NOTES:
1. This applies when IRS bit in the GiMR register is set to 0 (No data in the GiTB register).
2. This applies when IRS bit in the GiMR register is set to 1 (Transmit operation completed).

(2) When the Serial Clock is Generated in Channel 3 Phase-Delayed Waveform Output Mode (Group 1)



n: Setting value of the G1PO0 register
m: Setting value of the G1PO3 register

The above applies under the following conditions:
- In the G1MR register, the CKDIR bit is set to 0 (internal clock), the UFORM bit is set to 0 (LSB first)
- Bits CCS3 and CCS2 in the CCS register are set to 00b (Clock generated with waveform generation function)
- Bits IPOL and OPOL in the G1CR register are set to 0 (not inverted)

NOTE:
1. This applies when the IRS bit in the G1MR register is set to 0 (No data in the G1TB register).

Figure 22.50 Transmit and Receive Operation in Clock Synchronous Mode (Groups 0 and 1)

22.4.2 Clock Asynchronous (UART) Mode (Group 1)

Table 22.18 lists specifications of UART mode. Table 22.19 lists pin settings. Figures 22.51 and 22.52 show register settings. Figure 22.53 shows an example of a transmit operation. Figure 22.54 shows an example of a receive operation.

Table 22.18 UART Mode Specifications

Item	Specification
Data format	• Data length: 8 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	$\frac{fBT1}{2(n + 2)}$ n: Setting value of the G1PO0 register (0006h to FFFDh) • The CKDIR bit in the G1MR register is set to 0 (internal clock) • Bits CCS3 and CCS2 in the CCS register is set to 00b (Clock generated with waveform generation function) The internal transmit clock is generated in phase-delayed waveform output mode of the channel 3. The internal receive clock is generated by performing both the time measurement and phase-delayed waveform output in the channel 2.
Transmit start condition	Set registers associated with the waveform generation function and the G1MR register. Then wait for one or more internal transmit clock cycles before all of the following conditions are met to start the transmit operation. •The TE bit in the G1CR register is set to 1 (transmit operation enabled) •The TI bit in the G1CR register is 0 (data in the G1TB register)
Receive start condition	Set registers associated with the waveform generation function and the G1MR register. Then wait for one or more internal receive clock cycles before all of the following conditions are met to start the receive operation. •The RE bit in the G1CR register is set to 1 (receive operation enabled) •Detecting the start bit (“L” level)
Interrupt request generation timing	Transmit interrupt (The IRS bit in the G1MR register selects one of the following): •When the IRS bit is set to 0 (no data in the GiTB register): When data is transferred from the G1TB register to the transmit shift register (transmit operation started) •When the IRS bit is set to 1 (transmit operation completed): When the final stop bit is output from the transmit shift register The SIO1TR bit in the IIO3IR register becomes 1 (interrupt requested) when a transmit interrupt request is generated (Refer to Figure 11.18). Receive interrupt: •When data is transferred from the receive shift register to the G1RB register (receive operation completed) The SIO1RR bit in the IIO2IR register becomes 1 (interrupt requested) when a receive interrupt request is generated (Refer to Figure 11.18).
Error detection	• Overrun error 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 G1RB register. If an overrun error occurs, a read from the G1RB register returns an undefined value. • Framing error Framing error occurs when the number of the stop bits set by the STPS bit in the G1MR 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 G1MR register. Each error flag is updated when the data is transferred from the receive shift register to the G1RB register every time a receive operation is completed.
Selectable function	• LSB first or MSB first Data is transmitted or received from either bit 0 or bit 7.

Table 22.19 Pin Settings in UART Mode (Group 1)

Port	Function	G1POCR0 Register(2)	Bit Setting				
			IPS Register	PD7, PD11 Registers	PSC Register	PSL1, PSL5 Registers	PS1, PS5 Registers(1)
P7_3	ISTXD1 output	G1POCR0	–	–	PSC_3=1	PSL1_3=0	PS1_3=1
P7_5	ISRXD1 input	–	IPS1=0	PD7_5=0	–	–	PS1_5=0
P11_0	ISTXD1 output	G1POCR0	–	–	–	PSL5_0=0	PS5_0=1
P11_2	ISRXD1 input	–	IPS1=1	PD11_2=0	–	–	PS5_2=0

NOTES:
1. Set registers PS1 and PS5 after setting the other registers.
2. Set bits MOD2 to MOD0 in the G1POCR0 register to 111b (use communication function output).

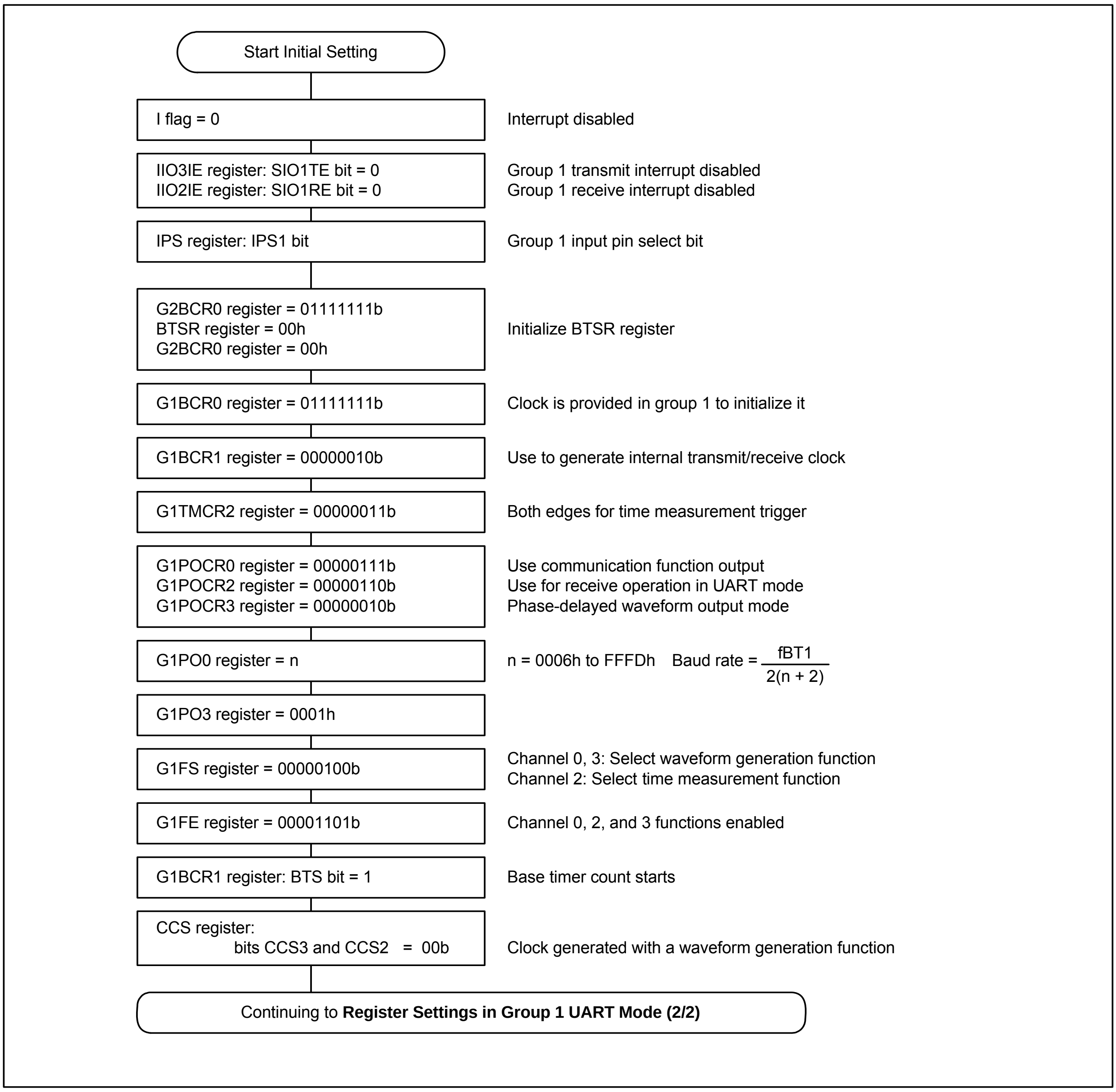
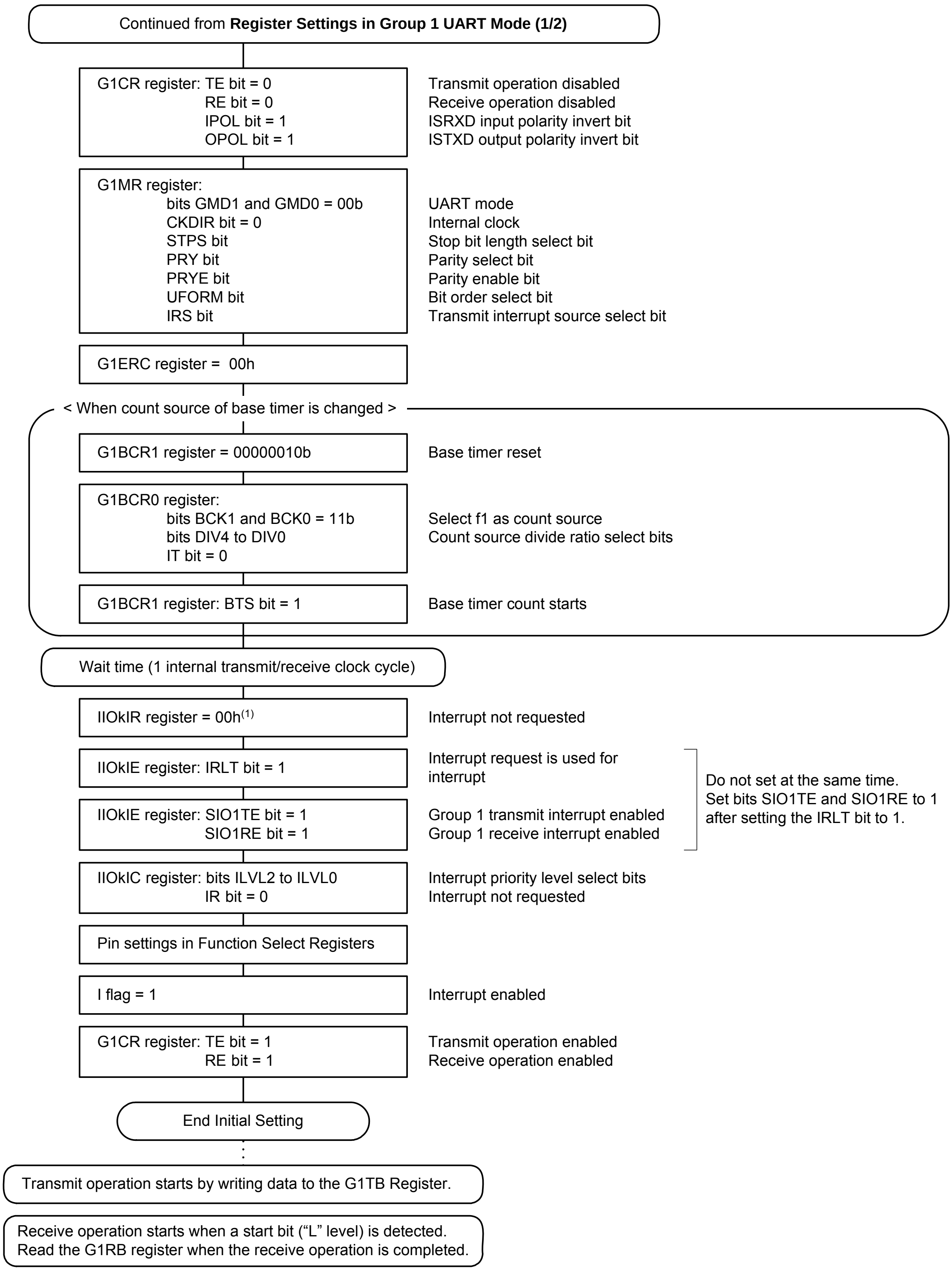


Figure 22.51 Register Settings in Group 1 UART Mode (1/2)

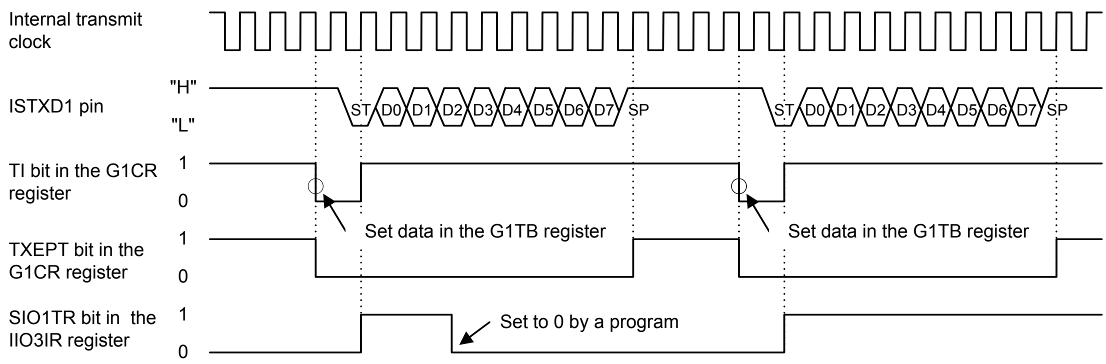


k = 2, 3

NOTE:

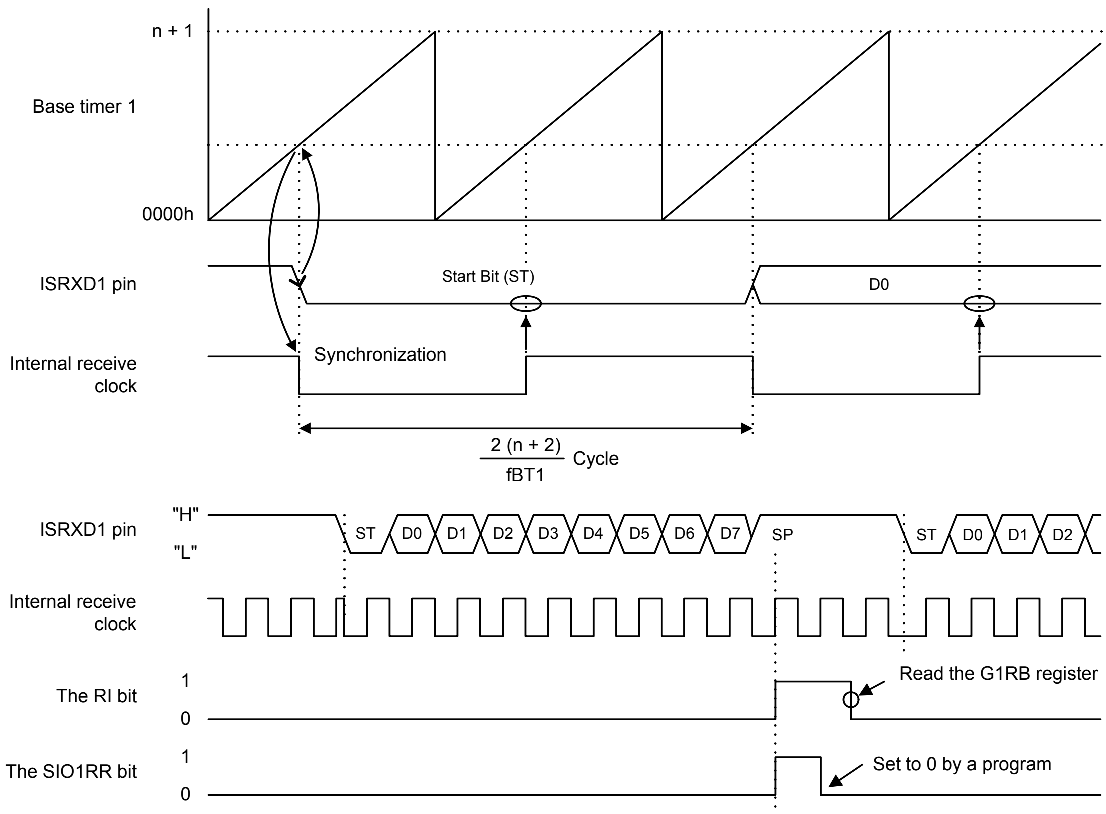
1. Set all the interrupt request flags to 0. If any of these flags remains 1, the IR bit in the IIOkIC register does not become 1 when an interrupt request is generated in the same register (Interrupt does not occur).

Figure 22.52 Register Settings in Group 1 UART Mode (2/2)



The above applies under the following conditions:
- The STPS bit in the G1MR register is set to 0 (1 stop bit)
- The PRYE bit in the G1MR register is set to 0 (parity disabled)
- The UFORM bit in the G1MR register is set to 0 (LSB first)
- The INV bit in registers G1POCR0 to G1POCR7 is set to 0 (output not inverted)
- The IRS bit in the G1MR register is set to 0 (no data in the G1TB register)

Figure 22.53 Transmit Operation in Group 1 UART Mode



The above applies under the following conditions:
- The STPS bit in the G1MR register is set to 0 (1 stop bit)
- The PRYE bit in the G1MR register is set to 0 (parity disabled)
- The UFORM bit in the G1MR register is set to 0 (LSB first)
- The INV bit in registers G1POCR0 to G1POCR7 is set to 0 (output not inverted)

SIO1RR bit: The bit in the IIO2IR register
RI bit: The bit in the G1CR register

Figure 22.54 Receive Operation in Group 1 UART Mode

22.4.3 HDLC Data Processing Mode (Group 0 and Group 1)

In HDLC data processing mode, bit stuffing, flag sequence detection, abort sequence detection and CRC calculation are available for HDLC data processing. In this mode, the MCU is unable to input or output data in no-return-to-zero-invert (NRZI) format (No pin is used). f1, f8 or f2n can be selected as the group 0 transfer clock. f1, f8, f2n or the clock generated in the channel 0 or 1 can be selected as the group 1 transfer clock.
To generate HDLC frame data, write source data to the GiTB register (i=0,1). The data conversion result is stored into the GiTO register. If data is in the GiTO register, the conversion is stopped. The conversion is resumed by reading the GiTO register. The HDLC data processing is performed even no data in the GiTB register. A CRC value is calculated every time one bit is converted.
To generate source data, write HDLC frame data to the GiRI register. The data in the GiRI register is transferred to the shift register. HDLC data processing starts when the value in the shift register matches the value in the GiCMP3 register (7Eh). The data conversion result is stored into the GiRB register.
Tables 22.20 and 22.21 list specifications of the HDLC data processing mode. Tables 22.22 and 22.23 list clock settings. Table 22.24 lists register settings.

Table 22.20 Specifications of the HDLC Data Processing Mode (1/2)

Item	Specification
Input data format	8-bit data fixed, bit alignment is optional
Output data format	8-bit data fixed
Transfer clock	See Tables 22.22 and 22.23
I/O method	• When HDLC frame data is generated from source data: A value set in the GiTB register (i=0,1) is converted with HDLC data processing and transferred to the GiTO register. • When source data is generated from HDLC frame data: A value set in the GiRI register is converted with HDLC data processing and transferred to the GiRB register.
Bit stuffing	When HDLC frame data is generated, a “0” is inserted after five continuous “1’s”. When source data is generated, a “0” is deleted after five continuous “1’s”.
Flag sequence detection	Write the flag sequence “7Eh” to the GiCMP3 register. When the GiDR register matches the GiCMP3 register, a special communication function interrupt is generated. (The SRTiR bit in the IIO4IR register becomes 1.)
Abort sequence detection	Write the abort sequence “FEh” to the GiCMPj register (j = 0, 1) and the masked data “01h” to the GiMSKj register. When the GiDR register and the GiCMPj register are compared and all the non-masked bits are matched, a special communication function interrupt is generated. (The SRTiR bit in the IIO4IR register becomes 1.)
CRC	Bits CRC1 and CRC0 are set to 11b ($X^{16}+X^{12}+X^5+1$) The CRCV bit is set to 1 (set to FFFFh) • When HDLC frame data is generated: CRC calculation result is stored into the GiTCRC register. The TCRCE bit in the GiETC register is set to 1 (transmit CRC used). Initialization: The CRC calculation result is initialized when the TE bit in the GiCR register is set to 0 (transmit disabled). • When source data is generated: CRC calculation result is stored into the GiRCRC register. The RCRCE bit in the GiERC register is set to 1 (receive CRC used). Initialization: The CRC calculation result is initialized when the GiDR register matches the GiCMP3 register by comparing the flag sequence “7Eh” (The ACRC bit in the GiEMR register is set to 1 (CRC is initialized)).
Data processing start condition	The following conditions are required to start HDLC frame data generation: • The TE bit in the GiCR register is set to 1 (transmit operation enabled) • Data is written to the GiTB register The following conditions are required to start source data generation: • The RE bit in the GiCR register is set to 1 (receive operation enabled) • Data is written to the GiRI register

Table 22.21 Specifications of the HDLC Data Processing Mode (2/2)

Item	Specification
Interrupt request generation timing	When HDLC frame data is generated: • The IRS bit in the GiMR register selects one of the following: - When the IRS bit is set to 0 (no data in the GiTB register) When data is transferred from the GiTB register to the transmit shift register (transmit operation started). - When the IRS bit is set to 1 (transmit operation completed) When data transfer from the transmit shift register to the GiTO register is completed. When one of the above occurs, the GiTOR bit in the IIO1IR or IIO3IR register becomes 1 (interrupt requested) (Refer to Figure 11.18). • When data, which is already converted to HDLC frame data, is transferred from the transmit shift register of the GiTO register to the transmit buffer, the GiTOR bit becomes 1. When source data is generated: • When data is transferred from the GiRI register to the GiRB register (receive operation completed), the GiRIR bit in the IIO0IR or IIO2IR register becomes 1 (interrupt requested). • When receive data is transferred from the receive buffer in the GiRI register to the receive shift register, the GiRIR bit becomes 1. • When the GiTB register is compared to the GiCMPj register (j = 0 to 3), the SRTiR bit in the IIO4IR register becomes 1 (interrupt requested).

Table 22.22 Clock Settings in HDLC Data Processing Mode (Group 0)

Transfer Clock(1)	CCS Register	
	CCS0 Bit	CCS1 Bit
f1	1	0
f8	1	1
f2n(2)	0	1

NOTES:
1. The transfer clock is generated when the RSHTE bit in the G0ERC register is set to 1 (receive shift operation enabled) while source data is generated.
2. Bits CNT3 to CNT0 in the TCSPR register select no division (n = 0) or divide-by-2n (n = 1 to 15).

Table 22.23 Clock Setting in HDLC Data Processing Mode (Group 1)

Transfer Clock(1)	CCS Register	
	CCS2 Bit	CCS3 Bit
$\frac{fBT1}{m+2}$ (NOTE 2)	0	0
f1	1	0
f8	1	1
f2n(3)	0	1

m: Setting value of the G1PO0 register (0001h to FFFDh)
NOTES:
1. The transfer clock is generated when the RSHTE bit in the G1ERC register is set to 1(receive shift operation enabled) while source data is generated.
2. The transfer clock is generated in single-phase waveform output mode of the channel 1.
3. Bits CNT3 to CNT0 in the TCSPR register select no division (n=0) or divide-by-2n (n=1 to 15).

Table 22.24 Register Settings in HDLC Data Processing Mode (Groups 0 and 1)

Register	Bit	Function
CCS	CCS1 and CCS0	Select transfer clock.
	CCS3 and CCS2	Select transfer clock.
G1BCR0[(1)]	BCK1 and BCK0	Select count source.
	DIV4 to DIV0	Select count source divide ratio.
	IT	Select the base timer interrupt generation timing.
G1BCR1[(1)]	–	Set to 0001 0010b.
G1POCR0[(1)]	–	Set to 0000 0000b.
G1POCR1[(1)]	–	Set to 0000 0000b.
G1PO0[(1)]	–	Set baud rate.
G1PO1[(1)]	–	Set the timing of the rising edge of the transfer clock. Timing of the falling edge (“H” width of the transfer clock) is fixed. Setting value of the G1PO1 register ≤ setting value of the G1PO0 register
G1FS[(1)]	FSC1 and FSC0	Set to 00b.
G1FE[(1)]	IFE1 and IFE0	Set to 11b.
GiMR	GMD1 and GMD0	Set to 11b.
	CKDIR	Set to 0.
	UFORM	Set to 0.
	IRS	Select a transmit interrupt source.
GiCR	TE	Set to 1 to enable a transmit operation (HDLC frame data generation from source data).
	TXEPT	Transmit shift register empty flag
	TI	GiTB register empty flag
	RE	Set to 1 to enable a receive operation (source data generation from HDLC frame data).
	RI	Receive completion flag
GiEMR	–	Set to 1111 0110b.
GiETC	TCRCE	Set to 1 (CRC calculation is performed when HDLC frame data is generated from source data).
	TBSF1	Set to 1 (“0” is inserted when HDLC frame data is generated).
GiERC	CMP2E to CMP0E	Select whether or not the GiDR register and GiCMPj register (j = 0 to 2) are compared.
	CMP3E	Set to 1.
	RCRCE	Set to 1 (CRC calculation is performed when source data is generated from HDLC frame data).
	RSHTE	When source data is generated, set to 1.
	RBSF1	Set to 1 (“0” is deleted when source data is generated).
GiIRF	IRF3 to IRF0	Select an interrupt source.
GiCMP0 and GiCMP1	–	Write FEh to detect an abort sequence.
GiCMP2	–	Set data to be compared.
GiCMP3	–	Write 7Eh.
GiMSK0 and GiMSK1	–	Write 01h to detect an abort sequence.
GiTCRC	–	The CRC code, which is calculated when generating HDLC frame data from source data, can be read.
GiRCRC	–	The CRC code, which is calculated when generating source data from HDLC frame data, can be read.
G1TB	–	Used to generate HDLC frame data. Write source data.
GiTO	–	Used to generate HDLC frame data. HDLC frame data, which is generated from source data, can be read.
GiRI	–	Used to generate source data. Write HDLC frame data.
G1RB	–	Used to generate source data. Source data, which is generated from HDLC frame data, can be read.

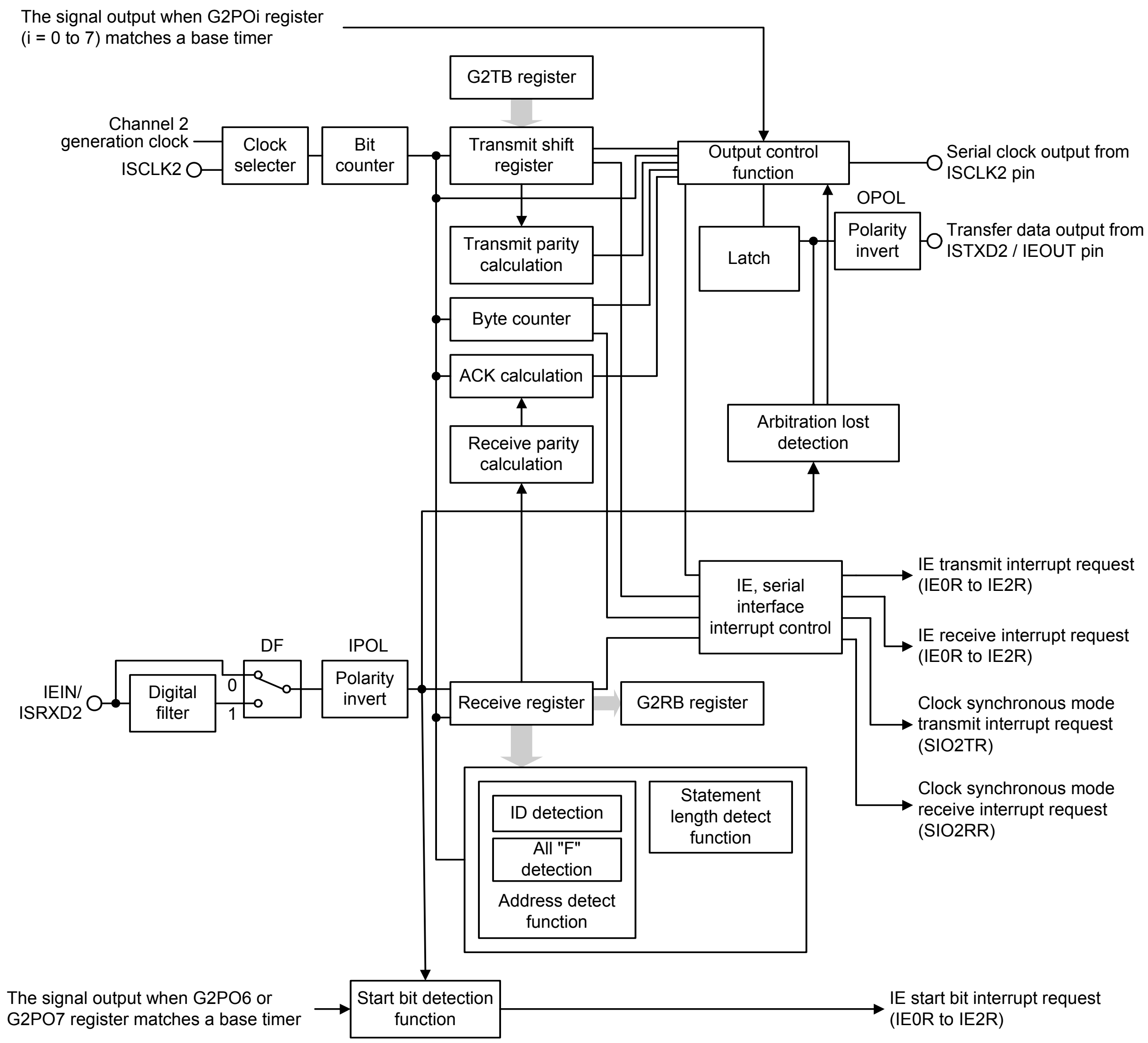
i = 0,1

NOTE:

1. These register settings are required when bits CCS3 and CCS2 in the CCS register are set to 00b (clock generated with the waveform generation function).

22.5 Group 2 Communication Function

In the group 2 communication function, variable data length clock synchronous serial communication is available. Figure 22.55 shows block diagram of group 2 communication function. Figures 22.56 to 22.60 show registers associated with the communication function.

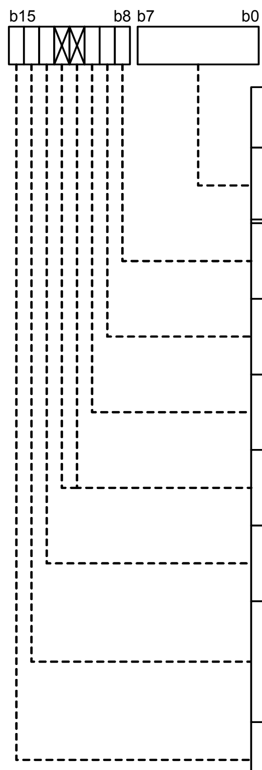


OPOL, IPOL: Bits in the G2CR register
DF: Bit in the IECR register
IE0R to IE2R: Bits in registers IIO7IR and IIO8IR
SIO2TR : Bit in the IIO6IR register
SIO2RR: Bit in the IIO5IR register

NOTE:
1. After a clock, which is selected in the G2BCR0 register, is supplied to the registers, each register value becomes the after reset value.

Figure 22.55 Group 2 Communication Function Block Diagram

Group 2 SI/O Transmit Buffer Register

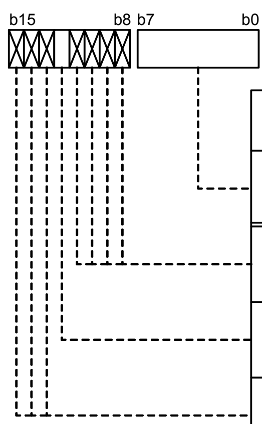


Symbol	Address	After Reset
G2TB	016Dh - 016Ch	Undefined

Bit Symbol	Bit Name	Function	RW
– (b7-b0)	Transmit buffer	Transmit Data	WO
SZ0	Data length select bits	b10 b9 b8 0 0 0 : 8 bits long 0 0 1 : 1 bits long 0 1 0 : 2 bits long 0 1 1 : 3 bits long 1 0 0 : 4 bits long 1 0 1 : 5 bits long 1 1 0 : 6 bits long 1 1 1 : 7 bits long	RW
SZ1			RW
SZ2			RW
– (b12-b11)	Unimplemented. Write 0. Read as undefined value.		–
A	ACK function select bit	0 : Adds no ACK bit 1 : Adds the ACK bit after last transmit bit	RW
PC	Parity calculation continuing bit	0 : Adds the parity bit after the transmit data 1 : Carries over a parity to the following transmit data(1)	RW
P	Parity function select bit	0 : No parity 1 : Parity (even parity only)	RW

NOTE:
1. Set the P bit to 0 before setting the PC bit to 1.

Group 2 SI/O Receive Buffer Register



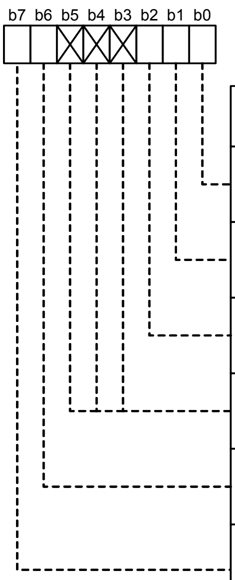
Symbol	Address	After Reset
G2RB	016Fh - 016Eh	Undefined

Bit Symbol	Bit Name	Function	RW
– (b7-b0)	Receive buffer	Receive data	RO
– (b11-b8)	Unimplemented. Write 0. Read as undefined value.		–
OER	Overrun error flag(1)	0 : No overrun error 1 : Overrun error detected	RO
– (b15-b13)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The OER bit becomes 0 when bits GMD1 and GMD0 in the G2MR register are set to 00b (communication unit is reset) or the RE bit in the G2CR register is set to 0 (receive operation disabled).

Figure 22.56 G2TB, G2RB Register

Group 2 SI/O Communication Mode Register



Symbol	Address	After Reset
G2MR	016Ah	00XX X000b

Bit Symbol	Bit Name	Function	RW
GMD0	Communication mode select bits	b1 b0 0 0 : Communication unit is reset (The OER bit becomes 0)[1] 0 1 : Clock synchronous mode[2] 1 0 : IE mode[2] 1 1 : Do not set to this value	RW
GMD1			RW
CKDIR	Clock select bit	0 : Internal clock 1 : External clock	RW
– (b5-b3)	Unimplemented. Write 0. Read as undefined value.		–
UFORM	Bit order select bit	0 : LSB first 1 : MSB first	RW
IRS	Transmit interrupt source select bit	0 : No data in the G2TB register (TI = 1) 1 : Transmit operation completed (TXEPT = 1)	RW

NOTES:
1. When changing mode, set bits GMD1 and GMD0 to 00b (communication unit is reset) and wait for one or more fBT2 clock cycles before setting to different mode.
2. Set bits GMD1 and GMD0 to 01b (clock synchronous mode) or 10b (IE mode) while fBT2 is stopped.

Figure 22.57 G2MR Register

Group 2 SI/O Communication Control Register

b7 b6 b5 b4 b3 b2 b1 b0

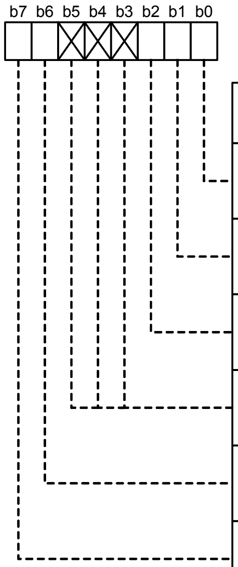
Symbol	Address	After Reset
G2CR	016Bh	0000 X000b

Bit Symbol	Bit Name	Function	RW
TE	Transmit operation enable bit	0 : Transmit operation disabled 1 : Transmit operation enabled	RW
TXEPT	Transmit shift register empty flag	0 : Data is in the transmit shift register (during transmit operation) 1 : No data is in the transmit shift register (transmit operation is completed)	RO
TI	G2TB register empty flag	0 : Data is in the G2TB register 1 : No data is in the G2TB register	RO
– (b3)	Unimplemented. Write 0. Read as undefined value.		–
RE	Receive operation enable bit(1)	0 : Receive operation disabled 1 : Receive operation enabled	RW
RI	Receive complete flag	0 : No data is in the G2RB register 1 : Data is in the G2RB register	RO
OPOL	ISTXD output polarity invert bit	0 : Not inverted 1 : Inverted	RW
IPOL	ISRXD input polarity invert bit(1)	0 : Not inverted 1 : Inverted	RW

NOTE:
1. The group 2 base timer may be reset when the RE or IPOL bit setting is changed. To avoid resetting, set the RST2 bit in the G2BCR1 register to 0 (base timer is not reset by a reset request from the communication function).

Figure 22.58 G2CR Register

Group 2 IEBus Control Register

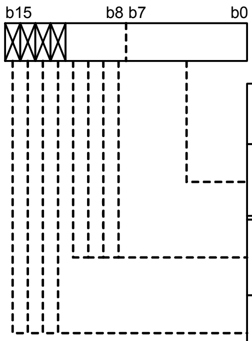


Symbol	Address	After Reset
IECR	0172h	00XX X000b

Bit Symbol	Bit Name	Function	RW
IEB	IEBus enable bit(1)	0 : IEBus disabled(2) 1 : IEBus enabled	RW
IETS	IEBus transmit operation start request bit	0 : Transmit operation completed 1 : Transmit operation started	RW
IEBBS	IEBus busy flag	0 : Idle state 1 : Busy state (start condition detected)	RO
– (b5-b3)	Unimplemented. Write 0. Read as undefined value.		–
DF	Digital filter select bit	0 : No digital filter 1 : Digital filter	RW
IEM	IEBus mode select bit	0 : Mode 1 1 : Mode 2	RW

NOTES:
1. Change the IEB bit setting while fBT2 is stopped.
2. When the IEB bit is set to 0, maintain the value for one or more fBT2 clock cycles. Set bits BCK1 and BCK0 in the G2BCR0 register to 00b (clock stop) when the IEB bit is set back to 1.

Group 2 IEBus Address Register



Symbol	Address	After Reset
IEAR	0171h - 0170h	Undefined

Function	RW
Address data	RW
Address data	RW
Unimplemented. Write 0. Read as undefined value.	–

Figure 22.59 IECR and IEAR Registers

Group 2 IEBus Transmit Interrupt Source Detect Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
IETIF	0173h	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
IETNF	Normal complete flag(1)	0 : Transmit operation is completed in error 1 : Transmit operation is successfully completed	RW
IEACK	ACK error flag(1)	0 : No error detected 1 : Error detected	RW
IETMB	Maximum transfer byte error flag(1)	0 : No error detected 1 : Error detected	RW
IETT	Timing error flag(1)	0 : No error detected 1 : Error detected	RW
IEABL	Arbitration lost flag(1)	0 : No error detected 1 : Error detected	RW
– (b7-b5)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. This bit can be set to 0 by a program, but cannot be set to 1. When the IEB bit in the IECR register is set to 0 (IEBus disabled), bits IETNF, IEACK, IETMB, IETT, and IEABL become 0.

Group 2 IEBus Receive Interrupt Source Detect Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
IERIF	0174h	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
IERNF	Normal completed flag(1)	0 : Receive operation is completed in error 1 : Receive operation is successfully completed	RW
IEPAR	Parity error flag(1)	0 : No error detected 1 : Error detected	RW
IERMB	Max. transfer byte error flag(1)	0 : No error detected 1 : Error detected	RW
IERT	Timing error flag(1)	0 : No error detected 1 : Error detected	RW
IERETC	Other source receive completed flag(1)	0 : No error detected 1 : Error detected	RW
– (b7-b5)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. This bit can be set to 0 by a program, but cannot be set to 1. When the IEB bit in the IECR register is set to 0 (IEBus disabled), bits IETNF, IEACK, IETMB, IETT, and IEABL become 0.

Figure 22.60 IETIF and IERIF Registers

22.5.1 Variable Data Length Clock Synchronous Mode (Group 2)

In variable data length clock synchronous mode, full-duplex clock synchronous serial communication is allowed. Transmit data can be selected from 1 to 8 bits long. Continuous transmit/receive operations enable to communicate more than 9 bit-long data. Table 22.25 lists specifications of the group 2 variable data length clock synchronous mode. Table 22.26 lists register settings. Table 22.27 lists pin settings. Figure 22.61 shows an example of a transmit and receive operation.

Table 22.25 Variable Data Length Clock Synchronous Mode Specifications (Group 2)

Item	Specification
Data format	Data length: variable
Serial clock(1)	When the CKDIR bit in the G2MR register is set to 0 (internal clock): $\frac{fBT2}{2(n+2)}$ n: setting value of the G2PO0 register (0001h to FFFDh) The G2PO0 register determines a baud rate and the serial clock is generated in phase-delayed waveform output mode of the channel 2. When the CKDIR bit is set to 1 (external clock): The serial clock is input from the ISCLK2 pin.
Transmit start condition	Transmit operation starts when all of the following conditions are met: • Set the TE bit in the G2CR register to 1 (transmit operation enabled) • Data is written to the G2TB register
Receive start condition	Receive operation starts when all of the following conditions are met: • Set the TE bit in the G2CR register to 1 (transmit operation enabled) • Data is written to the G2TB register • Set the RE bit in the G2CR register to 1 (receive operation enabled)
Interrupt request generation timing	Transmit interrupt (The IRS bit in the G2MR register selects one of the following): • The IRS bit is set to 0 (no data in the G2TB register): When data is transferred from the G2TB register to the transmit shift register (transmit operation started). • The IRS bit is set to 1 (transmit operation completed): When data transmit operation from the transmit shift register is completed. When the transmit interrupt request is generated, the SIO2TR bit in the IIO6IR register becomes 1 (interrupt requested) (See Figure 11.18). Receive interrupt: • When data is transferred from the receive shift register to the G2RB register (receive operation completed) When the receive interrupt request is generated, the SIO2RR bit in the IIO5IR register becomes 1 (interrupt requested) (See Figure 11.18).
Error detection	Overrun error Overrun error occurs when the j^{th} stop bit of the next data (data length: j bits (j = 1 to 8)) is received before reading the G2RB register. If an overrun error occurs, a read from the G2RB register returns an undefined value.
Selectable function	• LSB first or MSB first (Selectable only in 8-bits mode) Data is transmitted and received from either bit 0 or bit 7. Select LSB first except 8-bits mode. • ISTXD2 and ISRXD2 I/O polarity invert ISTXD2 pin output level and ISRXD2 pin input level are inverted

NOTE:
1. The serial clock must be fBT2 divided by six or lower frequency when the internal clock is selected, and the serial clock must be fBT2 divided by 20 or lower frequency when the external clock is selected. Additionally, meet the conditions shown on **Tables 27.26 and 27.49 Intelligent I/O communication function (Group 2)** in the chapter **27. Electrical Characteristics**.

Table 22.26 Register Settings in Variable Data Length Clock Synchronous Mode (Group 2)

Register	Bit	Function
G2BCR0	BCK1 and BCK0	Set to 11b
	DIV4 to DIV0	Select count source divide ratio
	IT	Set to 0
G2BCR1	–	Set to 0001 0010b
G2POCR0	–	Set to 0000 0111b
G2POCR1	–	Set to 0000 0111b
G2POCR2	–	Set to 0000 0010b
G2PO0	–	Set the value to compare for waveform generation Serial clock frequency: $\frac{fBT2}{2 \times (\text{setting value} + 2)}$
G2PO2	–	Set the value less than the setting value of the G2PO0 register
G2FE	IFE2 to IFE0	Set to 111b
G2MR	GMD1 and GMD0	Set to 01b
	CKDIR	Select either internal or external clock
	UFORM	Select either LSB first or MSB first
	IRS	Select the transmit interrupt source
G2CR	TE	Set to 1 when a transmit/receive operation is enabled
	TXEPT	Transmit shift register empty flag
	TI	G2TB register empty flag
	RE	Set to 1 when a receive operation is enabled
	RI	Receive complete flag
	OPOL	ISTXD2 output polarity invert (Set to 0 in normal use)
	IPOL	ISRXD2 input polarity invert (Set to 0 in normal use)
G2TB	–	Write data length and transmit data
G2RB	–	Received data and an error flag are stored

Table 22.27 Pin Settings in Variable Data Length Clock Synchronous Mode (Group 2)

Port	Function	G2POCR0 G2POCR1 Registers(4)	Bit Setting					
			IPS Register	PD6, PD7, PD9, PD13 Registers(2)	PSD1 Register	PSC Register	PSL0, PSL1, PSL3, PSL7 Registers	PS0, PS1 PS3, PS7 Registers(2)(3)
P6_4	ISCLK2 Input	–	IPS6 = 0	PD6_4 = 0	–	–	–	PS0_4 = 0
	ISCLK2 Output	G2POCR1	–	–	–	–	PSL0_4 = 1	PS0_4 = 1
P7_0(1)	ISTXD2 Output	G2POCR0	–	–	PSD1_0 = 0	PSC_0 = 1	PSL1_0 = 0	PS1_0 = 1
P7_1(1)	ISRXD2 Input	–	IPS5 and IPS4 = 00b	PD7_1 = 0	–	–	–	PS1_1 = 0
P9_1	ISRXD2 Input	–	IPS5 and IPS4 = 01b	PD9_1 = 0	–	–	–	PS3_1 = 0
P9_2	ISTXD2 Output	G2POCR0	–	–	–	–	PSL3_2 = 1	PS3_2 = 1
P13_4	ISTXD2 Output	G2POCR0	–	–	–	–	PSL7_4 = 0	PS7_4 = 1
P13_5	ISRXD2 Input	–	IPS5 and IPS4 =10b	PD13_5 = 0	–	–	–	PS7_5 = 0
P13_6	ISCLK2 Input	–	IPS6 = 1	PD13_6 = 0	–	–	–	PS7_6 = 0
	ISCLK2 Output	G2POCR1	–	–	–	–	PSL7_6 = 0	PS7_6 = 0

NOTES:
1. The P7_0 and P7_1 are the N-channel open drain output ports.
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. Set registers PS0, PS1, PS3, and PS7 after setting the other registers.
4. Set bits MOD2 to MOD0 in the corresponding register to 111b (use communication function output).

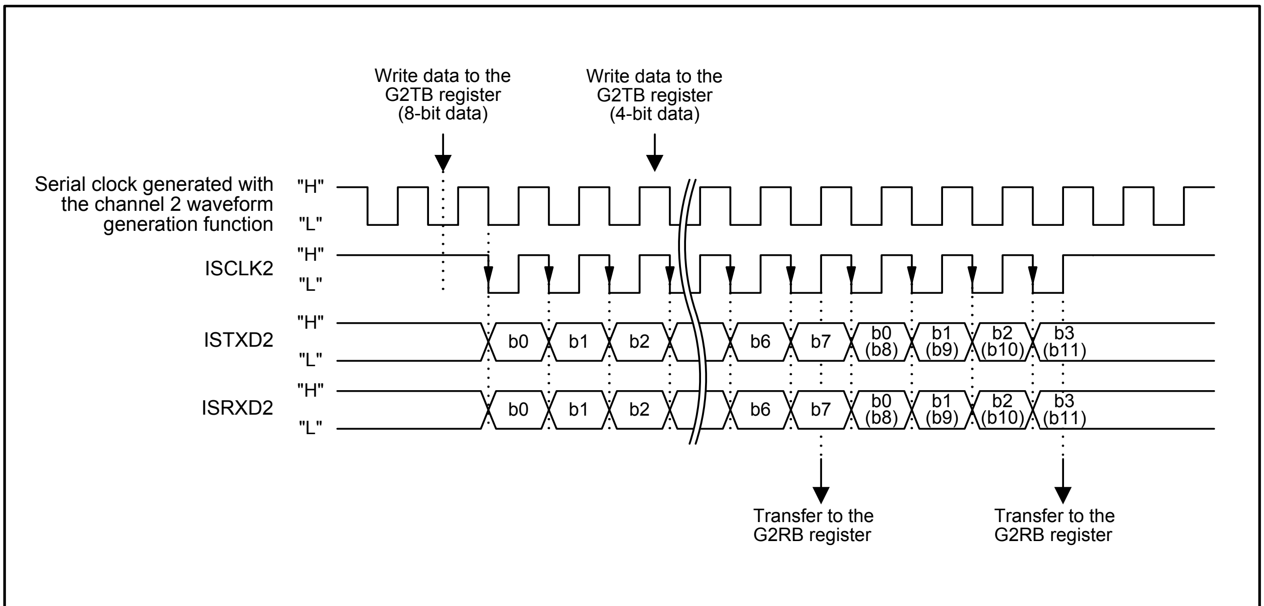


Figure 22.61 Transmit/Receive Operation in Variable Clock Synchronous Mode (Group 2)

23. CAN Module

NOTE

Only CAN0 can be used in the M32C/87A.
CAN Module is not available in the M32C/87B.

CAN (Controller Area Network) module included in the M32C/87 Group is a Full CAN module, supporting CAN Specification 2.0 Part B. Two channels, CAN0 and CAN1, can be used. Table 23.1 lists CAN module specifications of the CAN0 and CAN1 channels.

Table 23.1 CAN Module Specifications for CAN0 and CAN1

Item	Specification
Protocol	CAN Specification 2.0 Part B
Message slots	16 slots
Acceptance filter	Global mask: 1 (for the CANi message slots 0 to 13 (i = 0,1)) Local mask: 2 (for CANi message slots 14 and 15 respectively)
Baud rate(1)	$\text{Baud rate} = \frac{1}{\text{Tq x number of Tq per bit}}$ ---Max 1 Mbps $\text{Tq (time quantum)} = \frac{\text{BRP + 1}}{\text{CAN clock}}$ Number of Tq per bit = SS + PTS + PBS1 + PBS2 BRP: Setting value of registers C0BRP and C1BRP; 1 to 255 SS: Synchronization Segment; 1Tq PTS: Propagation Time Segment; 1 to 8Tq PBS1: Phase Buffer Segment 1; 2 to 8Tq PBS2: Phase Buffer Segment 2; 2 to 8Tq
Remote frame automatic answering function	Message slot which receives a remote frame transmits a data frame automatically
Time stamp function	The time stamp function is used with a 16-bit counter. Count source can be selected from the CAN bus bit clock divided by 1, 2, 3, or 4 $\text{CAN bus bit clock} = \frac{1}{\text{CAN bit time}}$ CAN bit time = Tq x number of Tq per bit
BasicCAN mode	The BasicCAN function can be used with the CANi message slots 14 and 15
Transmit abort function	A transmit request is aborted
Loopback function	Frame transmitted by the CAN module is received by the same CAN module
Forcible error active transition function	The CAN module is forcibly placed in an error active state by an error counter reset
Single-shot transmit function	The CAN module does not retransmit data even if arbitration lost or transmit error causes a transmit failure
Self-test function	The CAN module communicates internally to check on a CAN module state

NOTE:
1. Use an oscillator with maximum 1.58% oscillator tolerance.

Figure 23.1 shows a block diagram of the CAN module for CAN0 and CAN1. Figure 23.2 shows CANi message slot buffer (i = 0, 1) and CANi message slot (message slot) j (j = 0 to 15). Table 23.2 lists pin settings of the CAN module. The CPU cannot access the message slot directly. Allocate necessary message slot j to the CANi message slot buffer 0 or 1 and access the message slot j via the allocated address. The CiSBS register selects the message slot j to be allocated. The message slot buffer and message slot consist of 16 bytes shown in Figure 23.2.

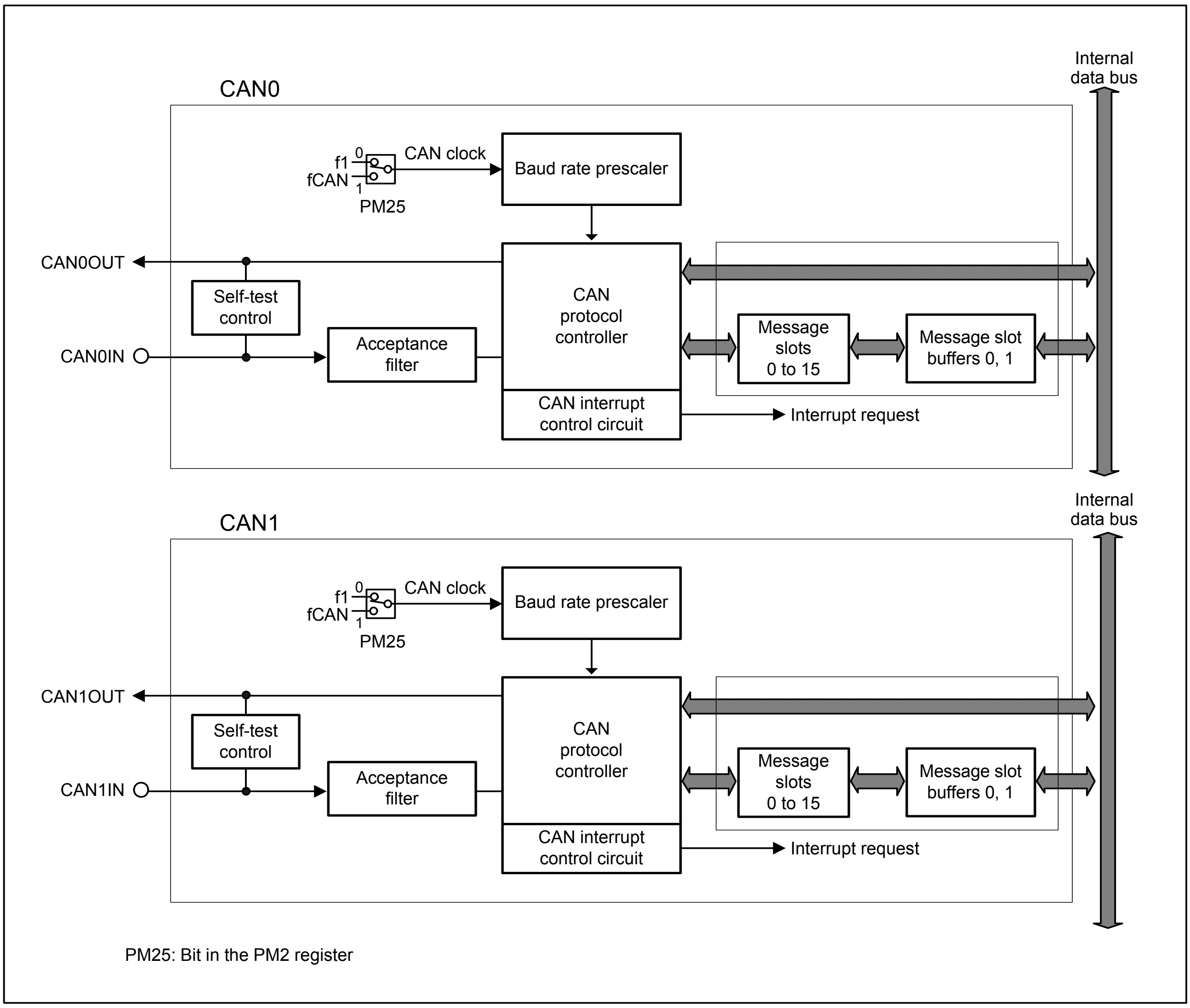


Figure 23.1 CAN Module Block Diagram for CAN0 and CAN1

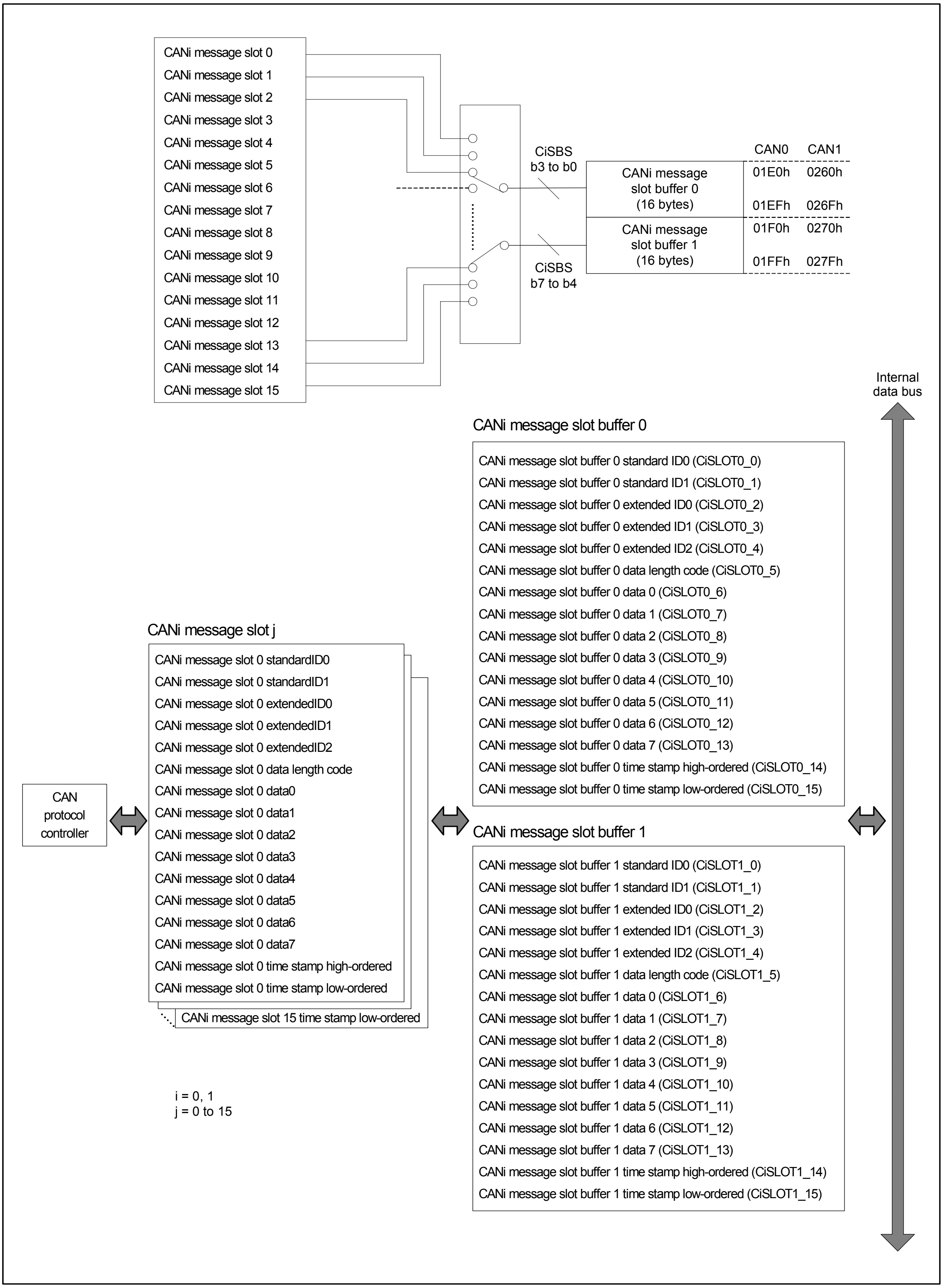


Figure 23.2 Message Slots and Message Slot Buffers for CAN0 and CAN1

Table 23.2 Pin Settings[1]

Port	Function	Bit and Setting				
		PD7 to PD9 Registers[2]	PSC, PSC2, PSC3 Registers	PSL1 to PSL3 Registers	PS1 to PS3 Registers[2]	IPS, IPSA Registers
P7_6	CAN0OUT	–	PSC_6 = 1	PSL1_6 = 0	PS1_6 = 1	–
P7_7	CAN0IN	PD7_7 = 0	–	–	PS1_7 = 0	IPS3 = 0
P8_2	CAN0OUT	–	PSC2_2 = 0	PSL2_2 = 1	PS2_2 = 1	–
	CAN1OUT	–	PSC2_2 = 1	PSL2_2 = 1	PS2_2 = 1	–
P8_3	CAN0IN	PD8_3 = 0	–	–	–	IPS3 = 1
	CAN1IN	PD8_3 = 0	–	–	–	IPSA_3 = 1
P9_5	CAN1IN / $\overline{\text{CAN1WU}}$	PD9_5 = 0	–	PSL3_5 = 0	PS3_5 = 0	IPSA_3 = 0
P9_6	CAN1OUT	–	PSC3_6 = 1	–	PS3_6 = 1	–

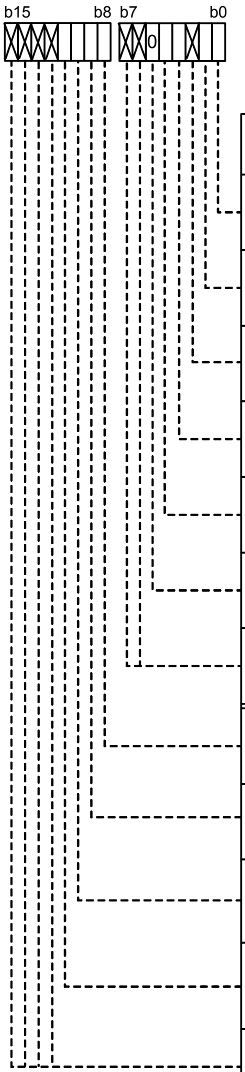
NOTES:
1. Set the registers from the left column sequentially.
2. Set the PD9 or PS3 register immediately after setting the PRC2 bit in the PRCR register to 1 (write enable). Do not generate an interrupt or a DMA or DMACII transfer between these two instructions.

23.1 CAN-Associated Registers

Figures 23.3 to 23.19, 23.21 to 23.27, and 23.30 to 23.36 show the CAN-associated registers. Refer to **23.2 CAN Clock and CPU Clock** for details.

23.1.1 CANi Control Register 0 (CiCTLR0 Register) (i = 0, 1)

CANi Control Register 0 (i = 0, 1)



Symbol	Address	After Reset(1)
C0CTLR0	0201h - 0200h	XXXX 0000 XX01 0X01b
C1CTLR0	0281h - 0280h	XXXX 0000 XX01 0X01b

Bit Symbol	Bit Name	Function	RW
RESET0	CAN reset bit 0(2)	0: CAN module is out of reset 1: CAN module is reset	RW
LOOPBACK	Loop back mode select bit	0: Loop back function disabled 1: Loop back function enabled	RW
– (b2)	Unimplemented. Write 0. Read as undefined value.		–
BASICCAN	BasicCAN mode select bit	0: BasicCAN mode function disabled 1: BasicCAN mode function enabled	RW
RESET1	CAN reset bit 1(2)	0: CAN module is out of reset 1: CAN module is reset	RW
– (b5)	Reserved bit	Set to 0	RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–
TSPRE0	Time stamp prescaler select bits	b9 b8 0 0: CAN bus bit clock 0 1: CAN bus bit clock divided by 2 selected 1 0: CAN bus bit clock divided by 3 selected 1 1: CAN bus bit clock divided by 4 selected	RW
TSPRE1			RW
TSRESET	Time stamp counter reset bit	0: Nothing occurs 1: The CiTSR register becomes 0000h, then this bit is automatically set back to 0	RW
ECRESET	Error counter reset bit	0: Nothing occurs 1: Registers CiTEC and CiREC become 00h, then this bit is automatically set back to 0	RW
– (b15-b12)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.
2. Set both the RESET1 and RESET0 bits to the same value simultaneously.

Figure 23.3 C0CTLR0 and C1CTLR0 Registers

23.1.1.1 RESET1 and RESET0 Bits

When both the RESET1 and RESET0 bits are set to 1 (CAN module is reset), the CAN module is immediately reset regardless of ongoing CAN communication.
After both the RESET1 and RESET0 bits are set to 1 and the CAN module reset is completed, the CiTSR register (i = 0, 1) becomes 0000h. Also, registers CiTEC and CiREC become 00h and both the STATE_ERRPAS and STATE_BUSOFF bits in the CiSTR register become 0.
When both the RESET1 and RESET0 bits are changed from 1 to 0, the CiTSR register starts counting and the CAN module is permitted to communicate after 11 consecutive recessive bits have been detected.

NOTES:
1. Set the same value to both the RESET1 and RESET0 bits simultaneously.
2. Ensure that the STATE_RESET bit in the CiSTR register becomes 1 (CAN module is in reset) after both the RESET1 and RESET0 bits are set to 1.
3. The CANiOUT pin outputs a high-level (“H”) signal as soon as both the RESET1 and RESET0 bits are set to 1. CAN bus error may occur by setting both the RESET1 and RESET0 bits to 1 while the CAN frame is being transmitted.
4. To select pins CANiIN and CANiOUT for CAN communication, set registers PS1, PS2, PS3, PSL1, PSL2, PSL3, PSC, PSC2, PSC3, IPS, IPSA, PD7, PD8, and PD9 while the STATE_RESET bit is 1 (CAN module is in reset).

23.1.1.2 LOOPBACK Bit

When the LOOPBACK bit is set to 1 (loopback function enabled) and the receive message slot has the identifier (ID) and frame format matched with a transmitted frame, the transmitted frame is stored to the receive message slot.

NOTES:
1. No ACK for the transmitted frame is returned.
2. Change the LOOPBACK bit setting while the STATE_RESET bit is 1 (CAN module is in reset).

23.1.1.3 BASICCAN Bit

When the BASICCAN bit is set to 1, the message slots 14 and 15 enter BasicCAN mode.
In BasicCAN mode, the message slots 14 and 15 are configured as double buffered.
Acceptance filtering permits the receive frames having the matching IDs to be stored into the message slots 14 and 15 alternately. Both data frame and remote frame can be received.
Use the following procedure to enter BasicCAN mode.

(1) Set the BASICCAN bit to 1.
(2) Set the same ID to the message slots 14 and 15.
(3) Set the same values in registers CiLMAR0 to CiLMAR4 and CiLMBR0 to CiLMBR4.
(4) Set the same value to bits IDE14 and IDE15 in the CiIDR register.
(5) Set registers CiMCTL14 and CiMCTL15 to receive a data frame.

NOTES:
1. Change the BASICCAN bit setting while the STATE_RESET bit is 1 (CAN module is in reset).
2. The message slot 14 is the first slot to become active after both the RESET1 and RESET0 bits are set to 0.
3. The message slots 0 to 13 are not affected by entering BasicCAN mode.

23.1.1.4 TSPRE1 and TSPRE0 Bits

Bits TSPRE1 and TSPRE0 determine the count source of the time stamp counter.

NOTE:
1. Change bits TSPRE1 and TSPRE0 setting while the STATE_RESET bit is 1 (CAN module is in reset).

23.1.1.5 TSRESET Bit

When the TSRESET bit is set to 1 (count reset), the CiTSR register becomes 0000h. The TSRESET bit is automatically set back to 0 after the CiTSR register becomes 0000h.

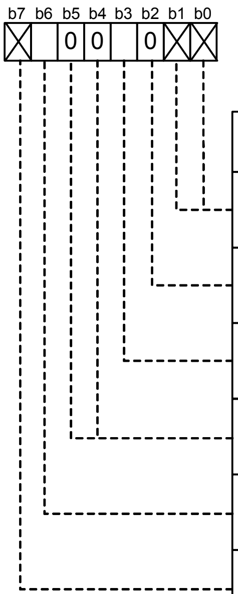
23.1.1.6 ECRESET Bit

When the ECRESET bit is set to 1, registers CiTEC and CiREC become 00h and the CAN module are forcibly placed in an error active state.
The ECRESET bit is automatically set back to 0 after the CAN module enters an error active state.

NOTES:
1. Once entering an error active state, the CAN module is permitted to communicate after 11 consecutive recessive bits have been detected on the CAN bus.
2. Set the ECRESET bit to 1 while the CAN module is in a bus-off or bus-idle state. Do not set the ECRESET bit to 1 while the CAN module is transmitting or receiving.

23.1.2 CANi Control Register 1 (CiCTLR1 Register) (i = 0, 1)

CANi Control Register 1 (i = 0, 1)



Symbol	Address	After Reset[(1)]
C0CTLR1	0241h	X000 00XXb
C1CTLR1	0251h	X000 00XXb

Bit Symbol	Bit Name	Function	RW
– (b1-b0)	Unimplemented. Write 0. Read as undefined value.		–
– (b2)	Reserved bit	Set to 0	RW
BANKSEL	CANi bank switch bit	0: Message slot control register and single-shot register selected 1: Mask register selected	RW
– (b5-b4)	Reserved bits	Set to 0	RW
INTSEL	CANi interrupt mode select bit	0: Output 3 types of interrupt via OR gate 1: Output 3 types of interrupt individually	RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.4 C0CTLR1 and C1CTLR1 Registers

23.1.2.1 BANKSEL Bit

The BANKSEL bit in the C0CTLR1 register selects the registers allocated to addresses 0220h to 023Fh. The BANKSEL bit in the C1CTLR1 register selects the registers allocated to addresses 02A0h to 02BFh.
Registers CiSSCTLR, CiSSSTR, and CiMCTL0 to CiMCTL15 can be accessed by setting the BANKSEL bit to 0. Registers CiGMR0 to CiGMR4, CiLMAR0 to CiLMAR4, and CiLMBR0 to CiLMBR4 can be accessed by setting the BANKSEL bit to 1.

23.1.2.2 INTSEL Bit

The INTSEL bit determines whether three types of interrupts (CANi transmit interrupt, CANi receive interrupt and CANi error interrupt) are output via OR gate or output individually.
Refer to **23.4 CAN Interrupts** for details.

NOTE:
1. Change the INTSEL bit setting when the STATE_RESET bit in the CiSTR register is 1 (CAN module is in reset).

23.1.3 CANi Sleep Control Register (CiSLPR Register) (i = 0, 1)

CANi Sleep Control Register (i = 0, 1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
C0SLPR	0242h	XXXX XXX0b
C1SLPR	0252h	XXXX XXX0b

Bit Symbol	Bit Name	Function	RW
SLEEP	Sleep mode control bit	0: Sleep mode entered 1: Sleep mode exited(1)	RW
− (b7-b1)	Unimplemented. Write 0. Read as undefined value.		−

NOTE:
1. Perform the initialization for the CAN module after CAN sleep mode is exited.
While the CAN0 module is in sleep mode, no SFR associated with CAN0 (allocated in addresses 01E0h to 0245h) can be accessed, except for the C0SLPR register.
While the CAN1 module is in sleep mode, no SFR associated with CAN1 (allocated in addresses 0250h to 02BFh) can be accessed, except for the C1SLPR register.

Figure 23.5 C0SLPR and C1SLPR Registers

23.1.3.1 SLEEP Bit

When the SLEEP bit is set to 0, the clock is not supplied to the CAN module and the CAN module enters sleep mode.
When the SLEEP bit is set to 1, the clock is supplied to the CAN module and the CAN module exits sleep mode.

NOTE:
1. Enter sleep mode after the STATE_RESET bit in the CiSTR register becomes 1 (CAN module is in reset).

23.1.4 CANi Status Register (CiSTR Register) (i = 0, 1)

CANi Status Register (i = 0, 1)

b15 b8 b7 b0

Symbol	Address	After Reset[1]
C0STR	0203h - 0202h	X000 0X01 0000 0000b
C1STR	0283h - 0282h	X000 0X01 0000 0000b

Bit Symbol	Bit Name	Function	RW
MBOX0	Active slot determinate bits	b3 b2 b1 b0 0 0 0 0: Message slot 0 0 0 0 1: Message slot 1 0 0 1 0: Message slot 2 0 0 1 1: Message slot 3 . . . . . . 1 1 0 0: Message slot 12 1 1 0 1: Message slot 13 1 1 1 0: Message slot 14 1 1 1 1: Message slot 15	RO
MBOX1			RO
MBOX2			RO
MBOX3			RO
TRMSUCC	Transmit/Receive complete state flag	b5 b4 0 0: Hasn't yet transmitted nor received 0 1: Transmit operation completed 1 0: Receive operation completed	RO
RECSUCC			RO
TRMSTATE	Transmit state flag	0: Not transmitting 1: While transmitting	RO
RECSTATE	Receive state flag	0: Not receiving 1: While receiving	RO
STATE_RESET	CAN reset state flag	0: CAN module is not in reset 1: CAN module is in reset	RO
STATE_LOOPBACK	Loop back state flag	0: Not in Loop back mode 1: Loop back mode	RO
– (b10)	Unimplemented. Write 0. Read as undefined value.		–
STATE_BASICCAN	BasicCAN state flag	0: Not in BasicCAN mode 1: BasicCAN mode	RO
STATE_BUSERROR	CAN bus error state flag	0: No error occurred 1: Error occurred	RO
STATE_ERRPAS	Error passive state flag	0: Not in error passive state 1: In Error passive state	RO
STATE_BUSOFF	Bus-off state flag	0: Not in bus-off state 1: Bus-off state	RO
– (b15)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.6 C0STR and C1STR Registers

23.1.4.1 MBOX3 to MBOX0 Bits

When a transmit operation is completed or a received data is stored, bits MBOX3 to MBOX0 indicate the memory slot number which is used for the operation.

23.1.4.2 TRMSUCC Bit

The TRMSUCC bit becomes 1 when a transmit operation is successfully completed.
The TRMSUCC bit becomes 0 when a receive operation is successfully completed.

23.1.4.3 RECSUCC Bit

The RECSUCC bit becomes 1 when a receive operation is successfully completed (regardless of whether a receive message has been stored in the message slot). When a message transmitted in loopback mode is received, the TRMSUCC bit becomes 1 and the RECSUCC bit becomes 0.
The RECSUCC bit becomes 0 when a transmit operation is successfully completed.

23.1.4.4 TRMSTATE Bit

The TRMSTATE bit becomes 1 when the CAN module is operating as a transmit node.
The TRMSTATE bit becomes 0 when the CAN module is in a bus-idle state or starts operating as a receive node.

23.1.4.5 RECSTATE Bit

The RECSTATE bit becomes 1 when the CAN module is operating as a receive node.
The RECSTATE bit becomes 0 when the CAN module is in a bus-idle state or starts operating as a transmit node.

23.1.4.6 STATE_RESET Bit

After both the RESET1 and RESET0 bits are set to 1 (CAN module is reset), the STATE_RESET bit becomes 1 as soon as the CAN module reset is completed.
The STATE_RESET bit becomes 0 when both the RESET1 and RESET0 bits are set to 0 (CAN module is out of reset).

23.1.4.7 STATE_LOOPBACK Bit

The STATE_ LOOPBACK bit is 1 while the CAN module is operating in loopback mode.
The STATE_LOOPBACK bit becomes 1 when the LOOPBACK bit in the CiCTLR0 register is set to 1 (loop back function enabled).
The STATE_LOOPBACK bit becomes 0 when the LOOPBACK bit is set to 0 (loop back function disabled).

23.1.4.8 STATE_BASICCAN Bit

The STATE_BASICCAN bit is 1 while the CAN module is operating in BasicCAN mode. Refer to **23.1.1.3 BASICCAN Bit** for information about BasicCAN mode.
The STATE_BASICCAN bit becomes 0 when the BASICCAN bit is set to 0 (BasicCAN mode function disabled).
The STATE_BASICCAN bit becomes 1 when the BASICCAN bit is set to 1 (BasicCAN mode function enabled) and registers CiMCTL14 and CiMCTL15 are set to receive a data frame.

23.1.4.9 STATE_BUSERROR Bit

The STATE_BUSERROR bit becomes 1 when a CAN communication error is detected.
The STATE_BUSERROR bit becomes 0 when transmit and receive operations are successfully completed (regardless of whether a receive message has been stored in the message slot).

NOTE:

1. When the STATE_BUSERROR bit is 1, the STATE_BUSERROR bit remains unchanged even if both the RESET1 and RESET0 bits are set to 1 (CAN module is reset).

23.1.4.10 STATE_ERRPAS Bit

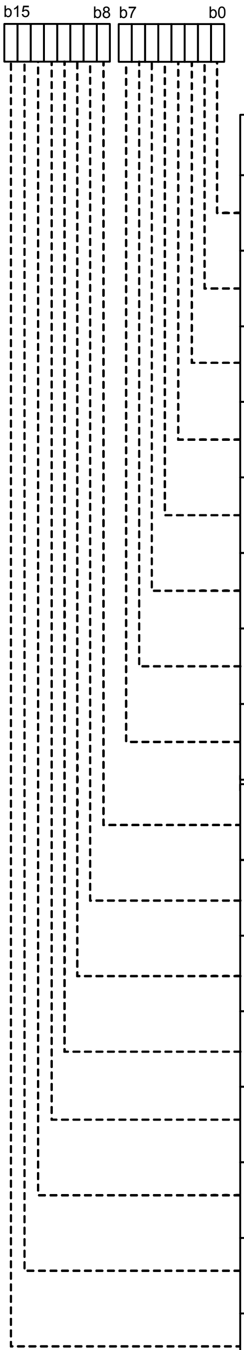
The STATE_ERRPAS bit becomes 1 when the value of the CiTEC or CiREC register (i = 0, 1) exceeds 127 which results in the CAN module to be placed in an error-passive state.
The STATE_ERRPAS bit becomes 0 when the CAN module exits an error-passive state to enter another error state.
The STATE_ERRPAS bit becomes 0 when both the RESET1 and RESET0 bits are set to 1 (CAN module is reset).

23.1.4.11 STATE_BUSOFF Bit

The STATE_BUSOFF bit becomes 1 when the value of the CiTEC register exceeds 255 which results in the CAN module to be placed in a bus-off state.
The STATE_BUSOFF bit becomes 0 when the CAN module exits a bus-off state to enter an error-active state.
The STATE_BUSOFF bit becomes 0 when both the RESET1 and RESET0 bits are set to 1 (CAN module is reset).

23.1.5 CANi Extended ID Register (CiIDR Register) (i = 0, 1)

CANi Extended ID Register[(1)] (i = 0,1)



Symbol	Address	After Reset[(2)]
C0IDR	0205h - 0204h	0000h
C1IDR	0285h - 0284h	0000h

Bit Symbol	Bit Name	Function	RW
IDE15	Extended ID15 (message slot 15)	Determine whether the corresponding message slot is the standard format or extended format 0: Standard format 1: Extended format	RW
IDE14	Extended ID14 (message slot 14)		RW
IDE13	Extended ID13 (message slot 13)		RW
IDE12	Extended ID12 (message slot 12)		RW
IDE11	Extended ID11 (message slot 11)		RW
IDE10	Extended ID10 (message slot 10)		RW
IDE9	Extended ID9 (message slot 9)		RW
IDE8	Extended ID8 (message slot 8)		RW
IDE7	Extended ID7 (message slot 7)		RW
IDE6	Extended ID6 (message slot 6)		RW
IDE5	Extended ID5 (message slot 5)		RW
IDE4	Extended ID4 (message slot 4)		RW
IDE3	Extended ID3 (message slot 3)		RW
IDE2	Extended ID2 (message slot 2)		RW
IDE1	Extended ID1 (message slot 1)		RW
IDE0	Extended ID0 (message slot 0)		RW

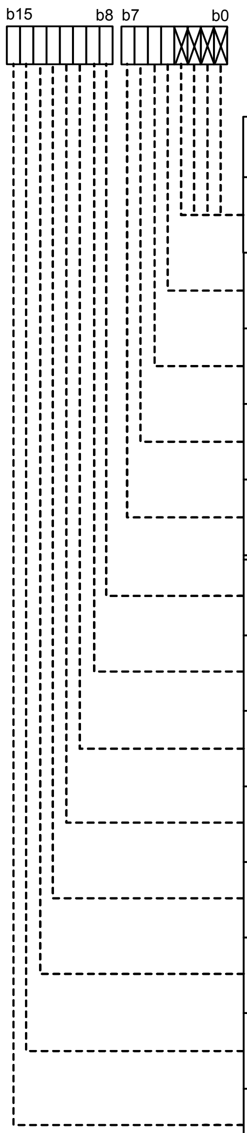
NOTES:
1. Change the CiIDR register while the CiMCTLj register (j = 0 to 15) of the corresponding message slot to the bit to be changed, is set to 00h.
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.7 C0IDR and C1IDR Registers

Bits in the CiIDR register determine a frame format used in the message slot corresponding to the individual bit.
The standard format is selected when the bit is set to 0.
The extended format is selected when the bit is set to 1.

23.1.6 CANi Configuration Register (CiCONR Register) (i = 0, 1)

CANi Configuration Register (i = 0, 1)(1)



Symbol	Address	After Reset(2)
C0CONR	0207h - 0206h	0000 0000 0000 XXXXb
C1CONR	0287h - 0286h	0000 0000 0000 XXXXb

Bit Symbol	Bit Name	Function	RW
– (b3-b0)	Unimplemented. Write 0. Read as undefined value.		–
SAM	Sampling count	0: Sample once 1: Sample three times	RW
PTS0	Propagation time segment	b7 b6 b5 0 0 0: 1Tq 0 0 1: 2Tq 0 1 0: 3Tq 0 1 1: 4Tq 1 0 0: 5Tq 1 0 1: 6Tq 1 1 0: 7Tq 1 1 1: 8Tq	RW
PTS1			RW
PTS2			RW
PBS10	Phase buffer segment 1	b10 b9 b8 0 0 0: Do not set to this value 0 0 1: 2Tq 0 1 0: 3Tq 0 1 1: 4Tq 1 0 0: 5Tq 1 0 1: 6Tq 1 1 0: 7Tq 1 1 1: 8Tq	RW
PBS11			RW
PBS12			RW
PBS20	Phase buffer segment 2	b13 b12 b11 0 0 0: Do not set to this value 0 0 1: 2Tq 0 1 0: 3Tq 0 1 1: 4Tq 1 0 0: 5Tq 1 0 1: 6Tq 1 1 0: 7Tq 1 1 1: 8Tq	RW
PBS21			RW
PBS22			RW
SJW0	Resynchronization jump width	b15 b14 0 0: 1Tq 0 1: 2Tq 1 0: 3Tq 1 1: 4Tq	RW
SJW1			RW

NOTES:
1. Set the CiCONR register while the STATE_RESET bit in the CiSTR register is 1 (CAN module is in reset).
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.8 C0CONR and C1CONR Registers

23.1.6.1 SAM Bit

The SAM bit determines the number of sampling points to be taken per bit.
When the SAM bit is set to 0, only one sample is taken per bit at the end of the Phase Buffer Segment 1 (PBS1) to determine the value of the bit.
When the SAM bit is set to 1, three samples per bit are taken; one time quantum and two time quanta before the end of PBS1, and at the end of PBS1. The value detected twice or more becomes the value of the sampled bit.

23.1.6.2 PTS2 to PTS0 Bits

Bits PTS2 to PTS0 determine the number of Tq for PTS.

23.1.6.3 PBS12 to PBS10 Bits

Bits PBS12 to PBS10 determine the number of Tq for PBS1. Set bits PBS12 to PBS10 to other than 000b.

23.1.6.4 PBS22 to PBS20 Bits

Bits PBS22 to PBS20 determine the number of Tq for PBS2. Set bits PBS22 to PBS20 to other than 000b.

23.1.6.5 SJW1 and SJW0 Bits

Bits SJW1 and SJW0 determine the number of Tq for SJW.

Table 23.3 Bit Timing when CAN Clock = 30 MHz

Baud Rate	BRP	Tq (ns)	Number of Tq Per Bit	PTS + PBS1	PBS2	Sampling Point
1 Mbps	1	66.7	15	12	2	87%
	1	66.7	15	11	3	80%
	1	66.7	15	10	4	73%
	2	100	10	7	2	80%
	2	100	10	6	3	70%
	2	100	10	5	4	60%
500 Kbps	2	100	20	16	3	85%
	2	100	20	15	4	80%
	2	100	20	14	5	75%
	3	133.3	15	12	2	87%
	3	133.3	15	11	3	80%
	3	133.3	15	10	4	73%
	4	166.7	12	9	2	83%
	4	166.7	12	8	3	75%
	4	166.7	12	7	4	67%
	5	200	10	7	2	80%
	5	200	10	6	3	70%
	5	200	10	5	4	60%

23.1.7 CANi Baud Rate Prescaler (CiBRP Register) (i = 0, 1)

CANi Baud Rate Prescaler (i = 0, 1)(1)

b7 b0

Symbol	Address	After Reset(2)
C0BRP	0217h	0000 0001b
C1BRP	0297h	0000 0001b

Function	Setting Range	RW
If the setting value is *n*, the CAN clock is divided by *n+1*.	01h to FFh(3)	RW

NOTES:
1. Set the CiBRP register while the STATE_RESET bit in the CiSTR register is 1 (CAN module is in reset).
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.
3. Do not set the CiBRP register to 00h (divide-by-1).

Figure 23.9 C0BRP and C1BRP Registers

The CiBRP register determines a Tq of the CAN bit time.

$$Tq = \frac{BRP + 1}{\text{CAN clock}}$$

Tq: Time quantum
BRP: Setting value of the CiBRP register (1 to 255)

$$\text{Baud rate} = \frac{1}{Tq \times \text{number of Tq per bit}}$$

Number of Tq per bit = SS + PTS + PBS1 + PBS2

The CAN bit time is comprised of the following four segments.

(1) SS: Synchronization Segment
This segment is used to monitor the falling edge of a bit in order to synchronize the various CAN modules.

(2) PTS: Propagation Time Segment
This segment is used to compensate for the physical delay times within the CAN network. The physical delay times within the network is twice the sum of the signal propagation delay on the CAN bus, the input comparator delay, and the output driver delay.

(3) PBS1: Phase Buffer Segment 1
This segment is used to compensate for the edge phase error caused by the frequency error. If the falling edge of a bit comes in later than expected, PBS1 is lengthened by up to the resynchronization jump width.

(4) PBS2: Phase Buffer Segment 2
This segment has the same functionality to PBS1. If the falling edge of a bit comes in sooner than expected, PBS2 is shortened by up to the resynchronization jump width.

- SJW: Resynchronization Jump Width
This is the amount of lengthening or shortening of the phase buffer segments to compensate for the phase error.

Figure 23.10 shows a bit timing diagram.

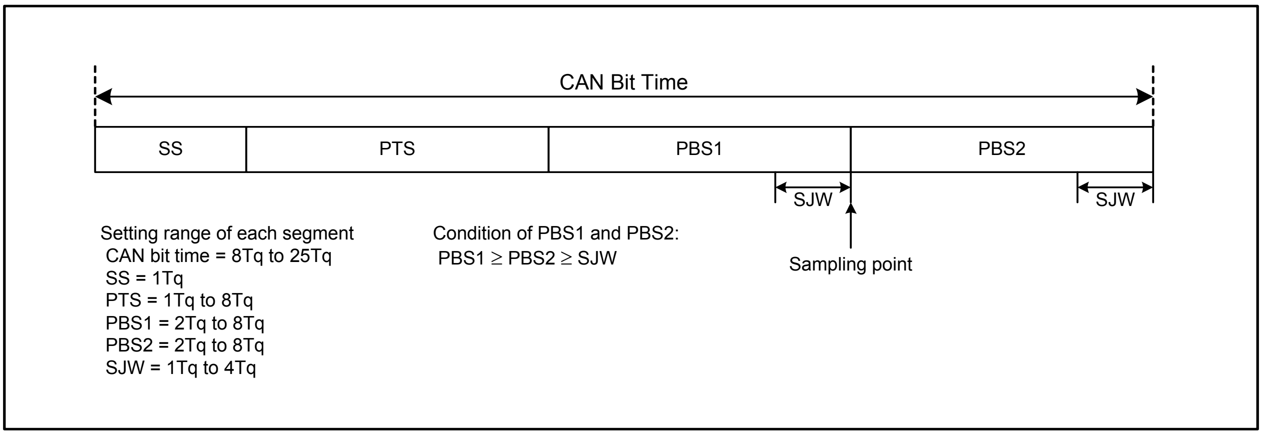


Figure 23.10 Bit Timing Diagram

23.1.8 CANi Time Stamp Register (CiTSR Register) (i = 0, 1)

CANi Time Stamp Register (i = 0, 1)

b15 b8 b7 b0

Symbol	Address	After Reset[1]
C0TSR	0209h - 0208h	0000h
C1TSR	0289h - 0288h	0000h

Function	RW
Value of time stamp	RO

NOTE:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.11 C0TSR and C1TSR Registers

The CiTSR register is a 16-bit counter. Bits TSPRE1 and TSPRE0 in the CiCTLR0 register determine the CAN bus bit clock divided by 1, 2, 3, or 4 as the count source.
When a transmit or receive operation is completed, the value of the CiTSR register is automatically stored into the message slot.
In loopback mode, the value of the CiTSR register is stored into the data frame receive message slot or remote frame receive message slot when a receive operation is completed, if the corresponding message slot is available to store the message. The value of the CiTSR register is not stored when a transmit operation is completed in loopback mode.

The CiTSR register starts a counter increment when both the RESET1 and RESET0 bits in the CiCTLR0 register are set to 0 (CAN module is out of reset).
The CiTSR register becomes 0000h in the following timings:

- At the next count timing after the CiTSR register becomes FFFFh.
- When both the RESET1 and RESET0 bits are set to 1 (CAN module is reset) by a program.
- When the TSRESET bit in the CiCTLR0 register is set to 1 (CiTSR register reset) by a program.

$$\text{CAN bus bit clock} = \frac{1}{\text{CAN bit time}}$$

23.1.9 CANi Transmit Error Count Register (CiTEC Register) (i = 0, 1)

CANi Transmit Error Count Register (i = 0, 1)

b7 b0

Symbol	Address	After Reset[1]
C0TEC	020Ah	00h
C1TEC	028Ah	00h

Function	RW
Transmit error count value	RO

NOTE:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.12 C0TEC and C1TEC Registers

In an error active and an error passive state, a transmit error count value is stored into the CiTEC register. The count is decremented when a transmit operation is successfully completed and incremented when a transmit error occurs.
In a bus-off state, the value in the CiTEC register is undefined. The CiTEC register becomes 00h when the CAN module is placed in an error active state again.

23.1.10 CANi Receive Error Count Register (CiREC Register) (i = 0, 1)

CANi Receive Error Count Register (i = 0, 1)

b7 b0

Symbol	Address	After Reset[1]
C0REC	020Bh	00h
C1REC	028Bh	00h

Function	RW
Receive error count value	RO

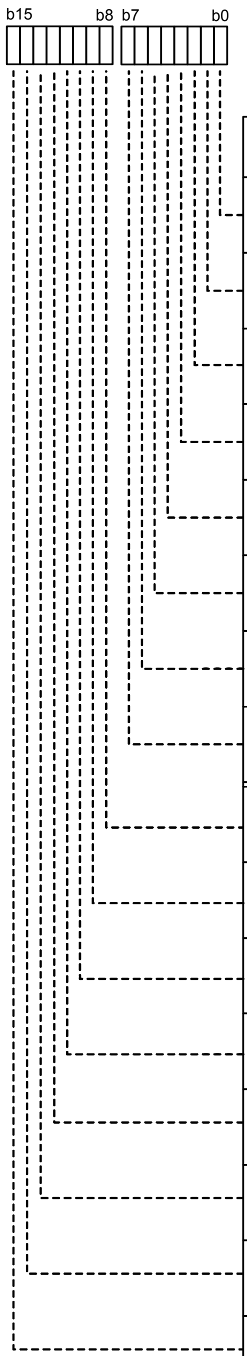
NOTE:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.13 C0REC and C1REC Registers

In an error active and an error passive state, a receive error count value is stored into the CiREC register. The count is decremented when a receive operation is successfully completed and incremented when a receive error occurs.
The CiREC register becomes 127 when a receive operation is successfully completed while the CiREC register equals or exceeds 128 (in an error passive state).
In a bus-off state, the value in the CiREC register is undefined. The CiREC register becomes 00h when the CAN module is placed in an error active state again.

23.1.11 CANi Slot Interrupt Status Register (CiSISTR Register) (i = 0, 1)

CANi Slot Interrupt Status Register (i = 0, 1)



Symbol	Address	After Reset[1]
C0SISTR	020Dh - 020Ch	0000h
C1SISTR	028Dh - 028Ch	0000h

Bit Symbol	Bit Name	Function	RW
SIS15	Message slot 15 interrupt request status bit	Determines whether an interrupt of a corresponding message slot is requested or not. 0: Interrupt not requested 1: Interrupt requested[2]	RW
SIS14	Message slot 14 interrupt request status bit		RW
SIS13	Message slot 13 interrupt request status bit		RW
SIS12	Message slot 12 interrupt request status bit		RW
SIS11	Message slot 11 interrupt request status bit		RW
SIS10	Message slot 10 interrupt request status bit		RW
SIS9	Message slot 9 interrupt request status bit		RW
SIS8	Message slot 8 interrupt request status bit		RW
SIS7	Message slot 7 interrupt request status bit		RW
SIS6	Message slot 6 interrupt request status bit		RW
SIS5	Message slot 5 interrupt request status bit		RW
SIS4	Message slot 4 interrupt request status bit		RW
SIS3	Message slot 3 interrupt request status bit		RW
SIS2	Message slot 2 interrupt request status bit		RW
SIS1	Message slot 1 interrupt request status bit		RW
SIS0	Message slot 0 interrupt request status bit		RW

NOTES:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.
2. Set each bit to 0 by a program. Writing a 1 has no effect.

Figure 23.14 C0SISTR and C1SISTR Registers

When using the CAN interrupt, the CiSISTR register (i = 0, 1) indicates which message slot has requested an interrupt. The SISj bit (j = 0 to 15) is not automatically set to 0 (interrupt not requested) even if the interrupt is acknowledged. Set the SISj bit to 0 by a program.
Use the MOV instruction to set the SISj bits to 0. Write a 0 to the bit which is to set to 0, and write a 1 to the bit which is to remain unchanged.

For example: To set the SIS0 bit in CAN0 to 0

```
mov.w #07FFFh, C0SISTR
```

Refer to **23.4 CAN Interrupts** for details.

23.1.11.1 Message Slot for Transmit Operation

The SISj bit becomes 1 (interrupt requested) when the value of the CiTSR register is stored into the message slot j after a transmit operation is completed.

23.1.11.2 Message Slot for Receive Operation

The SISj bit becomes 1 (interrupt requested) when the receive message is stored in the message slot j after a receive operation is completed.

NOTES:

1. If the RSPLOCK bit in registers CiMCTL0 to CiMCTL15 is set to 0 (automatic answering to the remote frame enabled), the SISj bit becomes 1 both when the remote frame receive operation is completed and when the following data frame transmit operation is completed.
2. In the remote frame transmit message slot, the SISj bit becomes 1 both when the remote frame transmit operation is completed and when the data frame receive operation is completed.
3. If an interrupt generation (the SISj bit becomes 1) and writing a 0 to the SISj bit by a program occur simultaneously, the SISj bit becomes 1.
4. Regardless of whether the SIMj bit in the CiSIMKR register is set to 0 (interrupt request masked) or to 1 (interrupt request enabled), the SISj bit becomes 1 at the completion of the transmit operation or at the completion of the receive operation.

23.1.12 CANi Slot Interrupt Mask Register (CiSIMKR Register) (i = 0, 1)

CANi Slot Interrupt Mask Register (i = 0, 1)(1)

b15 b8 b7 b0

Symbol	Address	After Reset(2)
C0SIMKR	0211h - 0210h	0000h
C1SIMKR	0291h - 0290h	0000h

Bit Symbol	Bit Name	Function	RW
SIM15	Message slot 15 interrupt request mask bit	Controls whether an interrupt request of the corresponding message slot is enabled or masked. 0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
SIM14	Message slot 14 interrupt request mask bit		RW
SIM13	Message slot 13 interrupt request mask bit		RW
SIM12	Message slot 12 interrupt request mask bit		RW
SIM11	Message slot 11 interrupt request mask bit		RW
SIM10	Message slot 10 interrupt request mask bit		RW
SIM9	Message slot 9 interrupt request mask bit		RW
SIM8	Message slot 8 interrupt request mask bit		RW
SIM7	Message slot 7 interrupt request mask bit		RW
SIM6	Message slot 6 interrupt request mask bit		RW
SIM5	Message slot 5 interrupt request mask bit		RW
SIM4	Message slot 4 interrupt request mask bit		RW
SIM3	Message slot 3 interrupt request mask bit		RW
SIM2	Message slot 2 interrupt request mask bit		RW
SIM1	Message slot 1 interrupt request mask bit		RW
SIM0	Message slot 0 interrupt request mask bit		RW

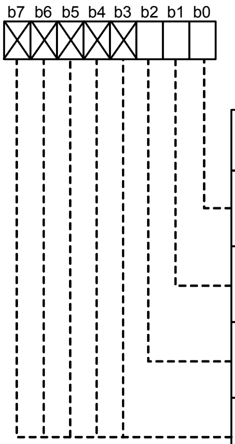
NOTES:
1. Set the CiSIMKR register while the CiMCTLj (j = 0 to 15) register of the corresponding message slot to the bit to be changed, is set to 00h.
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.15 C0SIMKR and C1SIMKR Registers

The CiSIMKR register determines whether an interrupt request generated by completing a transmit/receive operation in the corresponding message slot is enabled or disabled. When the SIMj bit (j = 0 to 15) is set to 1 (interrupt request enabled), an interrupt request generated by completing a transmit operation or a receive operation in the corresponding message slot is enabled. Refer to **23.4 CAN Interrupts** for details.

23.1.13 CANi Error Interrupt Mask Register (CiEIMKR Register) (i = 0, 1)

CANi Error Interrupt Mask Register (i = 0, 1)



Symbol	Address	After Reset(1)
C0EIMKR	0214h	XXXX X000b
C1EIMKR	0294h	XXXX X000b

Bit Symbol	Bit Name	Function	RW
BOIM	Bus-off interrupt mask bit	0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
EPIM	Error-passive interrupt mask bit	0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
BEIM	CAN bus-error interrupt mask bit	0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
– (b7-b3)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.16 C0EIMKR and C1EIMKR Registers

23.1.13.1 BOIM Bit

The BOIM bit determines whether an interrupt request is enabled or disabled when the CAN module is placed in a bus-off state. When the BOIM bit is set to 1, a bus-off interrupt request is enabled.

23.1.13.2 EPIM Bit

The EPIM bit determines whether an interrupt request is enabled or disabled when the CAN module is placed in an error passive state. When the EPIM bit is set to 1, an error passive interrupt request is enabled.

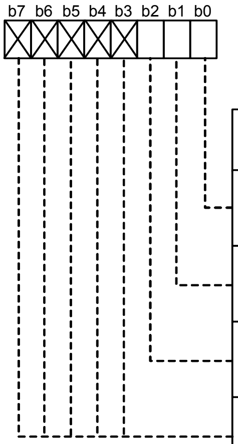
23.1.13.3 BEIM Bit

The BEIM bit determines whether an interrupt request is enabled or disabled when a CAN bus error occurs. When the BEIM bit is set to 1, a CAN bus error interrupt request is enabled.

Refer to **23.4 CAN Interrupts** for details.

23.1.14 CANi Error Interrupt Status Register (CiEISTR Register) (i = 0, 1)

CANi Error Interrupt Status Register (i = 0, 1)



Symbol	Address	After Reset(1)
C0EISTR	0215h	XXXX X000b
C1EISTR	0295h	XXXX X000b

Bit Symbol	Bit Name	Function	RW
BOIS	Bus-off interrupt status bit	0: Interrupt not requested 1: Interrupt requested(2)	RW
EPIS	Error-passive interrupt status bit	0: Interrupt not requested 1: Interrupt requested(2)	RW
BEIS	CAN bus-error interrupt status bit	0: Interrupt not requested 1: Interrupt requested(2)	RW
– (b7-b3)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.
2. Set each bit to 0 by a program. Writing a 1 has no effect.

Figure 23.17 C0EISTR and C1EISTR Registers

When using the CAN interrupt, the CiEISTR register determines an error interrupt source.
Bits BOIS, EPIS, and BEIS are not automatically set to 0 (interrupt not requested) even if the interrupt is acknowledged. Set these bits to 0 by a program.
Use the MOV instruction to set each bit in the CiEISTR register to 0. Write a 0 to the bit which is to set to 0, and write a 1 to the bit which is to remain unchanged.

For example: To set the BOIS bit in CAN0 to 0
mov.b #006h, C0EISTR

Refer to **23.4 CAN Interrupts** for details.

23.1.14.1 BOIS Bit

The BOIS bit becomes 1 when the CAN module is placed in a bus-off state.
NOTE:
1. Regardless of whether the BOIM bit in the CiEIMKR register is set to 0 (interrupt request masked) or 1 (interrupt request enabled), the BOIS bit becomes 1 when the CAN module becomes a bus-off state.

23.1.14.2 EPIS Bit

The EPIS bit becomes 1 when the CAN module is placed in an error passive state.
NOTE:
1. Regardless of whether the EPIM bit in the CiEIMKR register is set to 0 (interrupt request masked) or 1 (interrupt request enabled), the EPIS bit becomes 1 when the CAN module becomes an error-passive state.

23.1.14.3 BEIS Bit

The BEIS bit becomes 1 when a CAN bus error is detected.
NOTE:
1. Regardless of whether the BEIM bit in the CiEIMKR register is set to 0 (interrupt request masked) or 1 (interrupt request enabled), the BEIS bit becomes 1 when the CAN bus error is detected.

23.1.15 CANi Error Source Register (CiEFR Register) (i = 0, 1)

CANi Error Source Register (i = 0, 1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset(1)
C0EFR	0216h	00h
C1EFR	0296h	00h

Bit Symbol	Bit Name	Function	RW
ACKE	ACK error detect bit	0: ACK error not detected 1: ACK error detected(2)	RW
CRCE	CRC error detect bit	0: CRC error not detected 1: CRC error detected(2)	RW
FORME	FORM error detect bit	0: FORM error not detected 1: FORM error detected(2)	RW
STFE	Stuff error detect bit	0: Stuff error not detected 1: Stuff error detected(2)	RW
BITE0	Bit error detect bit 0	0: Bit error not detected while transmitting recessive "H" 1: Bit error detected while transmitting recessive "H"(2)	RW
BITE1	Bit error detect bit 1	0: Bit error not detected while transmitting dominant "L" 1: Bit error detected while transmitting dominant "L"(2)	RW
RCVE	Receive error detect bit	0: Error not detected while receiving 1: Error detected while receiving(2)	RW
TRE	Transmit error detect bit	0: Error not detected while receiving 1: Error detected while receiving(2)	RW

NOTES:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.
2. Set each bit to 0 by a program. Writing a 1 has no effect.

Figure 23.18 C0EFR and C1EFR Registers

The CiEFR register determines an error source when a CAN bus error occurs. Set each bit in the CiEFR register to 0 after reading the CiEFR register by a program.
Use the MOV instruction to set each bit in the CiEFR register to 0. Write a 0 to the bit which is to set to 0, and write a 1 to the bit which is to remain unchanged.

For example: To set the ACKE bit in CAN0 to 0
mov.b #0FEh, C0EFR

23.1.15.1 ACKE Bit

The ACKE bit becomes 1 when an ACK error is detected.

23.1.15.2 CRCE Bit

The CRC bit becomes 1 when a CRC error is detected.

23.1.15.3 FORME Bit

The FORME bit becomes 1 when a FORM error is detected.

23.1.15.4 STFE Bit

The STFE bit becomes 1 when a stuff error is detected.

23.1.15.5 BITE0 Bit

The BITE0 bit becomes 1 when a bit error is detected while transmitting recessive “H”.

23.1.15.6 BITE1 Bit

The BITE1 bit becomes 1 when a bit error is detected while transmitting dominant “L”.

23.1.15.7 RCVE Bit

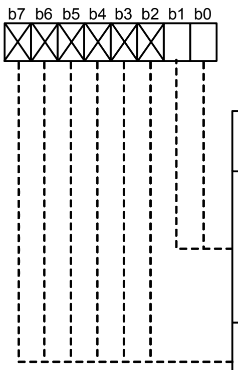
The RCVE bit becomes 1 when a CAN bus error is detected while receiving.

23.1.15.8 TRE Bit

The TRE bit becomes 1 when a CAN bus error is detected while transmitting.

23.1.16 CANi Mode Register (CiMDR Register) (i = 0, 1)

CANi Mode Register (i = 0, 1)(1)



Symbol	Address	After Reset(2)
C0MDR	0219h	XXXX XX00b
C1MDR	0299h	XXXX XX00b

Bit Symbol	Bit Name	Function	RW
CMOD	CAN operating mode select bit	b1 b0 0 0: Normal operating mode 0 1: Bus monitoring mode 1 0: Self-test mode 1 1: Do not set to this value	RW RW
– (b7-b2)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Set the CiMDR register while the STATE_RESET bit in the CiSTR register is 1 (CAN module is in reset).
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.19 C0MDR and C1MDR Registers

23.1.16.1 CMOD Bit

The CMOD bit selects a CAN operating mode.

- Normal operating mode: Normal transmit and receive operations are enabled.
- Bus monitoring mode(1): Only receive operation is enabled. Output signal from the CANiOUT pin is fixed to high level (“H”) in bus monitoring mode. The CAN module transmits neither ACK nor error frame.
- Self-test mode: The CAN module connects the CANiOUT pin to the CANiIN pin internally. The CAN module can communicate without additional device when using self-test mode and loop back mode. Output signal from the CANiOUT pin is fixed to “H” in self-test mode while transmitting. Figure 23.20 shows an image diagram in self-test mode.

NOTE:
1. Do not generate a transmit request in bus monitoring mode.
   The CAN module in bus monitoring mode considers dominant “L” is received regardless of whether the actual ACK bit is dominant “L” or recessive “H”. Therefore, when a transmit operation is completed until EOF, the CAN module determines a receive operation is successfully completed even if the ACK bit is recessive “H”.

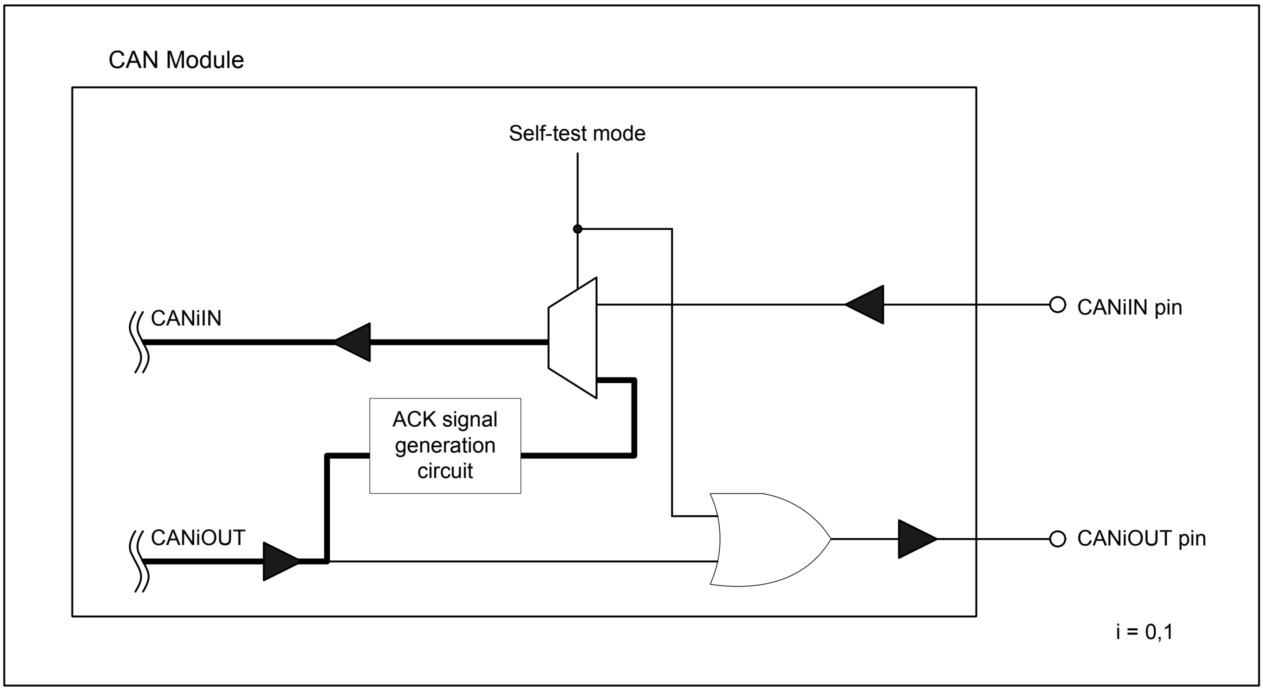
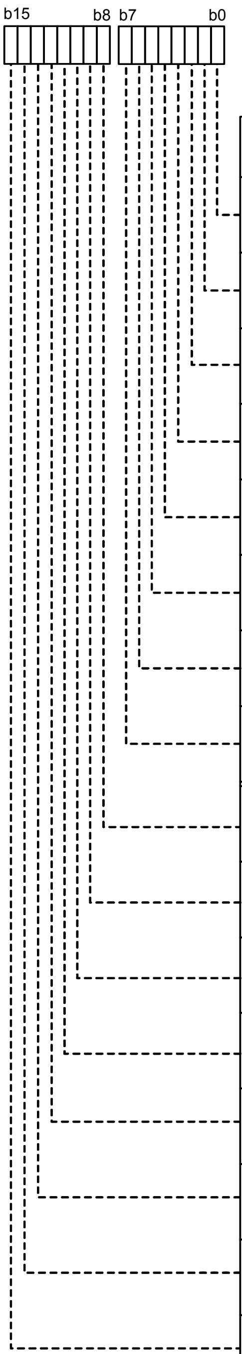


Figure 23.20 Self-Test Mode

23.1.17 CANi Single-Shot Control Register (CiSSCTLR Register) (i = 0, 1)

CANi Single-Shot Control Register (i = 0, 1)(1, 2)



Symbol	Address	After Reset(3)
C0SSCTLR	0221h - 0220h	0000h
C1SSCTLR	02A1h - 02A0h	0000h

Bit Symbol	Bit Name	Function	RW
SSC15	Message slot 15 single-shot control bit	0: Single-shot mode not used 1: Single-shot mode used	RW
SSC14	Message slot 14 single-shot control bit		RW
SSC13	Message slot 13 single-shot control bit		RW
SSC12	Message slot 12 single-shot control bit		RW
SSC11	Message slot 11 single-shot control bit		RW
SSC10	Message slot 10 single-shot control bit		RW
SSC9	Message slot 9 single-shot control bit		RW
SSC8	Message slot 8 single-shot control bit		RW
SSC7	Message slot 7 single-shot control bit		RW
SSC6	Message slot 6 single-shot control bit		RW
SSC5	Message slot 5 single-shot control bit		RW
SSC4	Message slot 4 single-shot control bit		RW
SSC3	Message slot 3 single-shot control bit		RW
SSC2	Message slot 2 single-shot control bit		RW
SSC1	Message slot 1 single-shot control bit		RW
SSC0	Message slot 0 single-shot control bit		RW

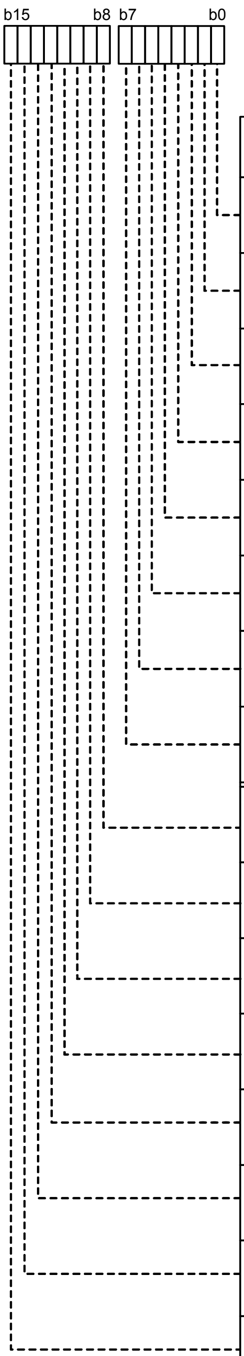
NOTES:
1. Set the CiSSCTLR register while the CiMCTLj register (j = 0 to 15) of the corresponding message slot to the bit to be changed, is set to 00h.
2. The CiSSCTLR register can be accessed when the BANKSEL bit in the CiCTLR1 register is set to 0 (message slot control register and single-shot register selected).
3. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 0.

Figure 23.21 C0SSCTLR and C1SSCTLR Registers

According to the CAN Specification 2.0 Part B, if a transmit operation is aborted due to the arbitration lost or transmit error, the CAN module continues retransmitting until the transmit operation is successfully completed. When a transmit operation is failed, the frame can be retransmitted if the SSCj bit (j = 0 to 15) in the CiSSCTLR register is set to 0, and the frame cannot be retransmitted if the SSCj bit is set to 1.

23.1.18 CANi Single-Shot Status Register (CiSSSTR Register) (i = 0, 1)

CANi Single-Shot Status Register (i = 0, 1)(1)



Symbol	Address	After Reset(2)
C0SSSTR	0225h - 0224h	0000h
C1SSSTR	02A5h - 02A4h	0000h

Bit Symbol	Bit Name	Function	RW
SSS15	Message slot 15 single-shot status bit	0: No Arbitration lost nor transmit error occurred 1: Arbitration lost or transmit error occurred(3)	RW
SSS14	Message slot 14 single-shot status bit		RW
SSS13	Message slot 13 single-shot status bit		RW
SSS12	Message slot 12 single-shot status bit		RW
SSS11	Message slot 11 single-shot status bit		RW
SSS10	Message slot 10 single-shot status bit		RW
SSS9	Message slot 9 single-shot status bit		RW
SSS8	Message slot 8 single-shot status bit		RW
SSS7	Message slot 7 single-shot status bit		RW
SSS6	Message slot 6 single-shot status bit		RW
SSS5	Message slot 5 single-shot status bit		RW
SSS4	Message slot 4 single-shot status bit		RW
SSS3	Message slot 3 single-shot status bit		RW
SSS2	Message slot 2 single-shot status bit		RW
SSS1	Message slot 1 single-shot status bit		RW
SSS0	Message slot 0 single-shot status bit		RW

NOTES:
1. The CiSSSTR register can be accessed when the BANKSEL bit in the CiCTLR1 is set to 0 (message slot control register and single-shot register selected).
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 0.
3. Set each bit to 0 by a program. Writing a 1 has no effect.

Figure 23.22 C0SSSTR and C1SSSTR Registers

If a transmit operation is aborted due to the arbitration lost or transmit error, the bit corresponding to the message slot j (j = 0 to 15) becomes 1. Set each bit in the CiSSSTR register to 0 after reading the CiSSSTR register by a program.
Use the MOV instruction to set the SSSj bit to 0. Write a 0 to the bit which is to set to 0, and write a 1 to the bit which is to remain unchanged.

For example: To set the SSS0 bit in CAN0 to 0

```
mov.w #07FFFh, C0SSSTR
```

23.1.19 CANi Global Mask Register, CANi Local Mask Register A, and CANi Local Mask Register B (CiGMRk, CiLMARk, and CiLMBRk Registers) (i = 0,1, k = 0 to 4)

CANi Global Mask Register Standard ID0(1) (i = 0, 1)
CANi Local Mask Register A Standard ID0(1)
CANi Local Mask Register B Standard ID0(1)

b7 b6 b5 b4 b3 b2 b1 b0

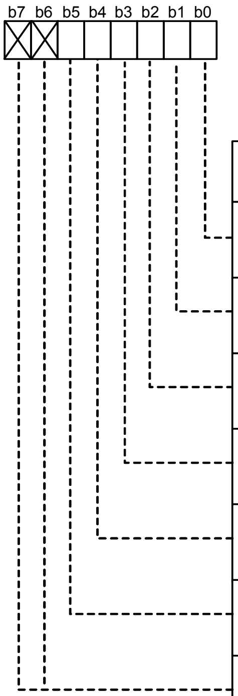
Symbol	Address	After Reset(2)
C0GMR0, C1GMR0	0228h, 02A8h	XXX0 0000b
C0LMAR0, C1LMAR0	0230h, 02B0h(3)	XXX0 0000b
C0LMBR0, C1LMBR0	0238h, 02B8h(4)	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
SID6M	Standard ID6	0: ID not checked 1: ID checked	RW
SID7M	Standard ID7		RW
SID8M	Standard ID8		RW
SID9M	Standard ID9		RW
SID10M	Standard ID10		RW
– (b7-b5)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Registers CiGMR0, CiLMAR0, and CiLMBR0 can be accessed when the BANKSEL bit in the CiCTLR1 register is set to 1 (mask register selected).
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 1.
3. The C0LMAR0 register shares the same address with the C0MCTL0 register, and the C1MAR0 register with the C1MCTL0 register.
4. The C0LMBR0 register shares the same address with the C0MCTL8 register, and the C1LMBR0 register with the C1MCTL8 register.

Figure 23.23 C0GMR0, C1GMR0, C0LMAR0, C1LMAR0, C0LMBR0, and C1LMBR0 Registers

CANi Global Mask Register Standard ID1[1] (i = 0, 1)
CANi Local Mask Register A Standard ID1[1]
CANi Local Mask Register B Standard ID1[1]



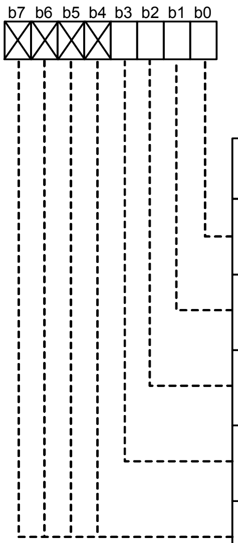
Symbol	Address	After Reset[2]
C0GMR1, C1GMR1	0229h, 02A9h	XX00 0000b
C0LMAR1, C1LMAR1	0231h, 02B1h[3]	XX00 0000b
C0LMBR1, C1LMBR1	0239h, 02B9h[4]	XX00 0000b

Bit Symbol	Bit Name	Function	RW
SID0M	Standard ID0	0: ID not checked 1: ID checked	RW
SID1M	Standard ID1		RW
SID2M	Standard ID2		RW
SID3M	Standard ID3		RW
SID4M	Standard ID4		RW
SID5M	Standard ID5		RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Registers CiGMR1, CiLMAR1, and CiLMBR1 can be accessed when the BANKSEL bit in the CiCTLR1 register is set to 1 (mask register selected).
2. Value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 1.
3. The C0LMAR1 register shares the same address with the C0MCTL1 register, and the C1LMAR1 register with the C1MCTL1 register.
4. The C0LMBR1 register shares the same address with the C0MCTL9 register, and the C1LMBR1 register with the C1MCTL9 register.

Figure 23.24 C0GMR1, C1GMR1, C0LMAR1, C1LMAR1, C0LMBR1, and C1LMBR1 Registers

CANi Global Mask Register Extended ID0(1) (i = 0, 1)
CANi Local Mask Register A Extended ID0(1)
CANi Local Mask Register B Extended ID0(1)



Symbol	Address	After Reset(2)
C0GMR2, C1GMR2	022Ah, 02AAh	XXXX 0000b
C0LMAR2, C1LMAR2	0232h, 02B2h(3)	XXXX 0000b
C0LMBR2, C1LMBR2	023Ah, 02BAh(4)	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
EID14M	Extended ID14	0: ID not checked 1: ID checked	RW
EID15M	Extended ID15		RW
EID16M	Extended ID16		RW
EID17M	Extended ID17		RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Registers CiGMR2, CiLMAR2, and CiLMBR2 can be accessed when the BANKSEL bit in the CiCTLR1 register is set to 1 (mask register selected).
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 1.
3. The C0LMAR2 register shares the same address with the C0MCTL2 register, and the C1LMAR2 register with the C1MCTL2 register.
4. The C0LMBR2 register shares the same address with the C0MCTL10 register, and the C1LMBR2 register with the C1MCTL10 register.

Figure 23.25 C0GMR2, C1GMR2, C0LMAR2, C1LMAR2, C0LMBR2, and C1LMBR2 Registers

CANi Global Mask Register Extended ID1(1) (i = 0,1)
CANi Local Mask Register A Extended ID1(1)
CANi Local Mask Register B Extended ID1(1)

b7 b6 b5 b4 b3 b2 b1 b0

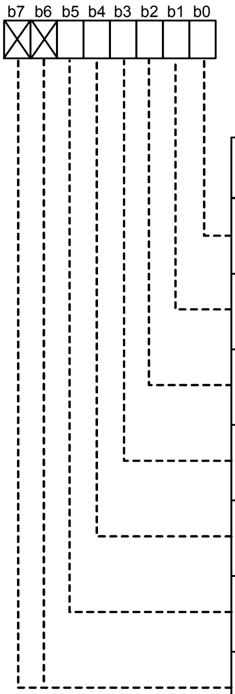
Symbol	Address	After Reset(2)
C0GMR3, C1GMR3	022Bh, 02ABh	00h
C0LMAR3, C1LMAR3	0233h, 02B3h(3)	00h
C0LMBR3, C1LMBR3	023Bh, 02BBh(4)	00h

Bit Symbol	Bit Name	Function	RW
EID6M	Extended ID6	0: ID not checked 1: ID checked	RW
EID7M	Extended ID7		RW
EID8M	Extended ID8		RW
EID9M	Extended ID9		RW
EID10M	Extended ID10		RW
EID11M	Extended ID11		RW
EID12M	Extended ID12		RW
EID13M	Extended ID13		RW

NOTES:
1. Registers CiGMR3, CiLMAR3, and CiLMBR3 can be accessed when the BANKSEL bit in the CiCTLR1 register is set to 1 (mask register selected).
2. Value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 1.
3. The C0LMAR3 register shares the same address with the C0MCTL3 register, and the C1LMAR3 register with the C1MCTL3 register.
4. The C0LMBR3 register shares the same address with the C0MCTL11 register, and the C1LMBR3 register with the C1MCTL11 register.

Figure 23.26 C0GMR3, C1GMR3, C0LMAR3, C1LMAR3, C0LMBR3, and C1LMBR3 Registers

CANi Global Mask Register Extended ID2(1) (i = 0, 1)
CANi Local Mask Register A Extended ID2(1)
CANi Local Mask Register B Extended ID2(1)



Symbol	Address	After Reset (2)
C0GMR4, C1GMR4	022Ch, 02ACh	XX00 0000b
C0LMAR4, C1LMAR4	0234h, 02B4h(3)	XX00 0000b
C0LMBR4, C1LMBR4	023Ch, 02BCh(4)	XX00 0000b

Bit Symbol	Bit Name	Function	RW
EID0M	Extended ID0	0: ID not checked 1: ID checked	RW
EID1M	Extended ID1		RW
EID2M	Extended ID2		RW
EID3M	Extended ID3		RW
EID4M	Extended ID4		RW
EID5M	Extended ID5		RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. Registers CiGMR4, CiLMAR4, and CiLMBR4 can be accessed when the BANKSEL bit in the CiCTLR1 register is set to 1 (mask register selected).
2. Value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 1.
3. The C0LMAR4 register shares the same address with the C0MCTL4 register, and the C1LMAR4 register with the C1MCTL4 register.
4. The C0LMBR4 register shares the same address with the C0MCTL12 register, and the C1LMBR4 register with the C1MCTL12 register.

Figure 23.27 C0GMR4, C1GMR4, C0LMAR4, C1LMAR4, C0LMBR4, and C1LMBR4 Registers

Registers CiGMRk, CiLMARk, and CiLMBRk (i = 0, 1, k = 0 to 4) are used for acceptance filtering.
By using these registers, users are able to select which messages to receive.
The CiGMRk register determines whether IDs in the message slots 0 to 13 are checked or not. The CiLMARk register determines whether ID in the message slot 14 is checked or not. The CiLMBRk register determines whether ID in the message slot 15 is checked or not.

- When the bit in the CiGMRk, CiLMARk, or CiLMBRk register is set to 0, the corresponding bit (ID bit) in the CANi message slot j's (j = 0 to 15) standard ID0, standard ID1, or extended ID0 to extended ID2 is masked in acceptance filtering. (The corresponding bit is assumed to have a matching ID.)
- When the bit in the CiGMRk, CiLMARk, CiLMBRk register is set to 1, the corresponding ID bit is compared with a received ID in acceptance filtering. When the received ID matches the ID set in the message slot j, the receive data is stored into the message slot having the matched ID.

NOTES:
1. Change the CiGMRk register while none of the message slots 0 to 13 has a receive request.
2. Change the CiLMARk register while the message slot 14 has no receive request.
3. Change the CiLMBRk register while the message slot 15 has no receive request.
4. When there are two or more receive message slots which have the matched ID with the received message, the received message is stored into the smallest-numbered message slot.

Figure 23.28 shows individual mask registers and corresponding message slots. Figure 23.29 shows the acceptance filtering.

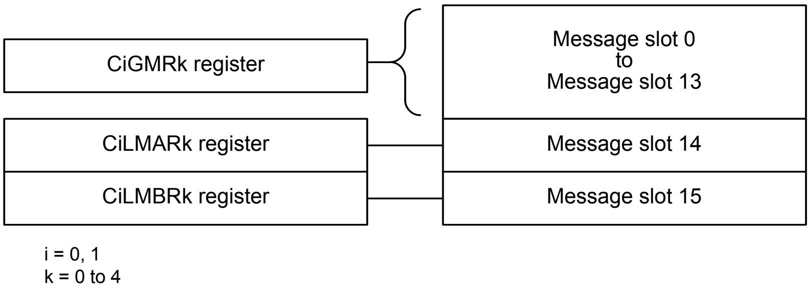


Figure 23.28 Individual Mask Registers and Message Slots

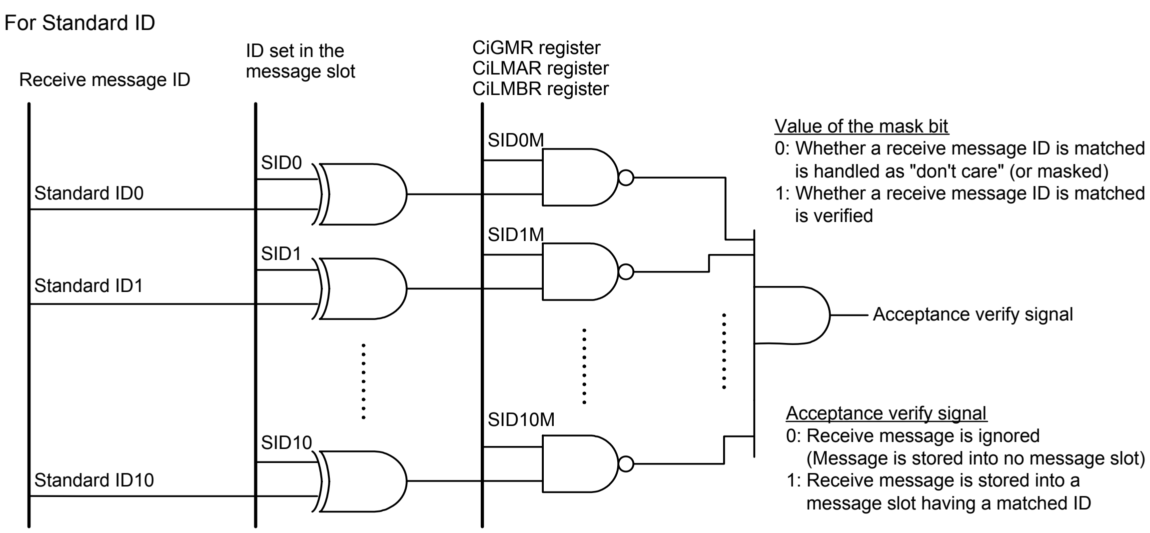


Figure 23.29 Acceptance Filtering

23.1.20 CANi Message Slot j Control Register (CiMCTLj Register) (i = 0, 1, j = 0 to 15)

CANi Message Slot j Control Register (i = 0,1, j = 0 to 15)(1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset(2)
C0MCTL0 to C0MCTL15	0230h - 023Fh(3)	00h
C1MCTL0 to C1MCTL15	02B0h - 02BFh(4)	00h

Bit Symbol	Bit Name	Function		RW
When transmitting, SENTDATA When receiving, NEWDATA	Transmit complete flag Receive complete flag	When transmitting 0: Transmit operation not completed 1: Transmit operation completed(5)	When receiving 0: Receive operation not completed 1: Receive operation completed(5)	RW
When transmitting, TRMACTIVE When receiving, INVALDATA	Transmitting flag Receiving flag	When transmitting 0: Not transmitting 1: Transmitting	When receiving 0: Not storing receive data 1: Storing receive data	RO
MSGLOST	Overwrite flag	0: Overwrite not occurred 1: Overwrite occurred(5)		RW
REMACTIVE	Remote frame transmit/receive status flag	Not in BasicCAN mode 0: Data frame 1: Remote frame In BasicCAN mode(6) 0: Data frame received (status) 1: Remote frame received (status)		RO
RSPLOCK	Automatic answering disable mode select bit	0: Automatic answering to the remote frame enabled 1: Automatic answering to the remote frame disabled		RW
REMOTE	Remote frame set bit	0: Data frame transmitted/received 1: Remote frame transmitted/received		RW
RECREQ	Receive request bit	0: Receive operation not requested 1: Receive operation requested(7)		RW
TRMREQ	Transmit request bit	0: Transmit operation not requested 1: Transmit operation requested(7)		RW

NOTES:
1. The CiMCTLj register can be accessed when the BANKSEL bit in the CiCTLR1 register is set to 0 (message slot control register and single-shot register selected).
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset, supplying the clock to the CAN module, and setting the BANKSEL bit to 0.
3. Registers C0MCTL0 to C0MCTL4 share addresses with registers C0LMAR0 to C0LMAR4, and registers C0MCTL8 to C0MCTL12 with registers C0LMBR0 to C0LMBR4 respectively.
4. Registers C1MCTL0 to C1MCTL4 share addresses with registers C1LMAR0 to C1LMAR4, and registers C1MCTL8 to C1MCTL12 with registers C1LMBR0 to C1LMBR4 respectively.
5. Set the bit to 0 by a program. Writing a 1 has no effect.
6. BasicCAN mode can be used with the message slot 14 and 15.
7. Do not set both the RECREQ and TRMREQ bits to 1 simultaneously.

Figure 23.30 C0MCTL0 to C0MCTL15 and C1MCTL0 to C1MCTL15 Registers

Table 23.4 CiMCTLj Register (i = 0, 1, j = 0 to 15) Settings for Transmit/Receive Operation

Bit Setting in the CiMCTLj Register						Transmit/Receive Operation Mode
TRMREQ	RECREQ	REMOTE	RSPLOCK	MSGLOST	SENTDATA NEWDATA	
0	0	0	0	0	0	No transmit nor receive operation
0	1	0	0	0	0	Data frame receive operation
0	1	1	1	0	0	Remote frame receive operation
0	1	1	0	0	0	Remote frame receive operation (Data frame is transmitted after remote frame is received.)
1	0	0	0	0	0	Data frame transmit operation
1	0	1	0	0	0	Remote frame transmit operation (Data frame is received after remote frame is transmitted.)

23.1.20.1 SENTDATA/NEWDATA Bit

The SENTDATA/NEWDATA bit indicates CAN message transmit/receive operation is completed. Set the SENTDATA/NEWDATA bit to 0 (transmit/receive operation not completed) by a program prior to transmitting or receiving. The SENTDATA/NEWDATA bit is not set to 0 automatically. While the TRMACTIVE/INVALDATA bit is 1 (transmitting or storing receive data), the SENTDATA/NEWDATA bit cannot be set to 0.

SENTDATA: The SENTDATA bit becomes 1 (transmit operation completed) when a transmit operation is completed in the transmit message slot.

NEWDATA: The NEWDATA bit becomes 1 (receive operation completed) after the message to be stored into the message slot j (j = 0 to 15) is successfully received.

NOTES:

1. To read a receive data from the message slot j, set the NEWDATA bit to 0 before reading. If the NEWDATA bit becomes 1 while reading the message slot, this indicates that new receive data has been stored into the message slot while reading and the returned data contains an undefined value. In this case, discard the data with an undefined value and then read the message slot again after setting the NEWDATA bit to 0.
2. When the remote frame is transmitted or received, the SENTDATA/NEWDATA bit remains unchanged even after remote frame transmit or receive operation is completed. The SENTDATA/NEWDATA bit becomes 1 when the following data frame transmit or receive operation is completed.

23.1.20.2 TRMACTIVE/INVALDATA Bit

The TRMACTIVE/INVALDATA bit indicates that the CAN protocol controller is accessing the message slot j. The TRMACTIVE/INVALDATA bit becomes 1 when the controller is accessing, and becomes 0 when not accessing.

TRMACTIVE: The TRMACTIVE bit becomes 1 (transmitting) when a transmit operation is started. The TRMACTIVE bit becomes 0 (not transmitting) when the CAN module loses arbitration, a CAN bus error occurs, or when a transmit operation is completed.

INVALDATA: The INVALDATA bit becomes 1 (storing receive data) while the received message is being stored into the message slot j after the receive operation is completed. The INVALDATA bit becomes 0 (not storing receive data) when the receive data has been stored. While the INVALDTA bit is 1, a value read from the message slot j is undefined.

23.1.20.3 MSGLOST Bit

The MSGLOST bit is enabled when the data frame receive operation or remote frame transmit operation (data frame is received after the remote frame is transmitted) shown in Table 23.4 is selected. The MSGLOST bit becomes 1 (overwrite occurred) when the message slot j (j = 0 to 15) is overwritten by a new receive data while the NEWDATA bit is 1 (receive operation completed).
Set the MSGLOST bit to 0 (overwrite not occurred) after reading it by a program.

23.1.20.4 REMACTIVE Bit

The REMACTIVE bit becomes 1 (remote frame) when the message slot j is set for the remote frame transmit or receive operation, while the STATE_BASICCAN bit in the CiSTR register is 0 (not in BasicCAN mode). Then, the REMACTIVE bit becomes 0 (data frame) after the remote frame transmit or receive operation is completed.
In BasicCAN mode, the REMACTIVE bit in the CiMCTL14 or CiMCTL15 register becomes 0 when the data frame is received, and becomes 1 when the remote frame is received.

23.1.20.5 RSPLOCK Bit

The RSPLOCK bit is enabled when the remote frame receive operation shown in Table 23.4 is selected. The RSPLOCK bit determines the operation after the remote frame is received.
When the RSPLOCK bit is set to 0 (automatic answering to remote frame enabled), a slot automatically switches to a transmit slot after the remote frame is received and the message set in the message slot is automatically transmitted as the data frame.
When the RSPLOCK bit is set to 1 (automatic answering to remote frame disabled), the message is not automatically transmitted after the remote frame is received.
Set the RSPLOCK bit to 0 when any transmit/receive mode other than remote frame receive mode is selected.

23.1.20.6 REMOTE Bit

The REMOTE bit determines transmit/receive mode shown in Table 23.4. Set the REMOTE bit to 0 to transmit or receive the data frame. Set it to 1 to transmit or receive the remote frame.
The following occurs when the remote frame is transmitted or received.

- Transmitting the remote frame
  A message set in the message slot j is transmitted as the remote frame. After a transmit operation is completed, the slot automatically switches to a data frame receive message slot.
  If the data frame is received before a remote frame transmit operation is completed, the data frame is stored into the message slot j and the remote frame is not transmitted.
- Receiving the remote frame
  The message slot receives the remote frame. The RSPLOCK bit determines the operation after the remote frame is received.

23.1.20.7 RECREQ Bit

The RECREQ bit determines transmit/receive mode shown in Table 23.4. When the RECREQ bit is set to 1 (receive operation requested), the message slot is set to receive the data frame or remote frame. If the REMOTE bit is set to 1 (remote frame transmitted/received) and the RSPLOCK bit is set to 0 (automatic answering to the remote frame enabled), the data frame is transmitted automatically after the remote frame is received, regardless of the RECREQ bit setting to 0.
Set the RECREQ bit to 0 (receive operation not requested) to transmit the data frame or remote frame.
Do not set both the TRMREQ and RECREQ bits in the same message slot to 1.

23.1.20.8 TRMREQ Bit

The TRMREQ bit determines transmit/receive mode shown in Table 23.4. When the TRMREQ bit is set to 1 (transmit operation requested), the data frame or remote frame is transmitted. If the REMOTE bit is set to 0 (remote frame transmitted/received), the message slot automatically switches to a receive slot for the data frame after the remote frame is transmitted, regardless of the TRMREQ bit setting to 1.
Set the TRMREQ bit to 0 (transmit operation not requested) to receive the data frame or remote frame.
Do not set both the TRMREQ and RECREQ bits in the same message slot to 1.

NOTES:

1. When a transmit operation request occurs in multiple message slots, the data frame or remote frame in the slot which has the smallest slot number is transmitted first.
2. In single-shot mode, if a transmit operation is aborted due to the arbitration lost or transmit error, the value in the CiMCTLj register is cleared to 00h.

23.1.21 CANi Slot Buffer Select Register (CiSBS Register) (i = 0, 1)

CANi Slot Buffer Select Register (i = 0, 1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset(2)
C0SBS	0240h	00h
C1SBS	0250h	00h

Bit Symbol	Bit Name	Function	RW
SBS00	CANi message slot buffer 0 number select bit(1)	b3 b2 b1 b0 0 0 0 0: Message slot 0 0 0 0 1: Message slot 1 0 0 1 0: Message slot 2 0 0 1 1: Message slot 3 . . . . . . 1 1 0 0: Message slot 12 1 1 0 1: Message slot 13 1 1 1 0: Message slot 14 1 1 1 1: Message slot 15	RW
SBS01			RW
SBS02			RW
SBS03			RW
SBS10	CANi message slot buffer 1 number select bit(1)	b7 b6 b5 b4 0 0 0 0: Message slot 0 0 0 0 1: Message slot 1 0 0 1 0: Message slot 2 0 0 1 1: Message slot 3 . . . . . . 1 1 0 0: Message slot 12 1 1 0 1: Message slot 13 1 1 1 0: Message slot 14 1 1 1 1: Message slot 15	RW
SBS11			RW
SBS12			RW
SBS13			RW

NOTES:
1. 16 CANi message slots are provided. Each message slot can be selected as a transmit message slot or receive message slot.
2. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

Figure 23.31 C0SBS and C1SBS Registers

23.1.21.1 SBS03 to SBS00 Bits

The message slot j (j = 0 to 15), selected with bits SBS03 to SBS00, is allocated to the CANi message slot buffer 0. The message slot j can be accessed via the allocated addresses (CAN0: addresses 01E0h to 01EFh; CAN1: 0260h to 026Fh).

23.1.21.2 SBS13 to SBS10 Bits

The message slot j, selected with bits SBS13 to SBS10, is allocated to the CANi message slot buffer 1. The message slot j can be accessed via the allocated addresses (CAN0: addresses 01F0h to 01FFh; CAN1: 0270h to 027Fh).

23.1.22 CANi Message Slot Buffer j (i = 0, 1; j = 0, 1)

CANi Message Slot Buffer j Standard ID0 (i = 0, 1; j = 0, 1)(1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
C0SLOT0_0, C0SLOT1_0	01E0h, 01F0h	Undefined
C1SLOT0_0, C1SLOT1_0	0260h, 0270h	Undefined

Bit Symbol	Bit Name	Function	RW
SID6	Standard ID6	Read or write standard ID6 in the message slot k (k = 0 to 15)	RW
SID7	Standard ID7	Read or write standard ID7 in the message slot k	RW
SID8	Standard ID8	Read or write standard ID8 in the message slot k	RW
SID9	Standard ID9	Read or write standard ID9 in the message slot k	RW
SID10	Standard ID10	Read or write standard ID10 in the message slot k	RW
– (b7-b5)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. Use the CiSBS register to select the message slot k which is accessed through the CiSLOTj_0 register.

CANi Message Slot Buffer j Standard ID1 (i = 0, 1; j = 0, 1)(1)

b7 b6 b5 b4 b3 b2 b1 b0

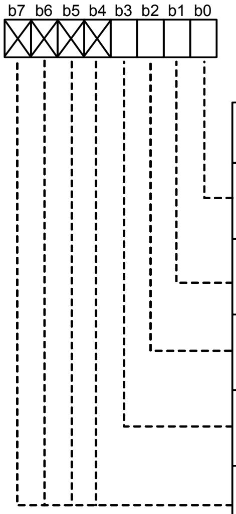
Symbol	Address	After Reset
C0SLOT0_1, C0SLOT1_1	01E1h, 01F1h	Undefined
C1SLOT0_1, C1SLOT1_1	0261h, 0271h	Undefined

Bit Symbol	Bit Name	Function	RW
SID0	Standard ID0	Read or write standard ID0 in the message slot k (k = 0 to 15)	RW
SID1	Standard ID1	Read or write standard ID1 in the message slot k	RW
SID2	Standard ID2	Read or write standard ID2 in the message slot k	RW
SID3	Standard ID3	Read or write standard ID3 in the message slot k	RW
SID4	Standard ID4	Read or write standard ID4 in the message slot k	RW
SID5	Standard ID5	Read or write standard ID5 in the message slot k	RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. Use the CiSBS register to select the message slot k which is accessed through the CiSLOTj_1 register.

Figure 23.32 C0SLOT0_0, C0SLOT1_0, C1SLOT0_0, and C1SLOT1_0 Registers, C0SLOT0_1, C0SLOT1_1, C1SLOT0_1, and C1SLOT1_1 Registers

CANi Message Slot Buffer j Extended ID0 (i = 0, 1; j = 0,1)[1][2]

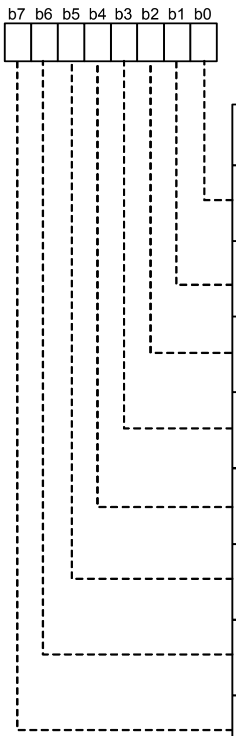


Symbol	Address	After Reset
C0SLOT0_2, C0SLOT1_2	01E2h, 01F2h	Undefined
C1SLOT0_2, C1SLOT1_2	0262h, 0272h	Undefined

Bit Symbol	Bit Name	Function	RW
EID14	Extended ID14	Read or write extended ID14 in the message slot k (k = 0 to 15)	RW
EID15	Extended ID15	Read or write extended ID15 in the message slot k	RW
EID16	Extended ID16	Read or write extended ID16 in the message slot k	RW
EID17	Extended ID17	Read or write extended ID17 in the message slot k	RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. If a receive slot is standard ID formatted, bits EID17 to EID14 are undefined when receive data is stored.
2. Use the CiSBS register to select the message slot k which is accessed through the CiSLOTj_2 register.

CANi Message Slot Buffer j Extended ID1 (i = 0,1; j = 0, 1)[1][2]



Symbol	Address	After Reset
C0SLOT0_3, C0SLOT1_3	01E3h, 01F3h	Undefined
C1SLOT0_3, C1SLOT1_3	0263h, 0273h	Undefined

Bit Symbol	Bit Name	Function	RW
EID6	Extended ID6	Read or write extended ID6 in the message slot k (k = 0 to 15)	RW
EID7	Extended ID7	Read or write extended ID7 in the message slot k	RW
EID8	Extended ID8	Read or write extended ID8 in the message slot k	RW
EID9	Extended ID9	Read or write extended ID9 in the message slot k	RW
EID10	Extended ID10	Read or write extended ID10 in the message slot k	RW
EID11	Extended ID11	Read or write extended ID11 in the message slot k	RW
EID12	Extended ID12	Read or write extended ID12 in the message slot k	RW
EID13	Extended ID13	Read or write extended ID13 in the message slot k	RW

NOTES:
1. If a receive slot is standard ID formatted, bits EID13 to EID6 are undefined when receive data is stored.
2. Use the CiSBS register to select the message slot k which is accessed through the CiSLOTj_3 register.

Figure 23.33 C0SLOT0_2, C0SLOT1_2, C1SLOT0_2, and C1SLOT1_2 Registers, C0SLOT0_3, C0SLOT1_3, C1SLOT0_3, and C1SLOT1_3 Registers

CANi Message Slot Buffer j Extended ID2 (i = 0, 1; j = 0, 1)(1)(2)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
C0SLOT0_4, C0SLOT1_4	01E4h, 01F4h	Undefined
C1SLOT0_4, C1SLOT1_4	0264h, 0274h	Undefined

Bit Symbol	Bit Name	Function	RW
EID0	Extended ID0	Read or write extended ID0 in the message slot k (k = 0 to 15)	RW
EID1	Extended ID1	Read or write extended ID1 in the message slot k	RW
EID2	Extended ID2	Read or write extended ID2 in the message slot k	RW
EID3	Extended ID3	Read or write extended ID3 in the message slot k	RW
EID4	Extended ID4	Read or write extended ID4 in the message slot k	RW
EID5	Extended ID5	Read or write extended ID5 in the message slot k	RW
– (b7-b6)	Unimplemented. Write 0. Read as undefined value.		–

NOTES:
1. If a receive slot is standard ID formatted, bits EID5 to EID0 are undefined when received data is stored.
2. Use the CiSBS register to select the message slot k which is accessed through the CiSLOTj_4 register.

CANi Message Slot Buffer j Data Length Code (i = 0, 1; j = 0, 1)(1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
C0SLOT0_5, C0SLOT1_5	01E5h, 01F5h	Undefined
C1SLOT0_5, C1SLOT1_5	0265h, 0275h	Undefined

Bit Symbol	Bit Name	Function	RW
DLC0	Data length set bit	Read or write the data length set bit in the message slot k (k = 0 to 15)	RW
DLC1			RW
DLC2			RW
DLC3			RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. Use the CiSBS register to select the message slot k which is accessed through the CiSLOTj_5 register.

Figure 23.34 C0SLOT0_4, C0SLOT1_4, C1SLOT0_4, and C1SLOT1_4 Registers C0SLOT0_5, C0SLOT1_5, C1SLOT0_5, and C1SLOT1_5 Registers

CANi Message Slot Buffer j Data m (i = 0, 1; j = 0, 1; m = 0 to 7)(1)(2)

b7 b0

Symbol	Address	After Reset
C0SLOT0_6 to C0SLOT0_13	01E6h - 01EDh	Undefined
C0SLOT1_6 to C0SLOT1_13	01F6h - 01FDh	Undefined
C1SLOT0_6 to C1SLOT0_13	0266h - 026Dh	Undefined
C1SLOT1_6 to C1SLOT1_13	0276h - 027Dh	Undefined

Function	Setting Range	RW
Read or write data *m* in the message slot *k* (*k* = 0 to 15)	00h to FFh	RW

NOTES:
1. Use the CiSBS register to select data *m* in the message slot k which is accessed through registers CiSLOTj_6 to CiSLOTj_13.
2. When a data frame receive operation is selected, data that exceeds the received data is undefined.

CANi Message Slot Buffer j Time Stamp High-Ordered (i = 0, 1; j = 0, 1)(1)

b7 b0

Symbol	Address	After Reset
C0SLOT0_14, C0SLOT1_14	01EEh, 01FEh	Undefined
C1SLOT0_14, C1SLOT1_14	026Eh, 027Eh	Undefined

Function	Setting Range	RW
Read or write time stamp high-ordered in the message slot k (*k* = 0 to 15)	00h to FFh	RW

NOTE:
1. Use the CiSBS register to select the time stamp high-ordered in the message slot *k* which is accessed through the CiSLOTj_14 register.

CANi Message Slot Buffer j Time Stamp Low-Ordered (i = 0, 1; j = 0, 1)(1)

b7 b0

Symbol	Address	After Reset
C0SLOT0_15, C0SLOT1_15	01EFh, 01FFh	Undefined
C1SLOT0_15, C1SLOT1_15	026Fh, 027Fh	Undefined

Function	Setting Range	RW
Read or write time stamp low-ordered in the message slot *k* (*k* = 0 to 15)	00h to FFh	RW

NOTE:
1. Use the CiSBS register to select the time stamp low-ordered in the message slot *k* which is accessed through the CiSLOTj_15 register.

Figure 23.35 C0SLOT0_6 to C0SLOT0_13, C0SLOT1_6 to C0SLOT1_13, C1SLOT0_6 to C1SLOT0_13, and C1SLOT1_6 to C1SLOT1_13 Registers, C0SLOT0_14, C0SLOT1_14, C1SLOT0_14, and C1SLOT1_14 Registers C0SLOT0_15, C0SLOT1_15, C1SLOT0_15, and C1SLOT1_15 Registers

The value in the message slot selected by the CiSBS register is returned by reading the message slot buffer. When writing to the message slot buffer, the value can be written in the message slot selected by the CiSBS register.
Write to the message slot k (k = 0 to 15) while the corresponding CiMCTLk register is set to 00h.

23.1.23 CANi Acceptance Filter Support Register (CiAFS Register) (i = 0, 1)

CANi Acceptance Filter Support Register (i = 0,1)

Symbol	Address	After Reset(1)
C0AFS	0245h - 0244h	0100h
C1AFS	0255h - 0254h	0100h

b15 b8 b7 b0

Function	Setting Range	RW
Data to determine the received ID is generated	0000h to FFFFh	RW

NOTE:
1. The value is obtained by setting the SLEEP bit in the CiSLPR register to 1 (sleep mode exited) after reset and supplying the clock to the CAN module.

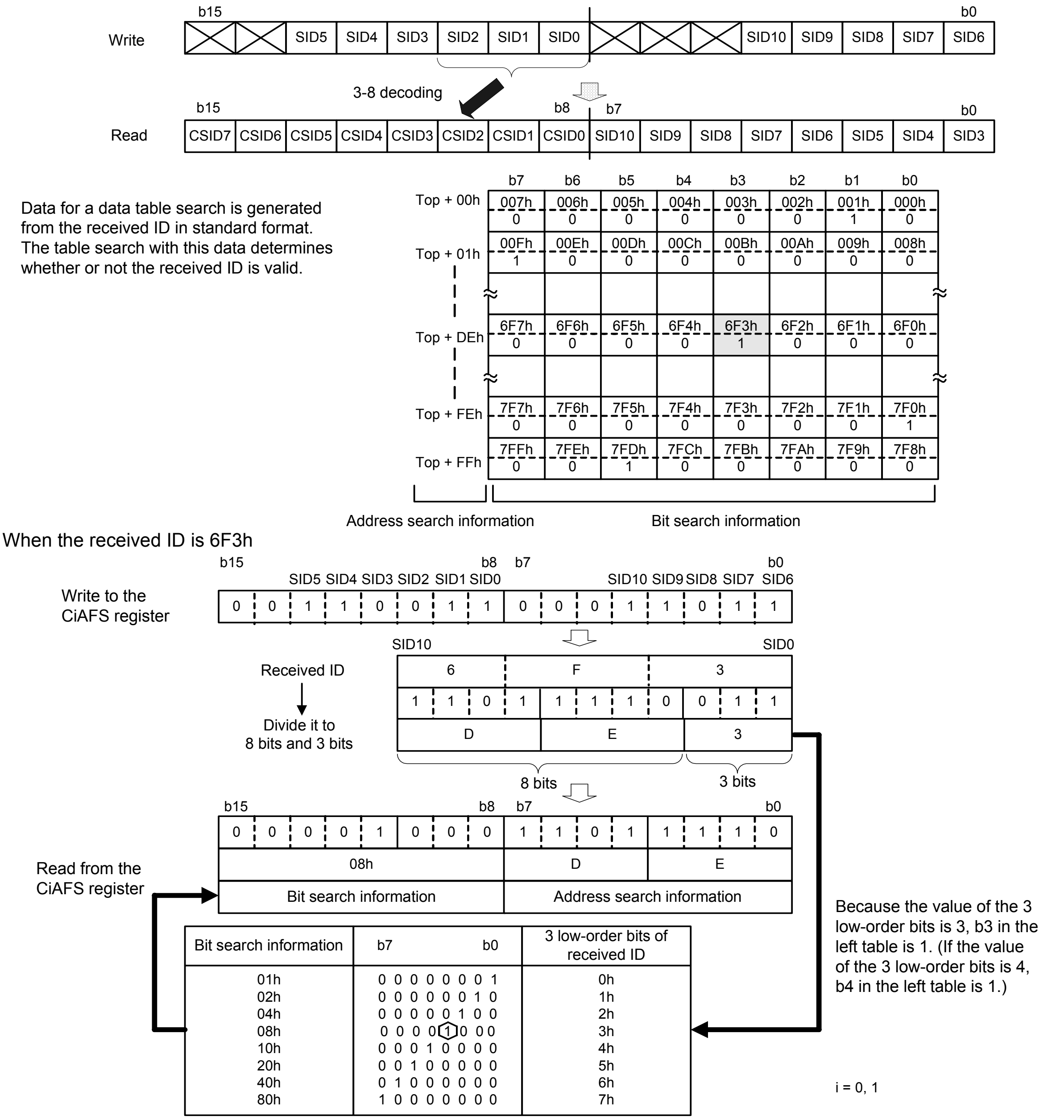


Figure 23.36 C0AFS and C1AFS Registers

The CiAFS register enables prompt performance of a table search to determine the validity of the received ID. This function is for standard-formatted ID only.

23.2 CAN Clock and CPU Clock

23.2.1 CAN Clock

The CAN clock is an operating clock for the CAN module. f1 is selected as the CAN clock when the PM25 bit in the PM2 register is set to 0. fCAN is selected as the CAN clock when the PM25 bit is set to 1. Set the PM25 bit while the SLEEP bit in CiSLPR register (i = 0, 1) is set to 0 (sleep mode).

23.2.2 CPU Clock

Follow the procedure below before accessing the CAN-associated registers.

- When the PM25 bit is set to 0 (f1):

(1) Set the PM24 bit to 0 (CPU clock is selected by the CM07 bit).
(2) Set the CM21 bit in the CM2 register to 0 (CPU clock is selected by the CM17 bit).
(3) Set bits MCD4 to MCD0 to 10010b (no division).
(4) Set the PM13 bit in the PM1 register to 1 (2 wait states).

- When the PM25 bit is set to 1 (fCAN):

(1) Set the PM24 bit to 1 (CPU clock is selected by the CM07 bit).
(2) Set the PM13 bit in the PM1 register to 1 (2 wait states).
(3) Wait for the time to switch clock.(1)

Do not enter wait mode or stop mode when the PM24 bit is set to 1.

NOTE:

1. The wait time to switch clock varies depending on the CPU clock frequency before and after the PM24 bit is changed.

- High frequency: Higher frequency compared “before the PM24 bit changes” with “after the PM24 bit changes”
- Low frequency: Lower frequency compared “before the PM24 bit changes” with “after the PM24 bit changes”

$$\text{Wait time to switch clock} \geq \frac{2 \text{ x High frequency}}{\text{Low frequency}} \text{ cycles}$$

23.3 Setting and Timing in CAN-Associated Registers

23.3.1 CAN Module Initialize Timing

Figure 23.37 shows an operation example when the CAN module is initialized.

(1) The CAN module can be initialized when the STATE_RESET bit in the CiSTR register (i = 0, 1) becomes 1 (CAN module is in reset) after both the RESET1 and RESET0 bits in the CiCTLR0 register are set to 1 (CAN module is reset).

(2) Set necessary CAN-associated registers.

(3) CAN communication can be established again when the STATE_RESET bit becomes 0 (CAN module is not in reset) after both the RESET1 and RESET0 bits are set to 0 (CAN module is out of reset).

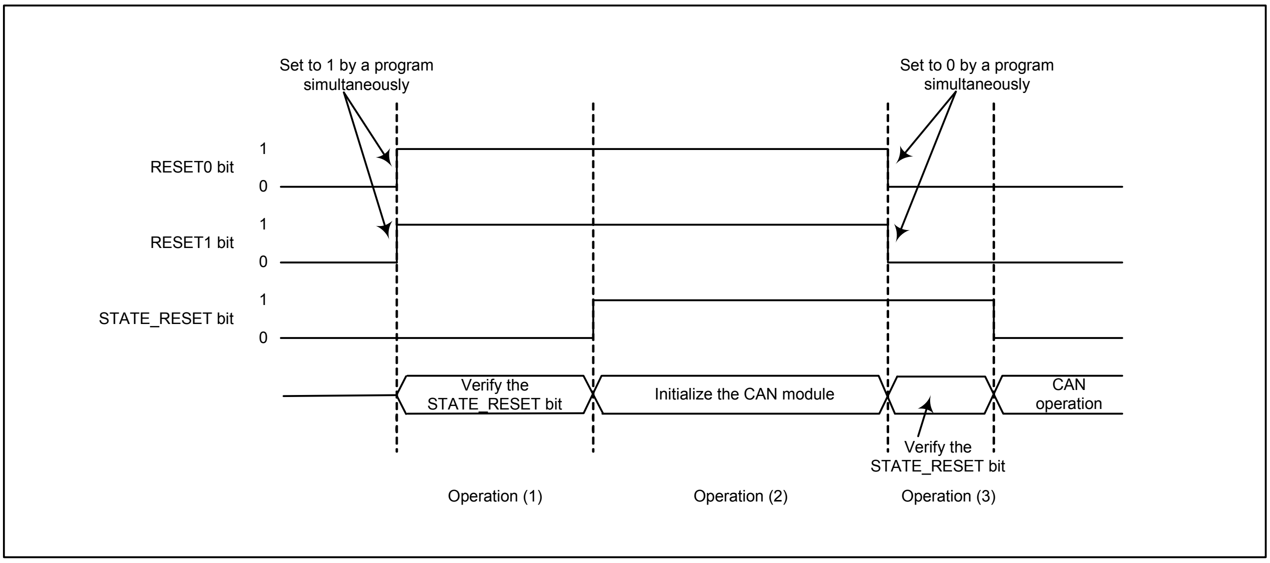


Figure 23.37 Example of CAN Module Initialize Operation

23.3.2 CAN Transmit Timing

Figure 23.38 shows an operation example when the CAN transmits a data frame or remote frame.

(1) When the TRMREQ bit in the CiMCTLj register (i = 0, 1; j = 0 to 15) is set to 1 (transmit operation requested) while the CAN bus is in an idle state, the TRMACTIVE bit in the CiMCTLj register becomes 1 (transmitting), the TRMSTATE bit in the CiSTR register becomes 1 (transmitting), and a CAN transmit operation is started.

(2) After a CAN transmit operation is completed, the SENTDATA bit in the CiMCTLj register becomes 1 (transmit operation completed), the TRMSUCC bit in the CiSTR register becomes 1 (transmit operation completed), and the SISj bit in the CiSISTR register becomes 1 (interrupt requested).

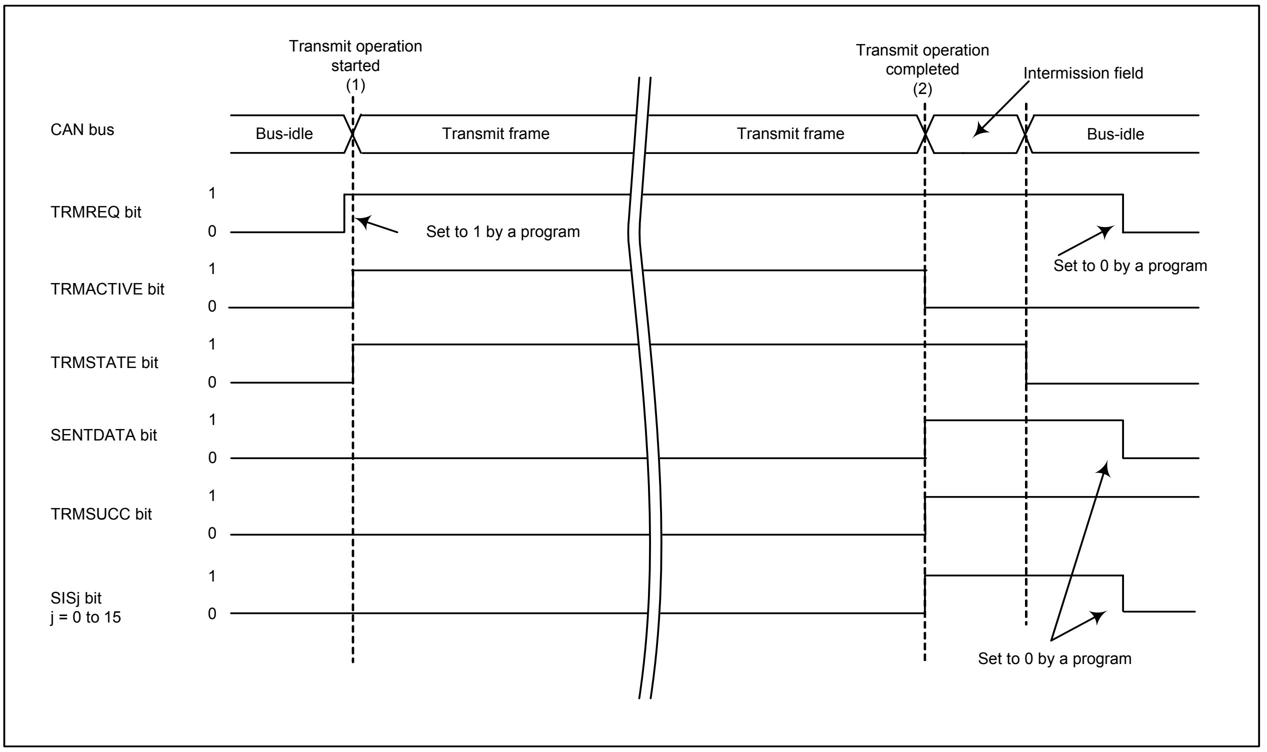


Figure 23.38 Example of CAN Data Frame Transmit Operation

23.3.3 CAN Receive Timing

Figure 23.39 shows an operation example when the CAN receives a data frame or remote frame.

(1) When the RECREQ bit in the CiMCTLj register (i = 0, 1; j = 0 to 15) is set to 1 (receive operation requested), the CAN module is ready to receive a data frame or remote frame.

(2) When a CAN receive operation is started, the RECSTATE bit in the CiSTR register becomes 1 (receiving).

(3) After the CAN receive operation is completed, the RECSUCC bit in the CiSTR register becomes 1 (receive operation completed). And then, the NEWDATA bit in the CiMCTLj register becomes 1 (receive operation completed) and the INVALDATA bit in the CiMCTLj register becomes 1 (storing receive data).

(4) After data is stored into the message slot, the INVALDATA bit becomes 0 (not storing receive data) and the SISj bit in the CiSISTR register becomes 1 (interrupt requested).

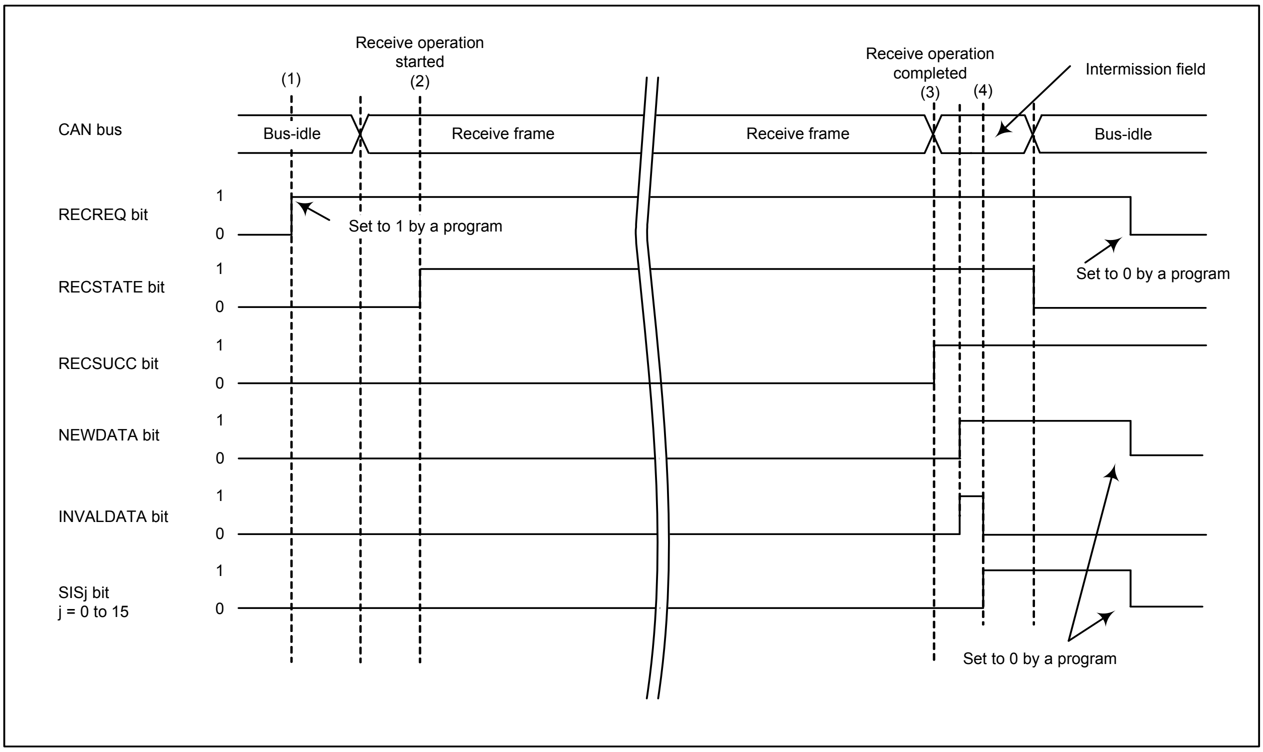


Figure 23.39 Example of CAN Data Frame Receive Operation

23.3.4 CAN Bus Error Timing

Figure 23.40 shows an operation example when a CAN bus error occurs.

(1) When a CAN bus error is detected, the STATE_BUSERROR bit in the CiSTR register becomes 1 (error occurred), the BEIS bit in the CiEISTR register becomes 1 (interrupt requested), and the CAN module transmits an error frame.

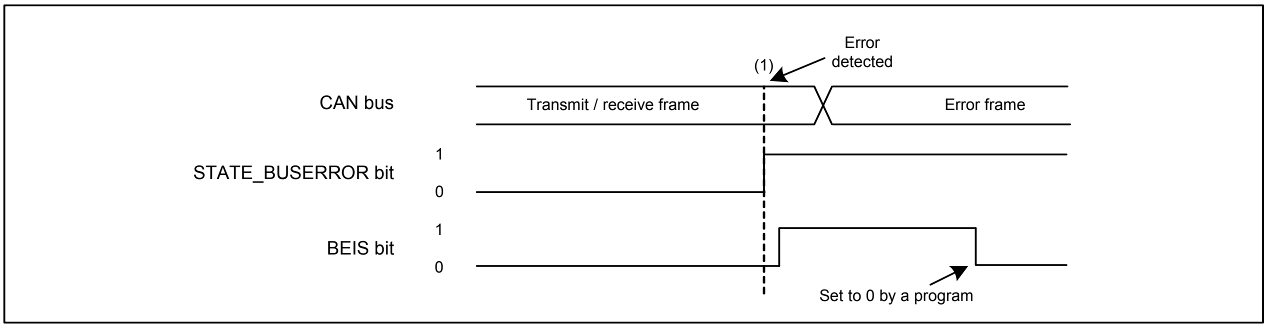


Figure 23.40 Operation Example when CAN Bus Error Occurs

23.4 CAN Interrupts

The CAN1 wake-up interrupt and CANij interrupt (i = 0, 1; j = 0 to 2) are provided as the CAN interrupts. The CAN1 wake-up interrupt, CANij interrupt are shared with the intelligent I/O interrupts. Refer to **11. Interrupts** for details on the interrupt. Figure 23.41 shows a block diagram of the CAN1 wake-up interrupt and CANij interrupt.

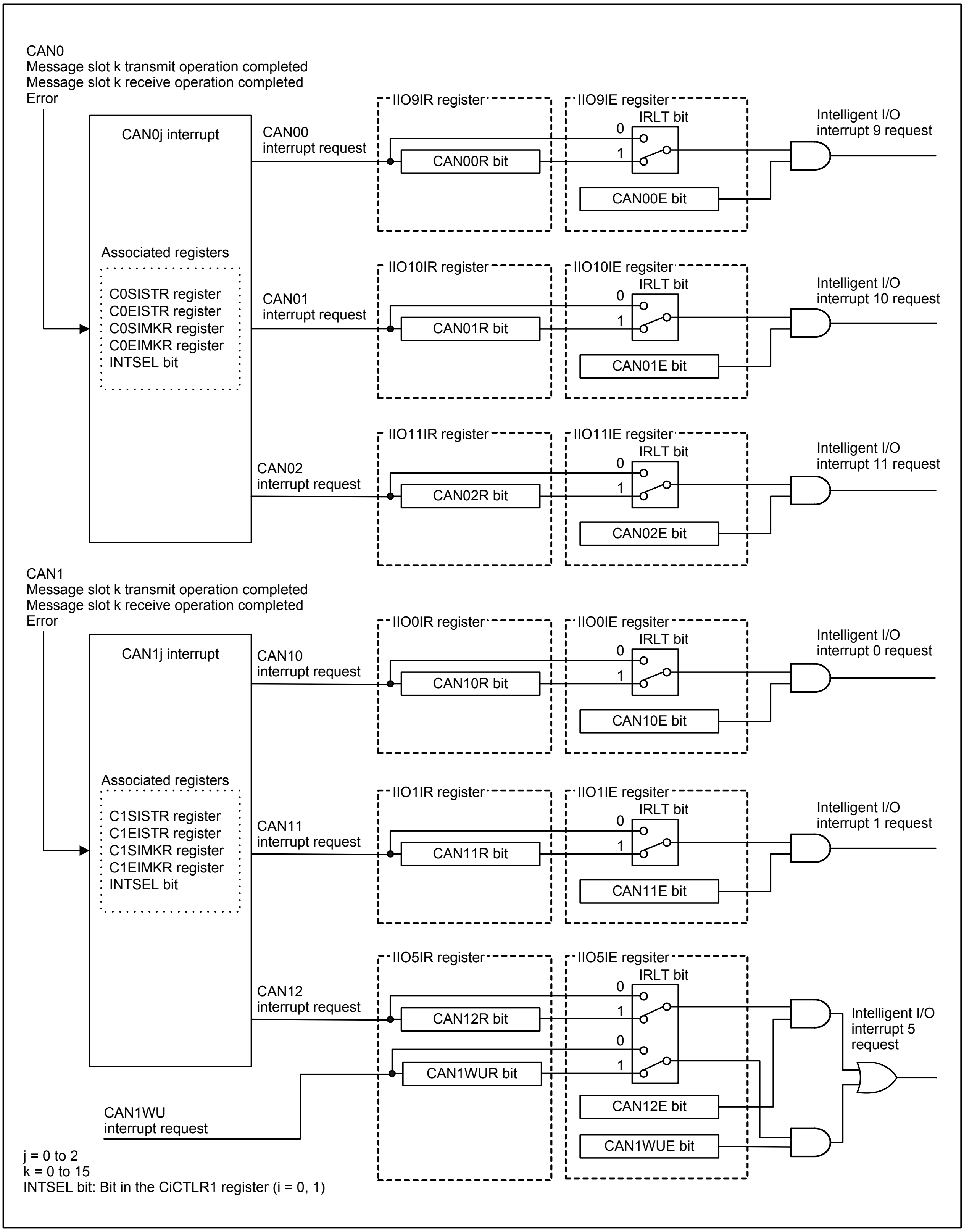


Figure 23.41 CAN1 Wake-UP Interrupt and CANij Interrupt Block Diagram

23.4.1 CAN1 Wake-Up Interrupt

When a signal applied to the $\overline{\text{CAN1WU}}$ pin is at the falling edge, the CAN1WUR bit in the IIO5IR register becomes 1 (interrupt requested), regardless of the value of the SLEEP bit in the C1SLPR register.
When P7_7 (CAN0IN) is used as a CAN0 input port, the CAN0 wake-up interrupt becomes available by using event counter mode of TA3IN that shares a pin with CAN0.
When P8_3 (CAN0IN/CAN1IN) is used as a CAN input port, the CAN0 and CAN1 wake-up interrupts become available by using $\overline{\text{INT1}}$ that shares a pin with CAN0IN/CAN1IN.

23.4.2 CANij Interrupt

The followings are the CANij interrupt request sources. (i = 0, 1; j = 0 to 2)

- CANi message slot k (k = 0 to 15) transmit operation completed
- CANi message slot k receive operation completed
- CANi bus error detected
- CANi error-passive state entered
- CANi bus-off state entered

When the INTSEL bit in the CiCTLR1 register is set to 0, the result of logical sum of interrupt requests from the above five sources becomes the CANij interrupt request.
When the INTSEL bit is set to 1, the interrupt requests from three types of CANij interrupt request sources, which are CANi message slot k transmit operation completed, CANi message slot k receive operation completed, and CANi error (bus error detected, error-passive state entered, and bus-off state entered), are individually output.

23.4.2.1 When the INTSEL Bit is Set to 0 (output CAN interrupt request via OR gate)

When the INTSEL bit is set to 0 (output CAN interrupt request via OR gate), all the CANi0, CANi1, and CANi2 interrupt requests are generated by any of the CANij interrupt source.
Table 23.5 lists interrupt sources and the corresponding interrupt registers (when INTSEL bit is set to 0). Figure 23.42 shows a CANij interrupt block diagram (when INTSEL bit is set to 0).

When a CANij interrupt request is generated, the interrupt status bit (the corresponding bit in the CiSISTR register or CiEISTR register) becomes 1 (interrupt requested). And then, if the interrupt mask bit (the corresponding bit in the CiSIMKR register or CiEIMKR register) is set to 1 (interrupt request enabled), all the corresponding CANijR bits in the IIOnIR register (n = 9, 10, 11 when i = 0; n = 0, 1, 5 when i = 1) become 1 (interrupt requested).

NOTE:

1. The interrupt status bits in registers CiSISTR and CiEISTR are not cleared to 0 automatically when an interrupt request is acknowledged. Set each bit to 0 by a program.
   While any of enabled status bits remains 1, the CANijR bit does not become 1 (interrupt requested) when a CANij interrupt request is generated.

Table 23.5 Interrupt Sources and Interrupt Registers (When INTSEL Bit is Set to 0)

CANij interrupt source	CANij Interrupt		Intelligent I/O interrupt
	Interrupt status bit 0: interrupt not requested 1: interrupt requested	Interrupt mask bit 0: interrupt request disabled 1: interrupt request enabled	Intelligent I/O interrupt request 0: interrupt not requested 1: interrupt requested
CANi message slot k receive operation completed	SISk bit in the CiSISTR register	SIMk bit in the CiSIMKR register	When i = 0, CAN0jR bit in registers IIO9IR, IIO10IR, and IIO11IR When i = 1, CAN1jR bit in registers IIO0IR, IIO1IR, and IIO5IR
CANi message slot k transmit operation completed			
CANi bus error detected	BEIS bit in the CiEISTR register	BEIM bit in the CiEIMKR register	
CANi error-passive state entered	EPIS bit in the CiEISTR register	EPIM bit in the CiEIMKR register	
CANi bus-off state entered	BOIS bit in the CiEISTR register	BOIM bit in the CiEIMKR register	

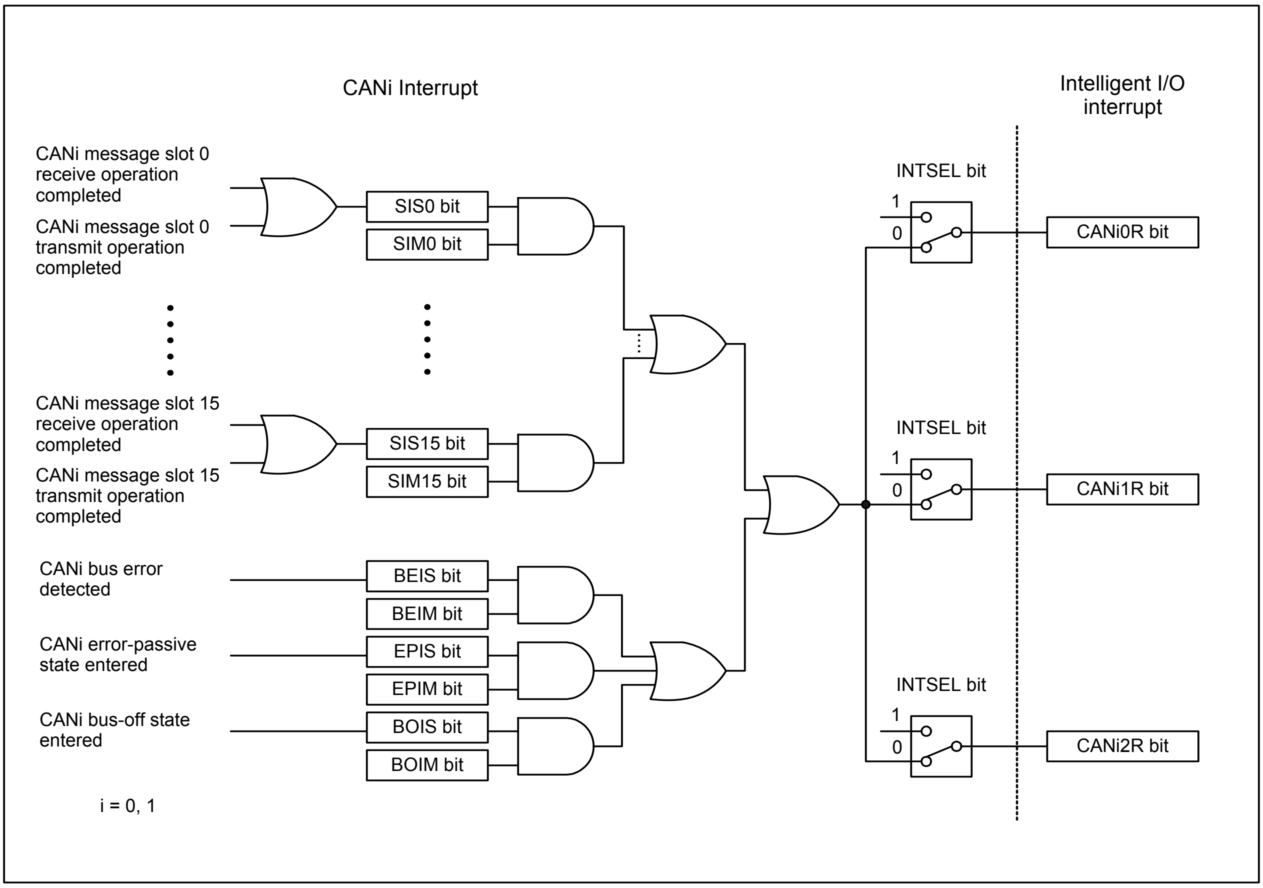


Figure 23.42 CANij Interrupt Block Diagram (When INTSEL Bit is Set to 0)

23.4.2.2 When the INTSEL Bit is Set to 1 (output CAN interrupt request individually)

When the INTSEL bit is set to 1 (output CAN interrupt request individually), the following three types of CANij interrupt sources output an interrupt request individually.

- When CANi message slot k transmit operation is completed, CANi0 interrupt request is generated.
- When CANi message slot k receive operation is completed, CANi1 interrupt request is generated.
- When CANi error (bus error detected, error-passive state entered, and bus-off state entered) occurs, CANi2 interrupt request is generated.

Table 23.6 lists interrupt sources and the corresponding interrupt registers (when INTSEL bit is set to 1). Figure 23.43 shows a CANij interrupt block diagram (when INTSEL bit is set to 1).
When a CANij interrupt request is generated, the interrupt status bit (the corresponding bit in the CiSISTR register or CiEISTR register) becomes 1 (interrupt requested). And then, if the interrupt mask bit (the corresponding bit in the CiSIMKR register or CiEIMKR register) is set to 1 (interrupt request enabled), the corresponding intelligent I/O interrupt request bit becomes 1 (interrupt requested).

NOTES:

1. The SISk bits in the CiSISTR register are not cleared to 0 automatically when an interrupt request is acknowledged. Set each bit to 0 by program. If the SISk bit remains 1, the CANi0R or CANi1R bit in the IIOnIR register (n = 9, 10 when i = 0, n = 0, 1 when i = 1) still becomes 1 (interrupt requested) when a CANi transmit/receive interrupt request is generated.
2. The bits in the CiEISTR register are not cleared to 0 automatically when an interrupt request is acknowledged. Set each bit to 0 by program. While any of enabled status bits remains 1, the CANi2R bit does not become 1 (interrupt requested) when a CANi error (bus error detected, error-passive state entered, and bus-off state entered) interrupt request is generated.

Table 23.6 Interrupt Sources and Interrupt Registers (When INTSEL Bit is Set to 1)

CANij interrupt source	CANij Interrupt		Intelligent I/O interrupt
	Interrupt status bit 0: interrupt not requested 1: interrupt requested	Interrupt mask bit 0: interrupt request disabled 1: interrupt request enabled	Intelligent I/O interrupt request 0: interrupt not requested 1: interrupt requested
CANi message slot k receive operation completed	SISk bit in the CiSISTR register	SIMk bit in the CiSIMKR register	When i = 0, CAN00R bit in the IIO9IR register When i = 1, CAN10R bit in the IIO0IR register
CANi message slot k transmit operation completed			When i = 0, CAN01R bit in the IIO10IR register When i = 1, CAN11R bit in the IIO1IR register
CANi bus error detected	BEIS bit in the CiEISTR register	BEIM bit in the CiEIMKR register	When i = 0, CAN02R bit in the IIO11IR register When i = 1, CAN12R bit in the IIO5IR register
CANi error-passive state entered	EPIS bit in the CiEISTR register	EPIM bit in the CiEIMKR register	
CANi bus-off state entered	BOIS bit in the CiEISTR register	BOIM bit in the CiEIMKR register	

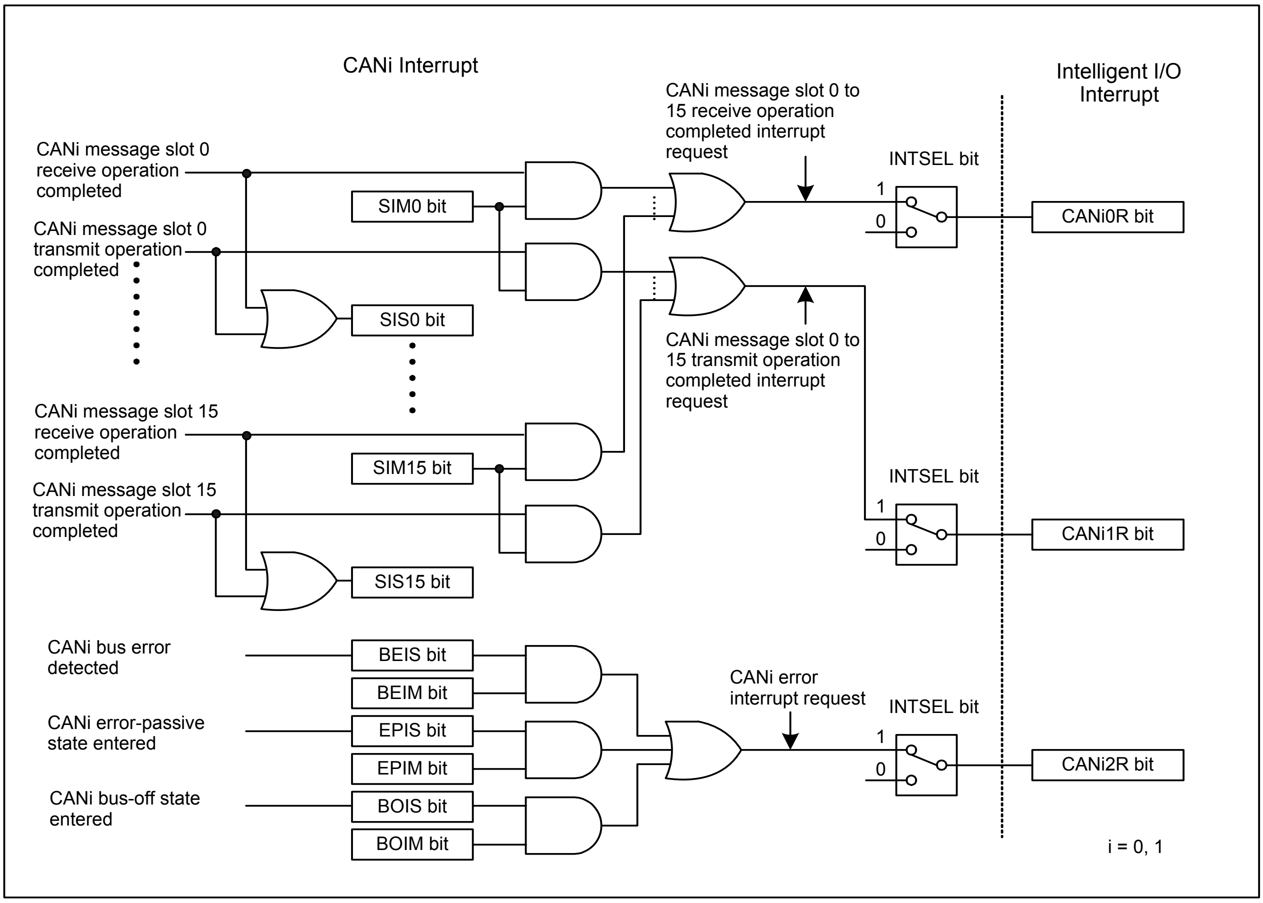


Figure 23.43 CANij Interrupt Block Diagram (When INTSEL Bit is Set to 1)

24. Real-Time Port (RTP)

When RTP output is selected, the values in the RTPiR register (i = 0 to 3) are output from pins RTPi_0 to RTPi_3 every time corresponding timer A or timer B underflows. Output values from pins RTPi_0 to RTPi_3 are undefined until the first timer A or timer B underflow. If the undefined RTP output becomes a problem, set pins as I/O ports in the Function Select Register A until the first timer A or timer B underflow. After the first timer A or timer B underflow, set the pins as RTP outputs in the Function Select Register A to E settings. Set timer A or timer B corresponding to RTP output to timer mode.

Figure 24.1 shows block diagram of RTP function. Figure 24.2 shows RTP-associated registers. Figure 24.3 shows RTP output timing. Table 24.1 lists pin settings.

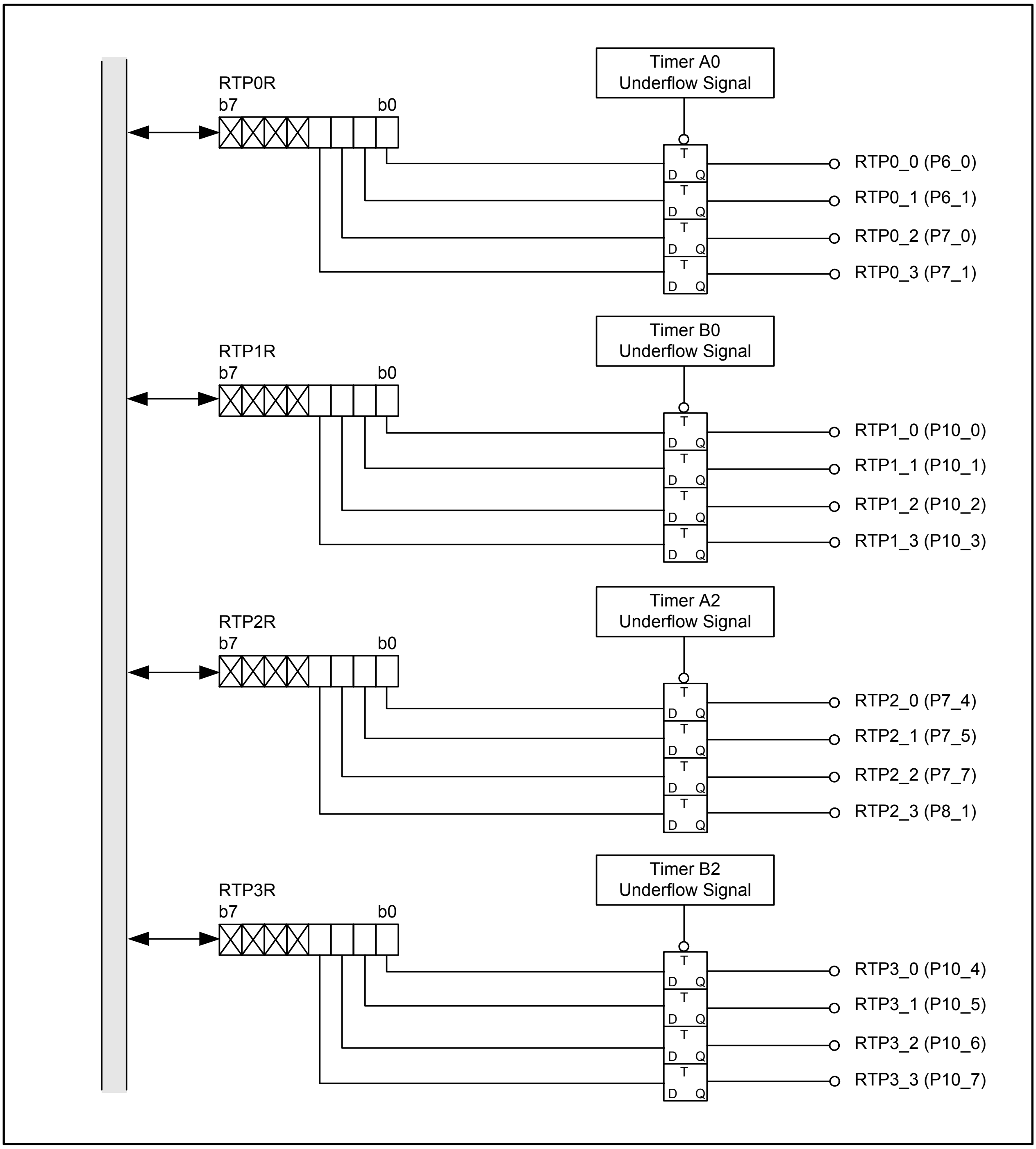


Figure 24.1 RTP Block Diagram

RTP Output Buffer Register i (i = 0 to 3)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
RTP0R to RTP3R	01D8h to 01DBh	XXh

Bit Symbol	Bit Name	Function	RW
RTPi0	RTPi_0 Output Buffer	0 : “L” output 1 : “H” output	WO
RTPi1	RTPi_1 Output Buffer	0 : “L” output 1 : “H” output	WO
RTPi2	RTPi_2 Output Buffer	0 : “L” output 1 : “H” output	WO
RTPi3	RTPi_3 Output Buffer	0 : “L” output 1 : “H” output	WO
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

Figure 24.2 RTP0R to RTP3R Registers

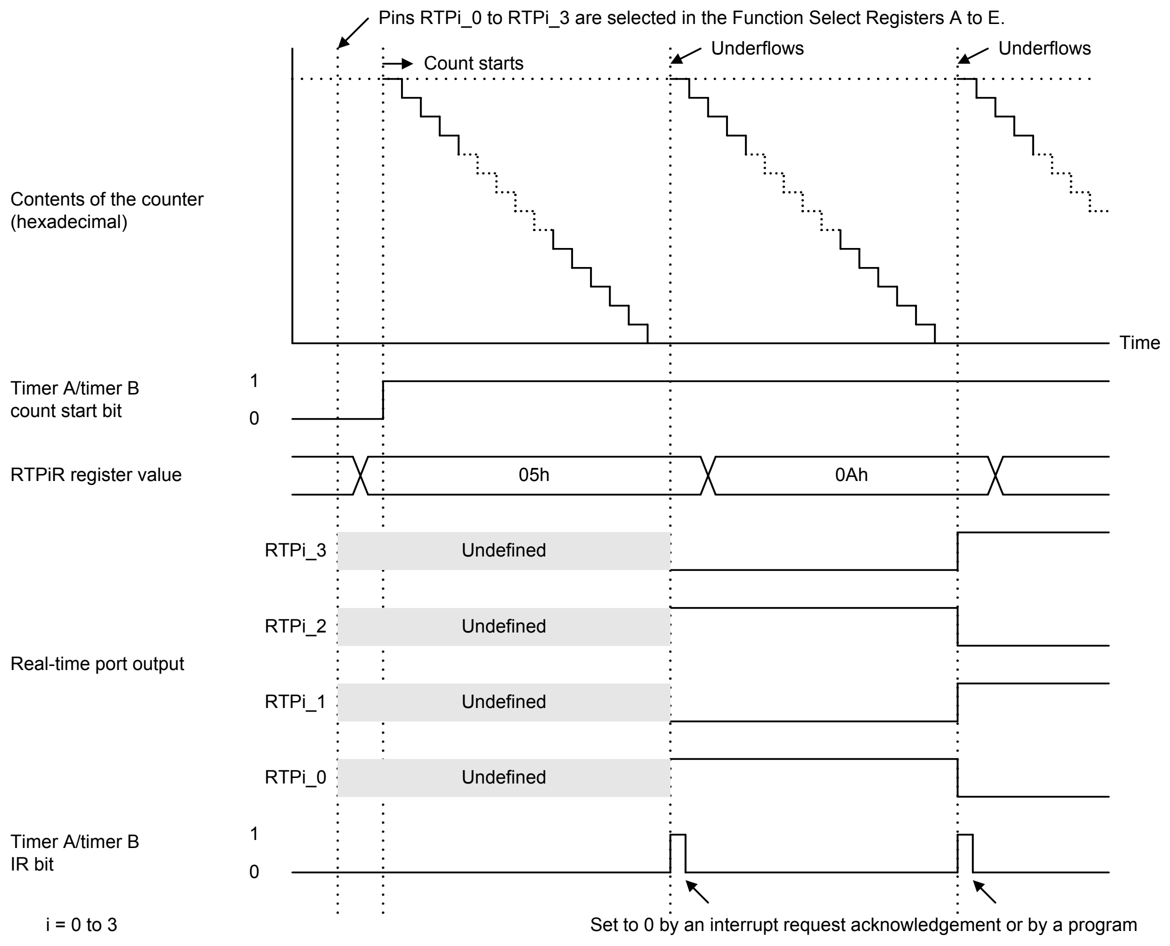


Figure 24.3 RTP Output Timing

Table 24.1 Pin Settings for Real Time Port

Port	Function	Bit Setting				
		PSE1, PSE2 Registers	PSD1, PSD2 Registers	PSC, PSC2 Registers	PSL0, PSL1, PSL2 Registers	PS0, PS1,PS2,PS4 Registers(1)
P6_0	RTP0_0	–	–	–	PSL0_0=1	PS0_0=1
P6_1	RTP0_1	–	–	–	PSL0_1=1	PS0_1=1
P7_0(2)	RTP0_2	PSE1_0=1	PSD1_0=1	PSC_0=1	PSL1_0=0	PS1_0=1
P7_1(2)	RTP0_3	PSE1_1=1	PSD1_1=1	PSC_1=1	PSL1_1=0	PS1_1=1
P7_4	RTP2_0	–	PSD1_4=1	PSC_4=1	PSL1_4=0	PS1_4=1
P7_5	RTP2_1	–	–	PSC_5=1	PSL1_5=1	PS1_5=1
P7_7	RTP2_2	PSE1_7=1	PSD1_7=1	–	PSL1_7=1	PS1_7=1
P8_1	RTP2_3	PSE2_1=1	PSD2_1=1	PSC2_1=1	PSL2_1=1	PS2_1=1
P10_0	RTP1_0	–	–	–	–	PS4_0=1
P10_1	RTP1_1	–	–	–	–	PS4_1=1
P10_2	RTP1_2	–	–	–	–	PS4_2=1
P10_3	RTP1_3	–	–	–	–	PS4_3=1
P10_4	RTP3_0	–	–	–	–	PS4_4=1
P10_5	RTP3_1	–	–	–	–	PS4_5=1
P10_6	RTP3_2	–	–	–	–	PS4_6=1
P10_7	RTP3_3	–	–	–	–	PS4_7=1

NOTES:
1. Set registers PS0, PS1, PS2, and PS4 after setting the other registers.
2. P7_0 and P7_1 are N-channel open drain output ports.

25. 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 $\overline{\mathrm{NMI}}$ input level since P8_5 shares its pin with $\overline{\mathrm{NMI}}$.
Figures 25.1 to 25.4 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.

25.1 Port Pi Direction Register (PDi Register, i = 0 to 15)

Figure 25.5 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, $\overline{\mathrm{A23}}$, D0 to D15, $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS3}}$, $\overline{\mathrm{WRL}}$ / $\overline{\mathrm{WR}}$, $\overline{\mathrm{WRH}}$ / $\overline{\mathrm{BHE}}$, $\overline{\mathrm{RD}}$, BCLK / ALE / CLKOUT, $\overline{\mathrm{HLDA}}$ / ALE, $\overline{\mathrm{HOLD}}$, ALE, and $\overline{\mathrm{RDY}}$. No bit controlling P8_5 is provided in the PDi register.

25.2 Port Pi Register (Pi Register, i = 0 to 15)

Figure 25.6 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, $\overline{\mathrm{A23}}$, D0 to D15, $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS3}}$, $\overline{\mathrm{WRL}}$/ $\overline{\mathrm{WR}}$, $\overline{\mathrm{WRH}}$ / $\overline{\mathrm{BHE}}$, $\overline{\mathrm{RD}}$, BCLK / ALE / CLKOUT, $\overline{\mathrm{HLDA}}$ / ALE, $\overline{\mathrm{HOLD}}$, ALE, and $\overline{\mathrm{RDY}}$.

25.3 Function Select Register A (PSj Register, j = 0 to 9)

Figures 25.7 to 25.11 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, PSL5 to PSL7, PSL9, PSC, PSC2, PSC3, PSC6, PSD1, PSD2, PSE1, and PSE2 to select which function to use.
Tables 25.3 to 25.13 list peripheral function output control settings for each pin.

25.4 Function Select Register B (PSLk Register, k = 0 to 3, 5 to 7, 9)

Figures 25.12 to 25.15 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 **25.11 Analog Input and Other Peripheral Function Input** for information on bits PSL3_3 to PSL3_6 in the PSL3 register.

25.5 Function Select Register C (PSC, PSC2, PSC3, and PSC6 Registers)

Figures 25.16 and 25.17 show registers PSC, PSC2, PSC3, and PSC6.
When multiple peripheral function outputs are assigned to a single pin, registers PSC, PSC2, PSC3, and PSC6 select which peripheral function output to use.
Refer to **25.11 Analog Input and Other Peripheral Function Input** for information on the PSC_7 bit in the PSC register.

25.6 Function Select Register D (PSD1 and PSD2 Registers)

Figure 25.18 shows registers PSD1 and PSD2.
When multiple peripheral function outputs are assigned to a single pin, registers PSD1 and PSD2 select which peripheral function output to use.

25.7 Function Select Register E (PSE1 and PSE2 Registers)

Figure 25.19 shows registers PSE1 and PSE2.
When multiple peripheral function outputs are assigned to a single pin, registers PSE1 and PSE2 select which peripheral function output to use.

25.8 Pull-up Control Register 0 to 4 (PUR0 to PUR4 Registers)

Figures 25.20 to 25.23 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.

25.9 Port Control Register (PCR Register)

Figure 25.24 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.

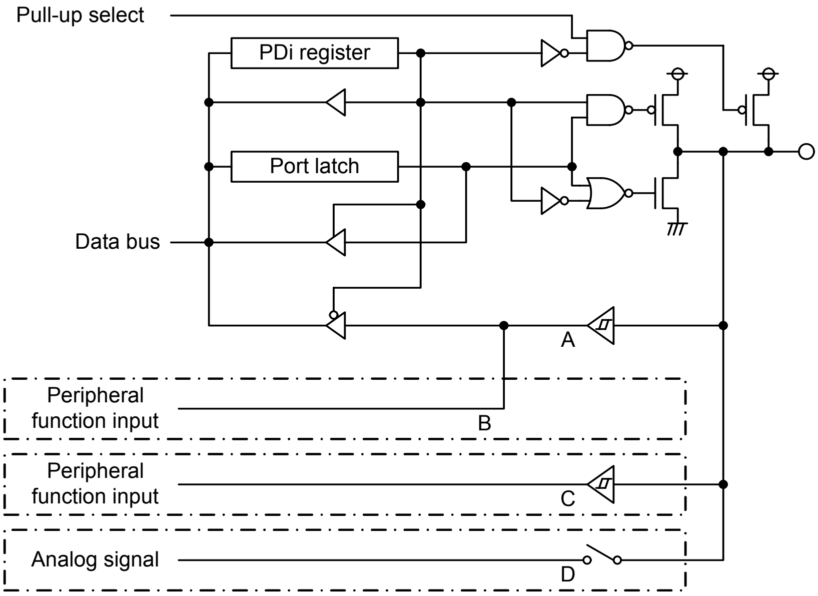
25.10 Input Function Select Register (IPS, IPSA, and IPSB Registers)

Figures 25.24 to 25.25 show registers IPS, IPSA, and IPSB.
Registers IPS and IPSA determine which pins are used as input pins for intelligent I/O or CAN.
Refer to **25.11 Analog Input and Other Peripheral Function Input** for information on the IPS2 bit in the IPS register and the IPSB register.

25.11 Analog Input and Other Peripheral Function Input

Bits PSL3_3 to PSL3_6 in the PSL3 register, the PSC_7 bit in the PSC register, the IPS2 bit in the IPS register, and bits IPSB_0 to IPSB_7 in the IPSB 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, AN_4 to AN_7 or AN15_0 to AN15_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.
When bits PSL3_3 to PSL3_6 (for P9_3 to P9_6), the IPS2 bit, and bits IPSB_0 to IPSB_7 (for P15_0 to P15_7) are set to 1, the input buffer for the peripheral functions except for the port function is disconnected.
For P10_4 to P10_7 (AN_4 to AN_7/$\overline{KI0}$ to $\overline{KI3}$), 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 $\overline{KI0}$ to $\overline{KI3}$ 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

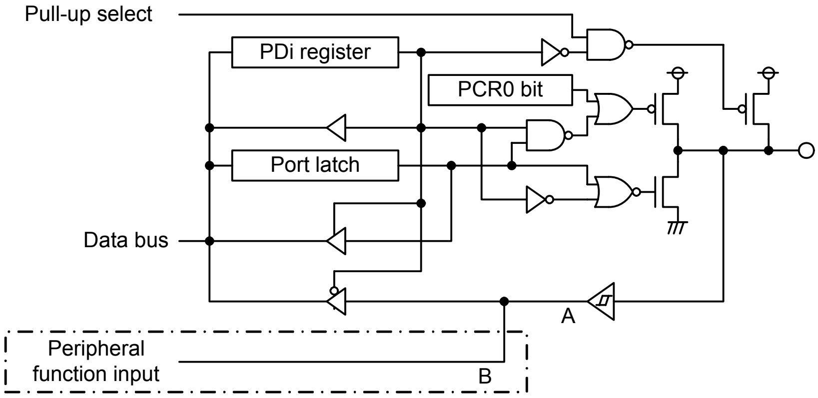


Port \ Option	(A) Hysteresis	(B) Peripheral function input	(C) Peripheral function input	(D) Analog I/F
P0_0 to P0_7 P2_0 to P2_7	–	–	–	○
P3_0 to P3_7 P4_0 to P4_7 P5_0 to P5_2	–	–	–	–
P5_5, P5_7	–	–	○	–
P8_3, P8_4	○	○	–	–
P8_6, P8_7	–	–	–	–

○ : Available –: Not available

Figure 25.1 Programmable I/O Ports (1/4)

Programmable I/O ports with the port control register

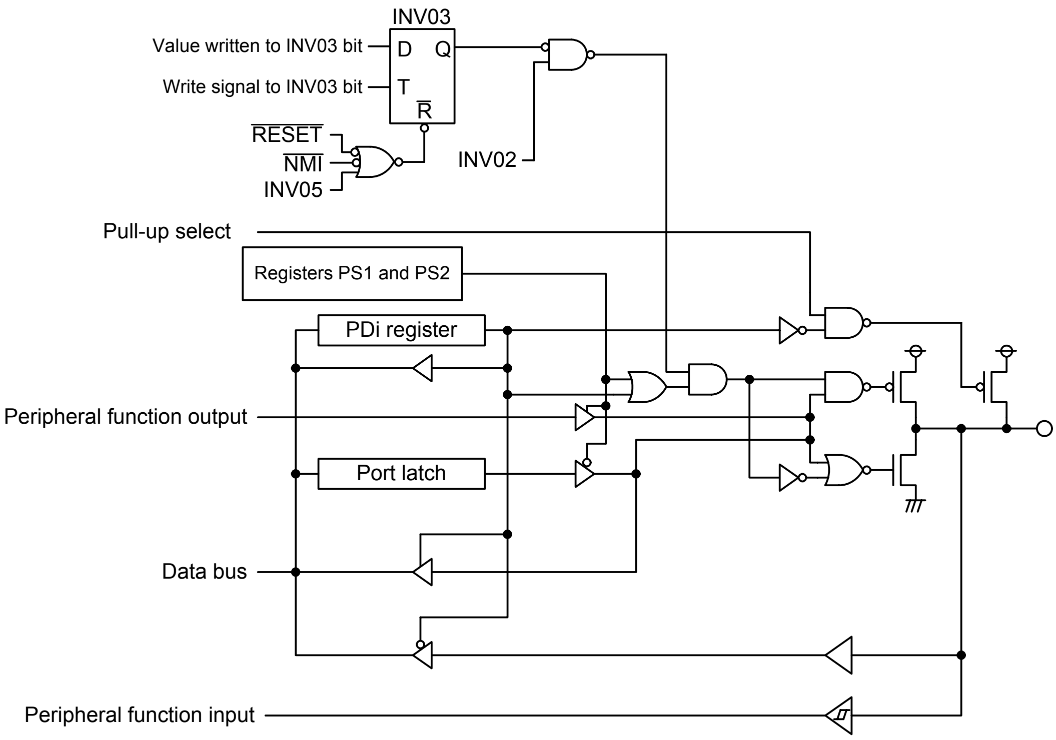


PCR0 bit: bit in the PCR register

Port \ Option	(A) Hysteresis	(B) Peripheral function input
P1_0 to P1_4	–	–
P1_5 to P1_7	○	○

○ : Available –: Not available

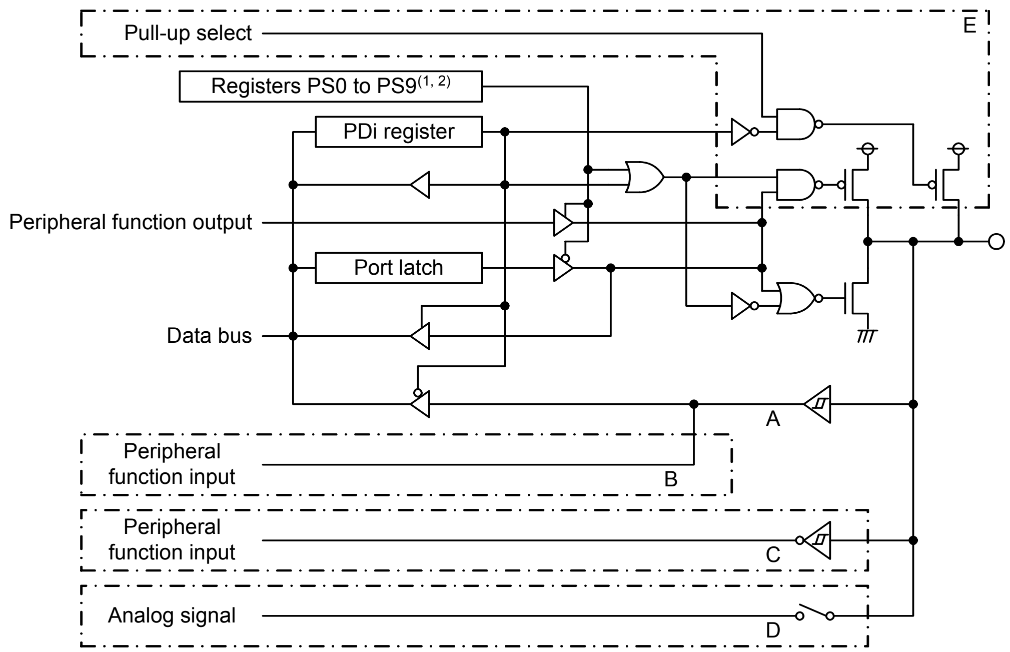
Programmable I/O ports with the function select register



Port P7_2 to P7_5, P8_0, P8_1

Figure 25.2 Programmable I/O Ports (2/4)

Programmable I/O ports with the function select register



	Port \ Option	(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	○	○	–	○	○
	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	–	–	○	–	○
(note 4)	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 25.3 Programmable I/O Ports (3/4)

Input-only port (P8_5)

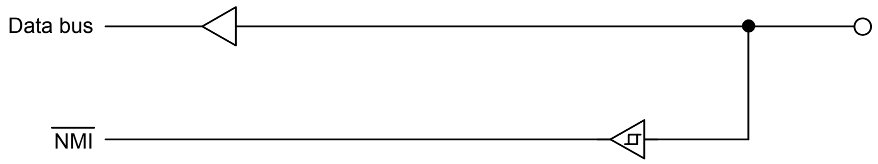


Figure 25.4 Programmable I/O Ports (4/4)

Port Pi Direction Register (i = 0 to 15)(2)

b7 b6 b5 b4 b3 b2 b1 b0

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, A23, D0 to D15, CS0 to CS3, WRL/ WR, WRH/BHE, RD, BCLK/ALE/CLKOUT, HLDA/ALE, HOLD, ALE, 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 25.5 PD0 to PD15 Registers

Port Pi Register (1, 2) (i = 0 to 15)

b7 b6 b5 b4 b3 b2 b1 b0

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, $\overline{A23}$, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WRL}$/ $\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, $\overline{RD}$, BCLK/ALE/CLKOUT, $\overline{HLDA}$/ALE, $\overline{HOLD}$, ALE, $\overline{RDY}$.
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 25.6 P0 to P15 Registers

Function Select Register A0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
PS0	03B0h	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

Symbol	Address	After Reset
PS1	03B1h	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: Select by the PSL1_7 bit	RW

Figure 25.7 PS0 Register, PS1 Register

Function Select Register A2

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

Symbol: PS2 Address: 03B4h After Reset: 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
PS2_2	Port P8_2 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL2_2 bit	RW
– (b4-b3)	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]

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: PS3 Address: 03B5h After Reset: 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: $\overline{RTS3}$	RW
PS3_4	Port P9_4 output function select bit	0: I/O port/peripheral function input 1: $\overline{RTS4}$	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: Select by the PSC3_6 bit	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 25.8 PS2 Register, PS3 Register

Function Select Register A4

b7 b6 b5 b4 b3 b2 b1 b0	Symbol	Address	After Reset
	PS4	03B8h	00h

Bit Symbol	Bit Name	Function	RW
PS4_0	Port P10_0 output function select bit	0: I/O port/peripheral function input 1: RTP1_0	RW
PS4_1	Port P10_1 output function select bit	0: I/O port/peripheral function input 1: RTP1_1	RW
PS4_2	Port P10_2 output function select bit	0: I/O port/peripheral function input 1: RTP1_2	RW
PS4_3	Port P10_3 output function select bit	0: I/O port/peripheral function input 1: RTP1_3	RW
PS4_4	Port P10_4 output function select bit	0: I/O port/peripheral function input 1: RTP3_0	RW
PS4_5	Port P10_5 output function select bit	0: I/O port/peripheral function input 1: RTP3_1	RW
PS4_6	Port P10_6 output function select bit	0: I/O port/peripheral function input 1: RTP3_2	RW
PS4_7	Port P10_7 output function select bit	0: I/O port/peripheral function input 1: RTP3_3	RW

Function Select Register A5(1)

b7 b6 b5 b4 b3 b2 b1 b0	Symbol	Address	After Reset
X X X 0	PS5	03B9h	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
PS5_0	Port P11_0 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL5_0 bit	RW
PS5_1	Port P11_1 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL5_1 bit	RW
PS5_2	Port P11_2 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL5_2 bit	RW
PS5_3	Port P11_3 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL5_3 bit	RW
– (b4)	Reserved bit	Set to 0	RW
– (b7-b5)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The PS5 register is provided in the 144-pin package only.

Figure 25.9 PS4 Register, PS5 Register

Function Select Register A6[1]

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

Symbol: PS6
Address: 03BCh
After Reset: 00h

Bit Symbol	Bit Name	Function	RW
PS6_0	Port P12_0 output function select bit	0: I/O port 1: Select by the PSL6_0 bit	RW
PS6_1	Port P12_1 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL6_1 bit	RW
– (b2)	Reserved bit	Set to 0	RW
PS6_3	Port P12_3 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL6_3 bit	RW
– (b7-b4)	Reserved bits	Set to 0	RW

NOTE:
1. The PS6 register is provided in the 144-pin package only.

Function Select Register A7[1]

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: PS7
Address: 03BDh
After Reset: 00h

Bit Symbol	Bit Name	Function	RW
PS7_0	Port P13_0 output function select bit	0: I/O port 1: Select by the PSL7_0 bit	RW
PS7_1	Port P13_1 output function select bit	0: I/O port 1: Select by the PSL7_1 bit	RW
PS7_2	Port P13_2 output function select bit	0: I/O port 1: Select by the PSL7_2 bit	RW
PS7_3	Port P13_3 output function select bit	0: I/O port 1: Select by the PSL7_3 bit	RW
PS7_4	Port P13_4 output function select bit	0: I/O port 1: Select by the PSL7_4 bit	RW
PS7_5	Port P13_5 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL7_5 bit	RW
PS7_6	Port P13_6 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL7_6 bit	RW
PS7_7	Port P13_7 output function select bit	0: I/O port 1: Select by the PSL7_7 bit	RW

NOTE:
1. The PS7 register is provided in the 144-pin package only.

Figure 25.10 PS6 Register, PS7 Register

Function Select Register A8[1]

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

Symbol	Address	After Reset
PS8	03A0h	X000 0000b

Bit Symbol	Bit Name	Function	RW
PS8_0	Port P14_0 output function select bit	0: I/O port/peripheral function input 1: OUTC1_4	RW
PS8_1	Port P14_1 output function select bit	0: I/O port/peripheral function input 1: OUTC1_5	RW
PS8_2	Port P14_2 output function select bit	0: I/O port/peripheral function input 1: OUTC1_6	RW
PS8_3	Port P14_3 output function select bit	0: I/O port/peripheral function input 1: OUTC1_7	RW
– (b6-b4)	Reserved bits	Set to 0	RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The PS8 register is provided in the 144-pin package only.

Function Select Register A9[1]

b7	b6	b5	b4	b3	b2	b1	b0
		0			0		

Symbol	Address	After Reset
PS9	03A1h	00h

Bit Symbol	Bit Name	Function	RW
PS9_0	Port P15_0 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL9_0 bit	RW
PS9_1	Port P15_1 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL9_1 bit	RW
– (b2)	Reserved bit	Set to 0	RW
PS9_3	Port P15_3 output function select bit	0: I/O port/peripheral function input 1: $\overline{RTS5}$	RW
PS9_4	Port P15_4 output function select bit	0: I/O port/peripheral function input 1: Select by the PSL9_4 bit	RW
– (b5)	Reserved bit	Set to 0	RW
PS9_6	Port P15_6 output function select bit	0: I/O port/peripheral function input 1: CLK6 output	RW
PS9_7	Port P15_7 output function select bit	0: I/O port/peripheral function input 1: $\overline{RTS6}$	RW

NOTE:
1. The PS9 register is provided in the 144-pin package only.

Figure 25.11 PS8 Register, PS9 Register

Function Select Register B0

b7	b6	b5	b4	b3	b2	b1	b0
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: RTP0_0	RW
PSL0_1	Port P6_1 peripheral function output select bit	0: CLK0 output 1: RTP0_1	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/IrDAOUT 1: Do not set to this value	RW
PSL0_4	Port P6_4 peripheral function output select bit	0: $\overline{RTS1}$ 1: OUTC2_1/ISCLK2 output	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

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: Select by the PSC_5 bit	RW
PSL1_6	Port P7_6 peripheral function output select bit	0: Select by the PSC_6 bit 1: TA3OUT output	RW
PSL1_7	Port P7_7 peripheral function output select bit	0: ISCLK0 output 1: Select by the PSD1_7 bit	RW

Figure 25.12 PSL0 Register, PSL1 Register

Function Select Register B2

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

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: Select by the PSC2_1 bit	RW
PSL2_2	Port P8_2 peripheral function output select bit	0: Do not set to this value 1: Select by the PSC2_2 bit	RW
– (b4-b3)	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

b7	b6	b5	b4	b3	b2	b1	b0
							0

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: OUTC2_0/ISTXD2/IEOUT	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 25.13 PSL2 Register, PSL3 Register

Function Select Register B5[1]

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

Symbol: PSL5 Address: 03BBh After Reset: XXX0 0000b

Bit Symbol	Bit Name	Function	RW
PSL5_0	Port P11_0 peripheral function output select bit	0: OUTC1_0/ISTXD1 1: Do not set to this value	RW
PSL5_1	Port P11_1 peripheral function output select bit	0: OUTC1_1/ISCLK1 output 1: Do not set to this value	RW
PSL5_2	Port P11_2 peripheral function output select bit	0: OUTC1_2 1: Do not set to this value	RW
PSL5_3	Port P11_3 peripheral function output select bit	0: OUTC1_3 1: Do not set to this value	RW
– (b4)	Reserved bit	Set to 0	RW
– (b7-b5)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The PSL5 register is provided in the 144-pin package only.

Function Select Register B6[1]

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

Symbol: PSL6 Address: 03BEh After Reset: 00h

Bit Symbol	Bit Name	Function	RW
PSL6_0	Port P12_0 peripheral function output select bit	0: Select by the PSC6_0 bit 1: Do not set to this value	RW
PSL6_1	Port P12_1 peripheral function output select bit	0: Select by the PSC6_1 bit 1: Do not set to this value	RW
– (b2)	Reserved bit	Set to 0	RW
PSL6_3	Port P12_3 peripheral function output select bit	0: Select by the PSC6_3 bit 1: Do not set to this value	RW
– (b7-b4)	Reserved bits	Set to 0	RW

NOTE:
1. The PSL6 register is provided in the 144-pin package only.

Figure 25.14 PSL5 Register, PSL6 Register

Function Select Register B7[1]

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

Symbol	Address	After Reset
PSL7	03BFh	00h

Bit Symbol	Bit Name	Function	RW
PSL7_0	Port P13_0 peripheral function output select bit	0: OUTC2_4 1: Do not set to this value	RW
PSL7_1	Port P13_1 peripheral function output select bit	0: OUTC2_5 1: Do not set to this value	RW
PSL7_2	Port P13_2 peripheral function output select bit	0: OUTC2_6 1: Do not set to this value	RW
PSL7_3	Port P13_3 peripheral function output select bit	0: OUTC2_3 1: Do not set to this value	RW
PSL7_4	Port P13_4 peripheral function output select bit	0: OUTC2_0/ISTXD2/IEOUT 1: Do not set to this value	RW
PSL7_5	Port P13_5 peripheral function output select bit	0: OUTC2_2 1: Do not set to this value	RW
PSL7_6	Port P13_6 peripheral function output select bit	0: OUTC2_1/ISCLK2 output 1: Do not set to this value	RW
PSL7_7	Port P13_7 peripheral function output select bit	0: OUTC2_7 1: Do not set to this value	RW

NOTE:
1. The PSL7 register is provided in the 144-pin package only.

Function Select Register B9[1]

b7 b6 b5 b4 b3 b2 b1 b0: X X X 1 X X _ _

Symbol	Address	After Reset
PSL9	03A3h	XXX0 XX00b

Bit Symbol	Bit Name	Function	RW
PSL9_0	Port P15_0 peripheral function output select bit	0: ISTXD0 1: TXD5	RW
PSL9_1	Port P15_1 peripheral function output select bit	0: ISCLK0 output 1: CLK5 output	RW
– (b3-b2)	Unimplemented. Write 0. Read as undefined value.		–
PSL9_4	Port P15_4 peripheral function output select bit	0: Do not set to this value 1: TXD6	RW
– (b7-b5)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The PSL9 register is provided in the 144-pin package only.

Figure 25.15 PSL7 Register, PSL9 Register

Function Select Register C

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: PSC Address: 03AFh After Reset: 00h

Bit Symbol	Bit Name	Function	RW
PSC_0	Port P7_0 peripheral function output select bit	0: TXD2/SDA2 output 1: Select by the PSD1_0 bit	RW
PSC_1	Port P7_1 peripheral function output select bit	0: SCL2 output 1: Select by the PSD1_1 bit	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: OUTC1_0/ISTXD1	RW
PSC_4	Port P7_4 peripheral function output select bit	0: TA2OUT output 1: Select by the PSD1_4 bit	RW
PSC_5	Port P7_5 peripheral function output select bit	0: OUTC1_2 1: RTP2_1	RW
PSC_6	Port P7_6 peripheral function output select bit	0: Select by the PSD1_6 bit 1: CAN0OUT(1)	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(2)	RW

NOTES:
1. Set to 0 in M32C/87B.
2. 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.

Function Select Register C2

b7 b6 b5 b4 b3 b2 b1 b0 (b1: 1)

Symbol: PSC2 Address: 03ACh After Reset: XXXX X00Xb

Bit Symbol	Bit Name	Function	RW
– (b0)	Unimplemented. Write 0. Read as undefined value.		–
PSC2_1	Port P8_1 peripheral function output select bit	0: Do not set to this value 1: Select by the PSD2_1 bit	RW
PSC2_2	Port P8_2 peripheral function output select bit	0: CAN0OUT 1: CAN1OUT(1)	RW
– (b7-b3)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. Set to 0 in M32C/87A.
   Do not set the PSC2_2 bit in M32C/87B. Write a 0, if necessary.

Figure 25.16 PSC Register, PSC2 Register

Function Select Register C3

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

Symbol	Address	After Reset
PSC3	03ADh	X0XX XXXXb

Bit Symbol	Bit Name	Function	RW
– (b5-b0)	Unimplemented. Write 0. Read as undefined value.		–
PSC3_6	Port P9_6 peripheral function output select bit	0: TXD4/SDA4 output 1: CAN1OUT(1)	RW
– (b7)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. Set to 0 in M32C/87A and M32C/87B.

Function Select Register C6(1)

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

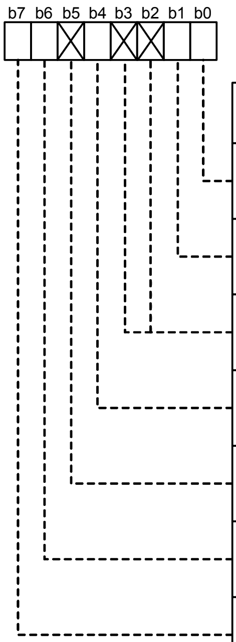
Symbol	Address	After Reset
PSC6	03AAh	XXXX 0X00b

Bit Symbol	Bit Name	Function	RW
PSC6_0	Port P12_0 peripheral function output select bit	0: Do not set to this value 1: TXD6	RW
PSC6_1	Port P12_1 peripheral function output select bit	0: Do not set to this value 1: CLK6 output	RW
– (b2)	Unimplemented. Write 0. Read as undefined value.		–
PSC6_3	Port P12_3 peripheral function output select bit	0: Do not set to this value 1: $\overline{RTS6}$	RW
– (b7-b4)	Unimplemented. Write 0. Read as undefined value.		–

NOTE:
1. The PSC6 register is provided in the 144-pin package only.

Figure 25.17 PSC3 Register, PSC6 Register

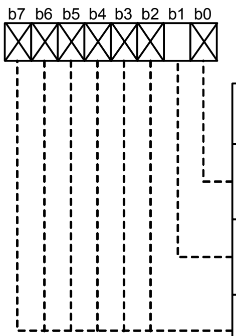
Function Select Register D1



Symbol: PSD1 | Address: 03A7h | After Reset: 00X0 XX00b

Bit Symbol	Bit Name	Function	RW
PSD1_0	Port P7_0 peripheral function output select bit	0: OUTC2_0/ISTXD2/IEOUT 1: Select by the PSE1_0 bit	RW
PSD1_1	Port P7_1 peripheral function output select bit	0: OUTC2_2 1: Select by the PSE1_1 bit	RW
– (b3-b2)	Unimplemented. Write 0. Read as undefined value.		–
PSD1_4	Port P7_4 peripheral function output select bit	0: OUTC1_1/ISCLK1 output 1: RTP2_0	RW
– (b5)	Unimplemented. Write 0. Read as undefined value.		–
PSD1_6	Port P7_6 peripheral function output select bit	0: ISTXD0 1: Select by the PSE1_6 bit	RW
PSD1_7	Port P7_7 peripheral function output select bit	0: OUTC1_4 1: Select by the PSE1_7 bit	RW

Function Select Register D2

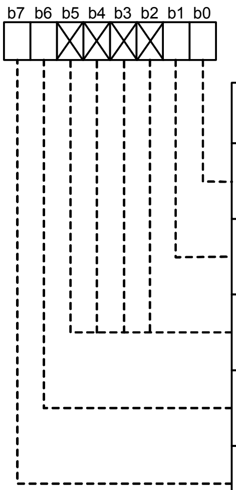


Symbol: PSD2 | Address: 03A8h | After Reset: XXXX XX0Xb

Bit Symbol	Bit Name	Function	RW
– (b0)	Unimplemented. Write 0. Read as undefined value.		–
PSD2_1	Port P8_1 peripheral function output select bit	0: OUTC1_5 1: Select by the PSE2_1 bit	RW
– (b7-b2)	Unimplemented. Write 0. Read as undefined value.		–

Figure 25.18 PSD1 Register, PSD2 Register

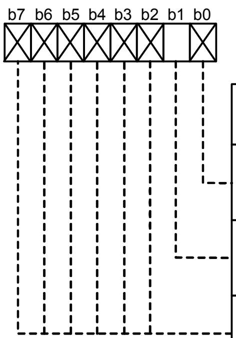
Function Select Register E1



Symbol	Address	After Reset
PSE1	03ABh	00XX XX00b

Bit Symbol	Bit Name	Function	RW
PSE1_0	Port P7_0 peripheral function output select bit	0: OUTC1_6 1: RTP0_2	RW
PSE1_1	Port P7_1 peripheral function output select bit	0: OUTC1_7 1: RTP0_3	RW
– (b5-b2)	Unimplemented. Write 0. Read as undefined value.		–
PSE1_6	Port P7_6 peripheral function output select bit	0: OUTC1_3 1: TXD5	RW
PSE1_7	Port P7_7 peripheral function output select bit	0: CLK5 output 1: RTP2_2	RW

Function Select Register E2



Symbol	Address	After Reset
PSE2	03A4h	XXXX XX0Xb

Bit Symbol	Bit Name	Function	RW
– (b0)	Unimplemented. Write 0. Read as undefined value.		–
PSE2_1	Port P8_1 peripheral function output select bit	0: $\overline{RTS5}$ 1: RTP2_3	RW
– (b7-b2)	Unimplemented. Write 0. Read as undefined value.		–

Figure 25.19 PSE1 Register, PSE2 Register

Pull-Up Control Register 0(1)

b7 b6 b5 b4 b3 b2 b1 b0

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)

b7 b6 b5 b4 b3 b2 b1 b0

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 25.20 PUR0 Register, PUR1 Register

Pull-Up Control Register 2

b7 b6 b5 b4 b3 b2 b1 b0

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 25.21 PUR2 Register

Pull-Up Control Register 3 <144-pin package>

b7 b6 b5 b4 b3 b2 b1 b0

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>

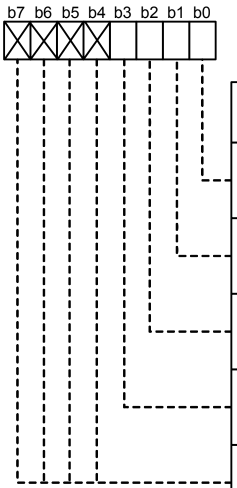
b7 b6 b5 b4 b3 b2 b1 b0
0 0 0 0 0 0

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 25.22 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 25.23 PUR4 Register

Port Control Register

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

Symbol	Address	After Reset
PCR	03FFh	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.

Input Function Select Register

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	After Reset
IPS	0178h	00h

Bit Symbol	Bit Name	Function	RW
IPS0	Group 0 input pin select bit 0	Assigns ISCLK0 input and ISRXD0 to the following ports 0: P7_7, P8_0 1: P15_1, P15_2	RW
IPS1	Group 1 input pin select bit 1	Assigns INPC1_0, INPC1_1/ISCLK1 input/ INPC1_2/ISRXD1, INPC1_3, INPC1_4, INPC1_5, INPC1_6, and INPC1_7 to the following ports. 0: P7_3, P7_4, P7_5, P7_6, P7_7, P8_1, P7_0, P7_1 1: P11_0, P11_1, P11_2, P11_3, P14_0, P14_1, P14_2, P14_3	RW
IPS2	Port P15 peripheral function input select bit[1]	0: Except AN15 1: AN15	RW
IPS3	CAN0IN function pin select bit	0: P7_7 1: P8_3	RW
IPS4	ISRXD2/IEIN function pin select bit	b5 b4 0 0: P7_1 0 1: P9_1 1 0: P13_5 1 1: Do not set to this value.	RW
IPS5			
IPS6	ISCLK2 function input pin select bit	0: P6_4 1: P13_6	RW
– (b7)	Reserved bit	Set to 0	RW

NOTE:
1. If AN15_0 to AN15_7 are used with the IPS2 bit setting to 0, current consumption may increase.

Figure 25.24 PCR Register, IPS Register

Input Function Select Register A

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

Symbol	Address	After Reset
IPSA	0179h	00h

Bit Symbol	Bit Name	Function	RW
IPSA_0	Intelligent I/O two-phase pulse input pin switch bit	0: P8_0, P8_1, $\overline{INT1}$ 1: P7_6, P7_7, $\overline{INT0}$	RW
– (b2-b1)	Reserved bits	Set to 0	RW
IPSA_3	CAN1IN function pin select bit	0: P9_5 1: P8_3[1]	RW
– (b7-b4)	Reserved bits	Set to 0	RW

NOTE:
1. Do not set the IPSA_3 bit in M32C/87A and M32C/87B. Write a 0, if necessary.

Input Function Select Register B

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
IPSB	0177h	00h

Bit Symbol	Bit Name	Function	RW
IPSB_0	Port P15_0 input function select bit	0: Except AN15_0[2] 1: AN15_0	RW
IPSB_1	Port P15_1 input function select bit	0: Except AN15_1[2] 1: AN15_1	RW
IPSB_2	Port P15_2 input function select bit	0: Except AN15_2[2] 1: AN15_2	RW
IPSB_3	Port P15_3 input function select bit	0: Except AN15_3[2] 1: AN15_3	RW
IPSB_4	Port P15_4 input function select bit	0: Except AN15_4[2] 1: AN15_4	RW
IPSB_5	Port P15_5 input function select bit	0: Except AN15_5[2] 1: AN15_5	RW
IPSB_6	Port P15_6 input function select bit	0: Except AN15_6[2] 1: AN15_6	RW
IPSB_7	Port P15_7 input function select bit	0: Except AN15_7[2] 1: AN15_7	RW

NOTES:
1. The IPSB register is enabled when the IPS2 bit in the IPS register is se to 0 (except AN15).
2. If the bits AN15_0 to AN15_7 are used with bits IPSB_0 to IPSB_7 setting to 0, current consumption may increase.

Figure 25.25 IPSA Register, IPSB Register

Table 25.1 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
$\overline{\text{NMI}}$ (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 25.2 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
$\overline{\text{BHE}}$, ALE, $\overline{\text{HLDA}}$, XOUT(2), BCLK	Leave the pin open
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$	Connect the pin to VCC2 via a resistor (pull-up)
$\overline{\text{NMI}}$(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.

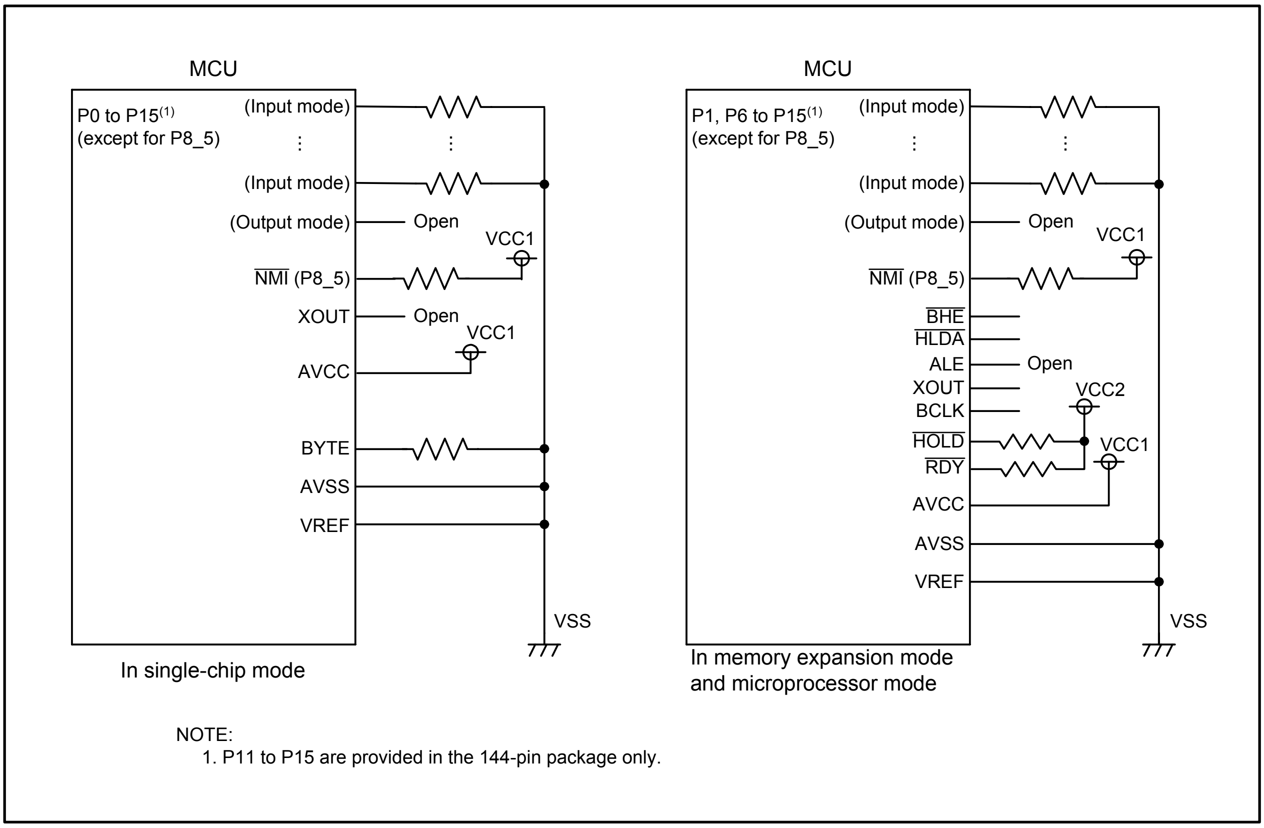


Figure 25.26 Unassigned Pin Handling

Table 25.3 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: RTP0_0
Bit 1	0: P6_1/CLK0 input 1: Select by the PSL0_1 bit	0: CLK0 output 1: RTP0_1
Bit 2	0: P6_2/RXD0/SCL0 input/IrDAIN 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/IrDAOUT 1: Do not set to this value
Bit 4	0: P6_4/$\overline{\text{CTS1}}$/$\overline{\text{SS1}}$/ISCLK2 input 1: Select by the PSL0_4 bit	0: $\overline{\text{RTS1}}$ 1: OUTC2_1/ISCLK2 output
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 25.4 Port P7 Peripheral Function Output Control

	PS1 Register	PSL1 Register	PSC Register	PSD1 Register	PSE1 Register
Bit 0	0: P7_0/ TA0OUT input/ SRXD2/INPC1_6/ SDA2 input 1: Select by the PSL1_0 bit	0: Select by the PSC_0 bit 1: TA0OUT output	0: TXD2/ SDA2 output 1: Select by the PSD1_0 bit	0: OUTC2_0/ ISTXD2/IEOUT 1: Select by the PSE1_0 bit	0: OUTC1_6 1: RTP0_2
Bit 1	0: P7_1/TA0IN/ TB5IN/RXD2/ SCL2 input/ INPC1_7/ ISRXD2/IEIN 1: Select by the PSL1_1 bit	0: Select by the PSC_1 bit 1: STXD2	0: SCL2 output 1: Select by the PSD1_1 bit	0: OUTC2_2 1: Select by the PSE1_1 bit	0: OUTC1_7 1: RTP0_3
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	Set to 0	Set to 0
Bit 3	0: P7_3/TA1IN/ $\overline{CTS2}$/$\overline{SS2}$/ INPC1_0 1: Select by the PSL1_3 bit	0: Select by the PSC_3 bit 1: $\overline{V}$	0: $\overline{RTS2}$ 1:OUTC1_0/ ISTXD1	Set to 0	Set to 0
Bit 4	0: P7_4/TA2OUT input/INPC1_1/ ISCLK1 input 1: Select by the PSL1_4 bit	0: Select by the PSC_4 bit 1: W	0: TA2OUT output 1: Select by the PSD1_4 bit	0: OUTC1_1 ISCLK1 output 1: RTP2_0	Set to 0
Bit 5	0: P7_5/TA2IN/ INPC1_2/ISRXD1 1: Select by the PSL1_5 bit	0: $\overline{W}$ 1: Select by the PSC_5 bit	0: OUTC1_2 1: RTP2_1	Set to 0	Set to 0
Bit 6	0: P7_6/TA3OUT input/INPC1_3 1: Select by the PSL1_6 bit	0: Select by the PSC_6 bit 1: TA3OUT output	0: Select by the PSD1_6 bit 1: CAN0OUT(1)	0: ISTXD0 1: Selected by the PSE1_6 bit	0: OUTC1_3 1: TXD5
Bit 7	0: P7_7/TA3IN/ CAN0IN/CLK5 input/INPC1_4/ ISCLK0 input 1: Select by the PSL1_7 bit	0: ISCLK0 output 1: Select by the PSD1_7 bit	–	0: OUTC1_4 1: Select by the PSE1_7 bit	0: CLK5 output 1: RTP2_2

NOTE:
1. Set to 0 in M32C/87B.

Table 25.5 Port P8 Peripheral Function Output Control

	PS2 Register	PSL2 Register	PSC2 Register	PSD2 Register	PSE2 Register
Bit 0	0: P8_0/TA4OUT input/RXD5/ ISRXD0 1: Select by the PSL2_0 bit	0: TA4OUT output 1: U	Set to 0	Set to 0	Set to 0
Bit 1	0: P8_1/TA4IN/ $\overline{CTS5}$/INPC1_5 1: Select by the PSL2_1 bit	0: $\overline{U}$ 1: Select by the PSC2_1 bit	0: Do not set to this value 1: Select by the PSD2_1 bit	0: OUTC1_5 1: Select by the PSE2_1 bit	0: $\overline{RTS5}$ 1: RTP2_3
Bit 2	0: P8_2/$\overline{INT0}$ 1: Select by the PSL2 _2 bit	0: Do not set to this value 1: Select by the PSC2_2 bit	0: CAN0OUT 1: CAN1OUT[1]	Set to 0	Set to 0
Bits 3 to 7	Set to 00000b				

NOTE:
1. Set to 0 in M32C/87A. Do not set the bit 2 in the PSC2 register in M32C/87B. Write a 0, if necessary.

Table 25.6 Port P9 Peripheral Function Output Control

	PS3 Register	PSL3 Register	PSC3 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	Set to 0
Bit 1	0: P9_1/TB1IN/RXD3/SCL3 input/ ISRXD2/IEIN 1: Select by the PSL3_1 bit	0: SCL3 output 1: STXD3	Set to 0
Bit 2	0: P9_2/TB2IN/SRXD3/ SDA3 input 1: Select by the PSL3_2 bit	0: TXD3/SDA3 output 1: OUTC2_0/ISTXD2/IEOUT	Set to 0
Bit 3	0: P9_3/TB3IN/$\overline{CTS3}$/$\overline{SS3}$/DA0 1: $\overline{RTS3}$	0: Peripheral function input 1: DA0	Set to 0
Bit 4	0: P9_4/TB4IN/$\overline{CTS4}$/$\overline{SS4}$/DA1 1: $\overline{RTS4}$	0: Peripheral function input 1: DA1	Set to 0
Bit 5	0: P9_5/ANEX0/CLK4 input/ CAN1IN/$\overline{CAN1WU}$ 1: CLK4 output	0: Peripheral function input except ANEX0 1: ANEX0	Set to 0
Bit 6	0: P9_6/SRXD4/ANEX1/ SDA4 input 1: Select by the PSC3_6 bit	0: Peripheral function input except ANEX1 1: ANEX1	0: TXD4/SDA4 output 1: CAN1OUT[1]
Bit 7	0: P9_7/RXD4/$\overline{ADTRG}$/ SCL4 input 1: Select by the PSL3_7 bit	0: SCL4 output 1: STXD4	Set to 0

NOTE:
1. Set to 0 in M32C/87A and M32C/87B.

Table 25.7 Port P10 Peripheral Function Output Control (1)

	PS4 Register
Bit 0	0: P10_0/AN_0 1: RTP1_0
Bit 1	0: P10_1/AN_1 1: RTP1_1
Bit 2	0: P10_2/AN_2 1: RTP1_2
Bit 3	0: P10_3/AN_3 1: RTP1_3
Bit 4	0: P10_4/AN_4/$\overline{KI0}$ 1: RTP3_0
Bit 5	0: P10_5/AN_5/$\overline{KI1}$ 1: RTP3_1
Bit 6	0: P10_6/AN_6/$\overline{KI2}$ 1: RTP3_2
Bit 7	0: P10_7/AN_7/$\overline{KI3}$ 1: RTP3_3

Table 25.8 Port P10 Peripheral Function Output Control (2)

	PSC Register
Bit 7	0: P10_4 to P10_7 or $\overline{KI0}$ to $\overline{KI3}$ 1: AN_4 to AN_7

Table 25.9 Port P11 Peripheral Function Output Control

	PS5 Register	PSL5 Register
Bit 0	0: P11_0/INPC1_0 1: Select by the PSL5_0 bit	0: OUTC1_0/ISTXD1 1: Do not set to this value
Bit 1	0: P11_1/INPC1_1/ISCLK1 input 1: Select by the PSL5_1 bit	0: OUTC1_1/ISCLK1 output 1: Do not set to this value
Bit 2	0: P11_2/INPC1_2/ISRXD1 1: Select by the PSL5_2 bit	0: OUTC1_2 1: Do not set to this value
Bit 3	0: P11_3/INPC1_3 1: Select by the PSL5_3 bit	0: OUTC1_3 1: Do not set to this value
Bits 4 to 7	Set to 0000b	

Table 25.10 Port P12 Peripheral Function Output Control

	PS6 Register	PSL6 Register	PSC6 Register
Bit 0	0: P12_0 1: Select by the PSL6_0 bit	0: Select by the PSC6_0 bit 1: Do not set to this value	0: Do not set to this value 1: TXD6
Bit 1	0: P12_1/CLK6 input 1: Select by the PSL6_1 bit	0: Select by the PSC6_1 bit 1: Do not set to this value	0: Do not set to this value 1: CLK6 output
Bit 2	Set to 0		
Bit 3	0: P12_3/$\overline{CTS6}$ 1: Select by the PSL6_3 bit	0: Select by the PSC6_3 bit 1: Do not set to this value	0: Do not set to this value 1: $\overline{RTS6}$
Bits 4 to 7	Set to 0000b		

Table 25.11 Port P13 Peripheral Function Output Control

	PS7 Register	PSL7 Register
Bit 0	0: P13_0 1: Select by the PSL7_0 bit	0: OUTC2_4 1: Do not set to this value
Bit 1	0: P13_1 1: Select by the PSL7_1 bit	0: OUTC2_5 1: Do not set to this value
Bit 2	0: P13_2 1: Select by the PSL7_2 bit	0: OUTC2_6 1: Do not set to this value
Bit 3	0: P13_3 1: Select by the PSL7_3 bit	0: OUTC2_3 1: Do not set to this value
Bit 4	0: P13_4 1: Select by the PSL7_4 bit	0: OUTC2_0/ISTXD2/IEOUT 1: Do not set to this value
Bit 5	0: P13_5/ISRXD2/IEIN 1: Select by the PSL7_5 bit	0: OUTC2_2 1: Do not set to this value
Bit 6	0: P13_6/ISCLK2 input 1: Select by the PSL7_6 bit	0: OUTC2_1/ISCLK2 output 1: Do not set to this value
Bit 7	0: P13_7 1: Select by the PSL7_7 bit	0: OUTC2_7 1: Do not set to this value

Table 25.12 Port P14 Peripheral Function Output Control

	PS8 Register
Bit 0	0: P14_0/INPC1_4 1: OUTC1_4
Bit 1	0: P14_1/INPC1_5 1: OUTC1_5
Bit 2	0: P14_2/INPC1_6 1: OUTC1_6
Bit 3	0: P14_3/INPC1_7 1: OUTC1_7
Bits 4 to 7	Set to 0000b

Table 25.13 Port P15 Peripheral Function Output Control

	PS9 Register	PSL9 Register
Bit 0	0: P15_0/AN15_0 1: Select by the PSL9_0 bit	0: ISTXD0 1: TXD5
Bit 1	0: P15_1/AN15_1/ISCLK0 input/CLK5 input 1: Select by the PSL9_1 bit	0: ISCLK0 output 1: CLK5 output
Bit 2	Set to 0	
Bit 3	0: P15_3/AN15_3/$\overline{\text{CTS5}}$ 1: $\overline{\text{RTS5}}$	Set to 0
Bit 4	0: P15_4/AN15_4 1: Select by the PSL9_4 bit	0: Do not set to this value 1: TXD6
Bit 5	Set to 0	
Bit 6	0: P15_6/AN15_6/CLK6 input 1: CLK6 output	Set to 0
Bit 7	0: P15_7/AN15_7/$\overline{\text{CTS6}}$ 1: $\overline{\text{RTS6}}$	Set to 0

26. 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. The flash memory has the user ROM area and boot ROM area, and the rewrite control program for the standard serial I/O mode is stored in the boot ROM area.
Table 26.1 lists specifications of the flash memory. (See **Tables 1.1 to 1.4** for the items not listed in Table 26.1.) Table 26.2 lists overview of flash memory rewrite mode.

Table 26.1 Flash Memory Specifications

Item	Specification
Flash memory rewrite mode	3 modes (CPU rewrite mode, standard serial I/O mode, parallel I/O mode)
Erase unit	On a block basis (See Figure 26.1)
Program unit	16 bits, 8 bits(1)
Erase and program control method	Software commands control erasing and programming on the flash memory
Protect method	The lock bit protects each block in the flash memory
Number of commands	7 commands
Erase and program endurance	100 times(2)
Flash memory access disable function	ROM code protect function (parallel I/O mode) ID code check function (standard serial I/O mode)

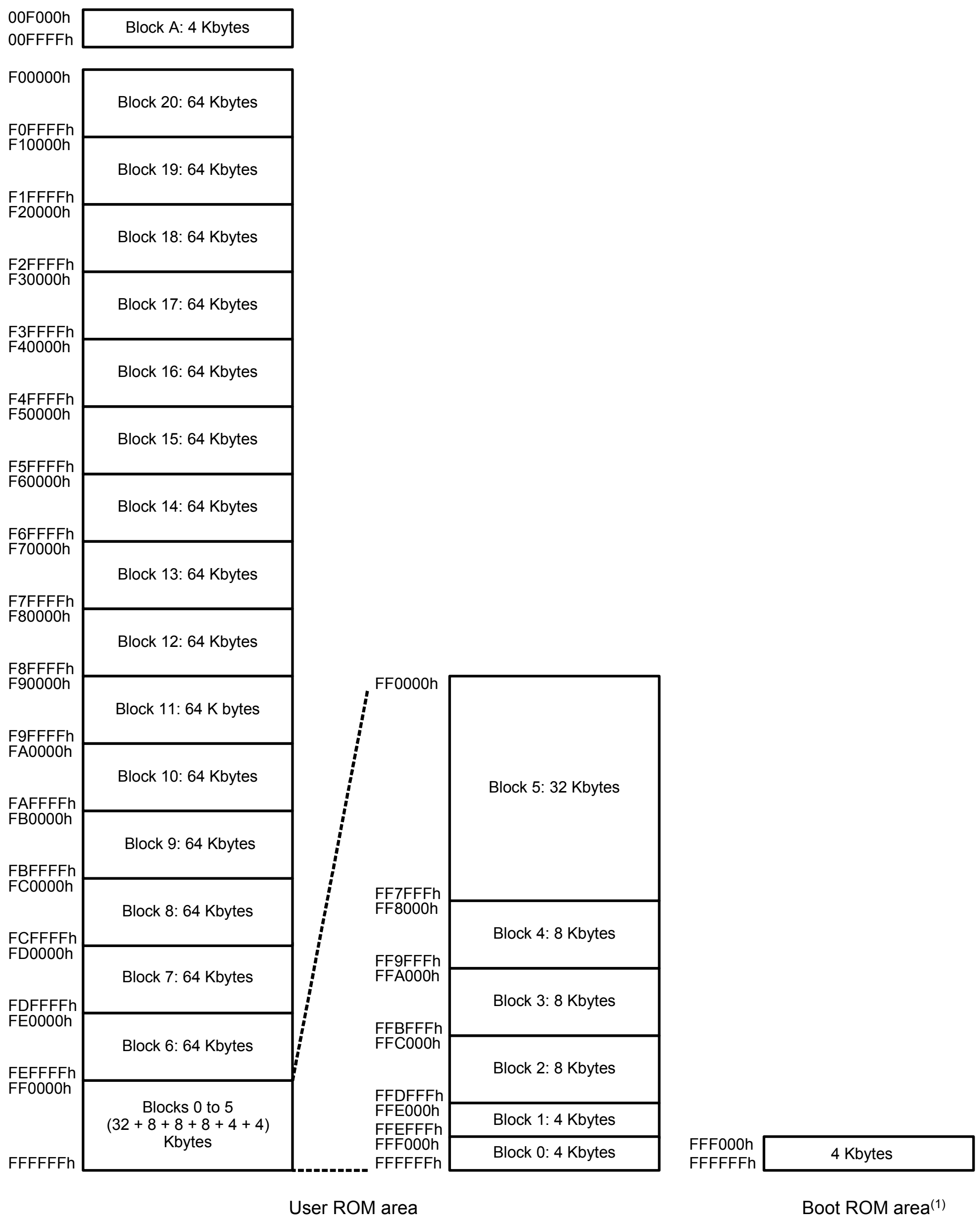
NOTES:
1. The flash memory can be programmed in 8-bit (byte) units in parallel I/O mode only.
2. The erase and program endurance is the number of erase operations performed on individual blocks. For example, if the block A is erased without programming, the erased and program count stands at one for the block A.

Table 26.2 Flash Memory Rewrite Mode Overview

Flash Memory Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	User ROM area is programmed by the CPU executing 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 and boot ROM areas are programmed using a dedicated parallel programmer.
Rewritable area	User ROM area	User ROM area	User ROM area Boot ROM area
Operating mode	Single-chip mode Memory expansion mode (EW0 mode) Boot mode (EW0 mode)	Boot mode	Parallel I/O mode
ROM programmer	–	Serial programmer	Parallel programmer

26.1 Memory Map

Figure 26.1 shows the flash memory map. The user ROM area has an area to store programs, and another 4-Kbyte area as the block A for data storage.
The user ROM area is divided into blocks, each of which can be protected (locked) from erasing or programming.
The user ROM area can be programmed in CPU rewrite mode, standard serial I/O mode, or parallel I/O mode.
The addresses of the boot ROM area are overlapped with the addresses of the user ROM area. The boot ROM area can only be rewritten in parallel I/O mode.



NOTES:
1. The rewrite control program for standard serial I/O mode is stored in the boot ROM area before shipment. This area can be rewritten only in parallel I/O mode.
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 26.1 Flash Memory Map

26.1.1 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 EPM (P5_5) pin and apply an "H" (pull-up) to the 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. If a given rewrite control program is written in the boot ROM area, the flash memory can be rewritten along the implemented system.

26.2 Functions to Prevent Access to Flash Memory

Parallel I/O mode has a ROM code protect function, and standard I/O mode has an ID code check function to prevent the flash memory from being read or programmed.

26.2.1 ROM Code Protect Function

The ROM code protect function disables reading or programming the contents of the flash memory in parallel I/O mode. To use ROM code protect function, set the ROMCP1 bits in the ROMCP address.

The ROMCP address is placed in a user ROM area. Figure 26.2 shows the ROMCP address.

26.2.2 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 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 ID code check function, write the program which specifies the ID code to these addresses.

NOTE:
1. FFFFFFFFh is the factory default setting.

ROM Code Protect Control Address[5]

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

Symbol: ROMCP
Address: FFFFFFh
Factory setting: FFh[4]

Bit Symbol	Bit Name	Function	RW
− (b5-b0)	Reserved bits	Set to 1 to use ROM protect function	RW
ROMCP1	ROM code protect set bits[1][2][3]	b7 b6 0 0: ROM code protect enabled 0 1: ROM code protect enabled 1 0: ROM code protect enabled 1 1: ROM code protect disabled	RW

NOTES:
1. When the ROM code protect is enabled by setting the ROMCP1 bits to values other than 11b, the flash memory is protected against reading or rewriting the contents in parallel I/O mode.
2. Set the bits 5 to 0 to 111111b when the ROMCP1 bits are set to other than 11b.
3. To disable the ROM code protect, erase the block including the ROMCP address in standard serial I/O mode or CPU rewrite mode.
4. The ROMCP address is set to FFh when the block including the ROMCP address is erased.
5. When the ROMCP address is set to 00h or FFh, the ROM code protect function is disabled.

Figure 26.2 ROMCP Address

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	ROMCP	Reset Vector

4 bytes

Figure 26.3 Addresses for Stored ID Codes

26.3 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 26.1 can be programmed. The boot ROM area cannot be rewritten. EW0 mode and EW1 mode are provided as CPU rewrite mode.

Table 26.3 lists specifications of EW0 mode and EW1 mode. Figures 26.4 and 26.5 show associated registers. Figure 26.6 shows a setting procedure for EW0 mode. Figure 26.7 shows a setting procedure for EW1 mode. Figure 26.8 shows a setting procedure to enter and exit low power mode.

Table 26.3 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 • Boot mode	• Single-chip mode
Areas where a rewrite program can be stored	• User ROM area (Single-chip mode, memory expansion mode) • Boot ROM area (Boot mode)	• User ROM area
Software command	All commands are available.	• All commands, except read status register command, are available.
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	Operating	In a hold state (Stop) (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 (it is acknowledged after completion of erase or program operation.)

NOTES:
1. In both the EW0 mode and EW1 mode, when an $\overline{NMI}$ 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.

26.3.1 Flash Memory Control Register (FMR0 and FMR1 Registers)

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
			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 $\overline{NMI}$ 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 26.4 FMR0 Register

26.3.1.1 FMR00 Bit

The FMR00 bit indicates the operating status of the flash memory. It becomes 0 while the program command, block erase command, lock bit program command, or read lock bit status command is being executed, otherwise, it is 1.

26.3.1.2 FMR01 Bit

The flash memory can accept a command when the FMR01 bit is set to 1 (CPU rewrite mode enabled). Set the FMR05 bit to 1 (user ROM area accessed) as well if the MCU is in boot mode.

26.3.1.3 FMR02 Bit

The lock bit becomes invalid by setting the FMR02 bit to 1 (lock bit disabled). (Refer to **26.3.3 Data Protect Function** for details.) The lock bit becomes valid by setting the FMR02 bit to 0 (lock bit enabled).
The FMR02 bit does not change a lock bit status but disables a lock bit function. When the block erase command is executed while the FMR02 bit is set to 1, the lock bit status changes from 0 (locked) to 1 (unlocked).

26.3.1.4 FMSTP Bit

The FMSTP bit is used to initialize the flash memory control circuits, and also to reduce power consumption in the flash memory. Access to the flash memory is disabled when the FMSTP bit is set to 1 (flash memory stopped). 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:

- 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.

Figure 26.8 shows a flow chart illustrating entering and exiting low power mode. Follow the procedure on the flow chart.
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.

26.3.1.5 FMR05 Bit

The FMR05 bit selects access to either the boot ROM area or user ROM area in boot mode. Set to 0 to access (read) the boot ROM area or set to 1 to access (read, write, or erase) the user ROM area.

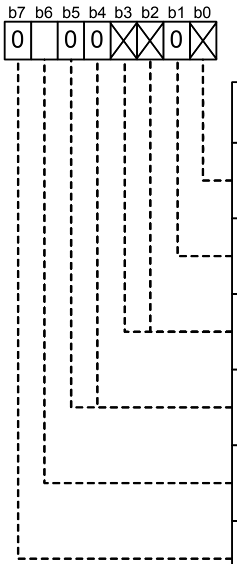
26.3.1.6 FMR06 Bit

The FMR06 bit is a read-only bit indicating the status of a program operation. The FMR06 bit becomes 1 when a program error occurs; otherwise, it is 0. Refer to **26.3.5 Full Status Check** for details.

26.3.1.7 FMR07 Bit

The FMR07 bit is a read-only bit indicating the status of an erase operation. The FMR07 bit becomes 1 when an erase error occurs; otherwise, it is 0. Refer to **26.3.5 Full Status Check** for details.

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.	–
FMR11	EW1 mode select bit(1)	0: EW0 mode 1: EW1 mode	RW
– (b3-b2)	Reserved bits	Read as undefined value.	–
– (b5-b4)	Reserved bits	Set to 0	RW
FMR16	Lock bit status flag	0: Locked 1: Unlocked	RO
– (b7)	Reserved bit	Set to 0	RW

NOTE:
1. To set the FMR11 bit to 1, write a 1 to the FMR11 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. Set the FMR11 bit while "H" is applied to the $\overline{NMI}$ pin. When the FMR01 bit is set to 0, the FMR11 bit also becomes 0.

Figure 26.5 FMR1 Register

26.3.1.8 FMR11 Bit

When the FMR11 bit is set to 0 (EW0 mode), the flash memory enters EW0 mode.
When the FMR11 bit is set to 1 (EW1 mode), the flash memory enters EW1 mode.

26.3.1.9 FMR16 Bit

The FMR16 bit is a read-only bit indicating the execution result of the read lock bit status command.
When a block, on where the read lock bit status command is executed, is locked, the FMR16 bit becomes 0.
When a block, on where the read lock bit status command is executed, is unlocked, the FMR16 bit becomes 1.

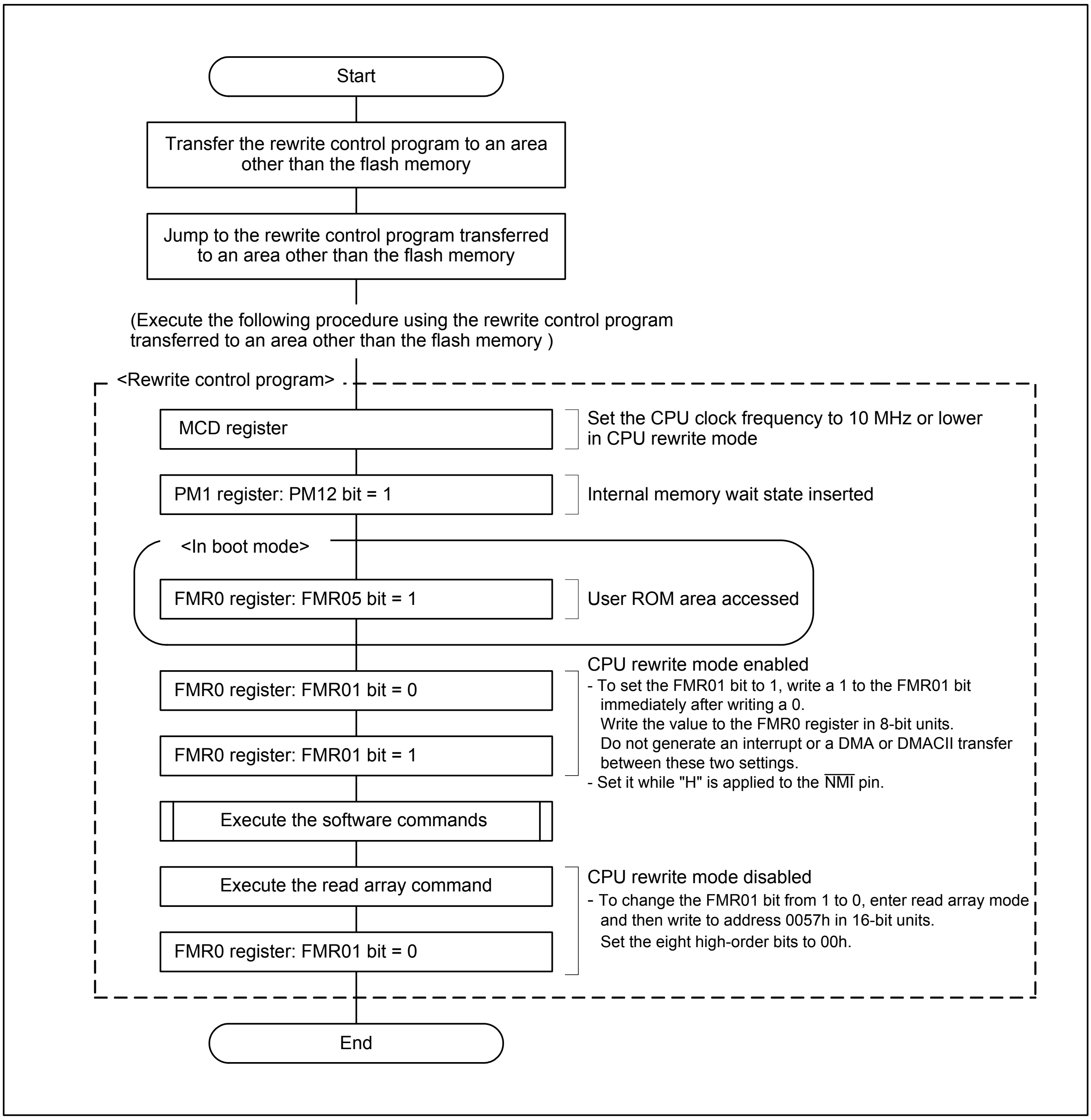
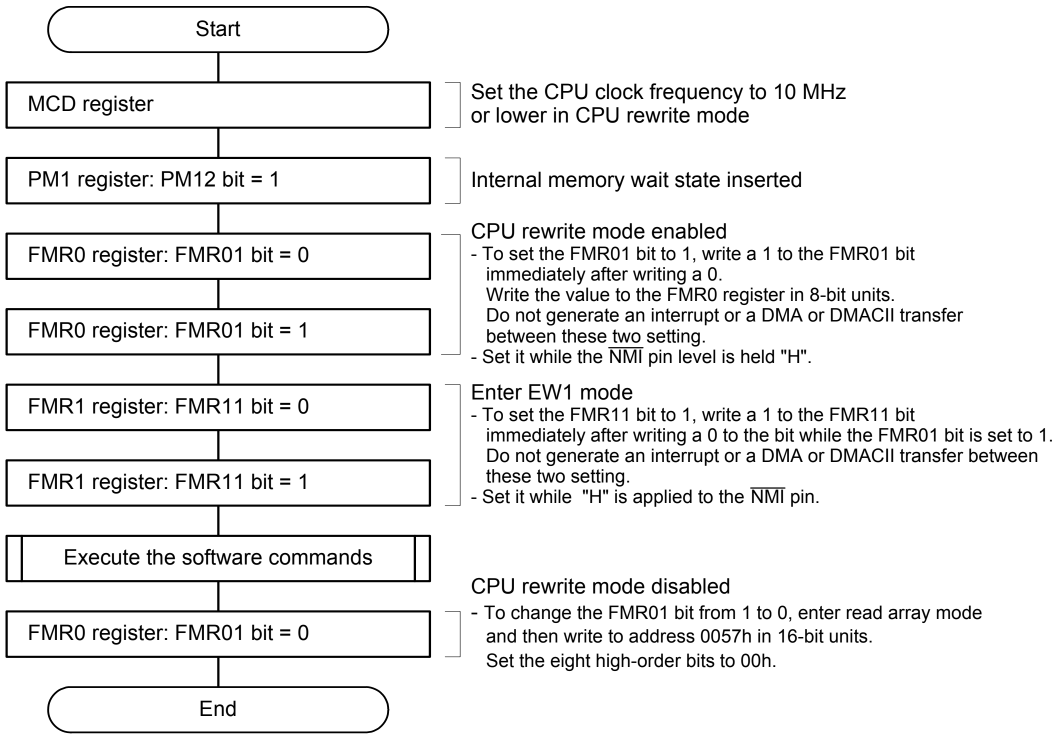


Figure 26.6 Setting Procedure for EW0 Mode



NOTE:
1. Do not use EW1 mode in memory expansion mode or boot mode.

Figure 26.7 Setting Procedure for EW1 Mode

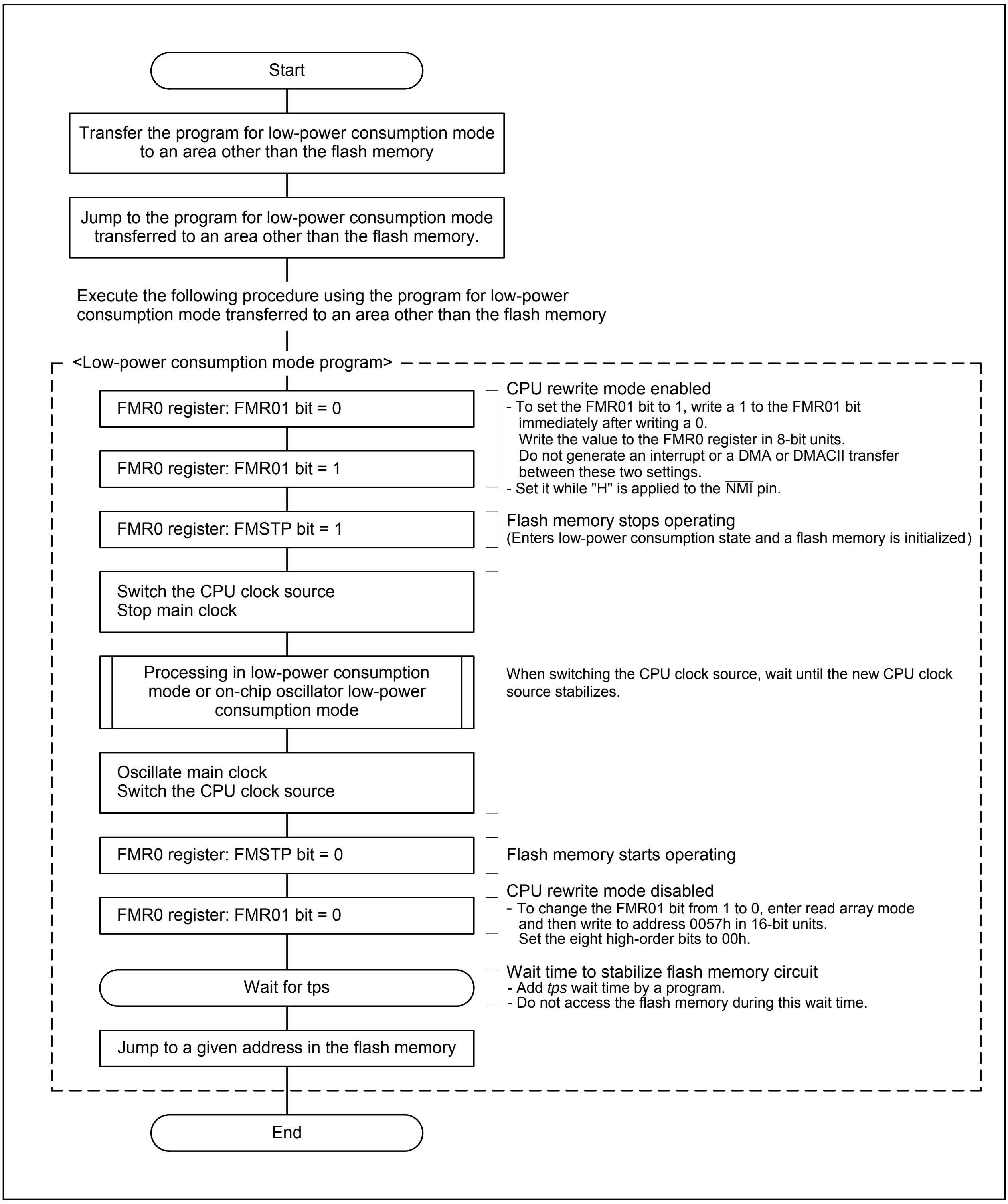


Figure 26.8 Setting Procedure to Enter and Exit Low Power Mode

26.3.2 Software Commands

Read or write commands and data from or to 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 26.4 Software Commands

Software Command	First Bus Cycle			Second Bus Cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read array	Write	x	xxFFh	–	–	–
Read status register	Write	x	xx70h	Read	x	SRD
Clear status register	Write	x	xx50h	–	–	–
Program	Write	WA	xx40h	Write	WA	WD
Block erase	Write	x	xx20h	Write	BA	xxD0h
Lock bit program	Write	BA	xx77h	Write	BA	xxD0h
Read lock bit status	Write	x	xx71h	Write	BA	xxD0h

SRD: Data in the status register (D7 to D0)
WA: Write address (The address specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.)
WD: 16-bit write data
BA: Highest-order even address of a block
x: Any even address in the user ROM area
xx: 8 high-order bits of command code (ignored)

26.3.2.1 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.

26.3.2.2 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 **26.3.4 Status Register (SRD Register)** for details). To read the status register, read an even address in the user ROM area. Do not execute this command in EW1 mode.

26.3.2.3 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.

26.3.2.4 Program Command

The program command is used to write data to the flash memory in 16-bit units.
A program operation (program and verify data) starts by writing the command code xx40h in the first bus cycle and data to the write address in the second bus cycle. The address value specified in the first bus cycle must be the same even address as the write address specified in the second bus cycle.
The FMR00 bit in the FMR0 register 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 **26.3.5 Full Status Check** for details.)
Do not execute the program command to the same address more than once without executing the block erase command. Figure 26.9 shows a flow chart of the program command.
The lock bit can protect each block from being programmed inadvertently. (Refer to **26.3.3 Data Protect Function** for details.)
In EW1 mode, do not execute this 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.

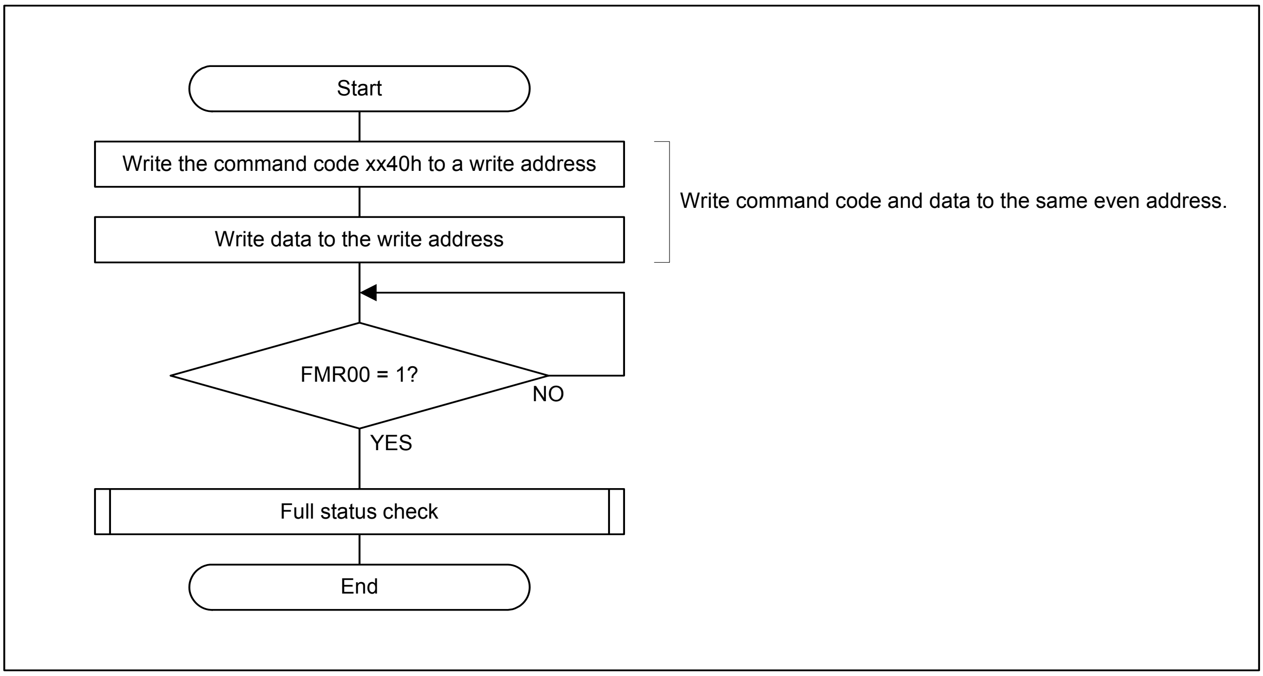


Figure 26.9 Program Command

26.3.2.5 Block Erase Command

The block erase command is used to erase a specified block.
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 **26.3.5 Full Status Check** for details.)
Figure 26.10 shows a flow chart of block erase command.
The lock bit can protect each block from being erased inadvertently. (Refer to **26.3.3 Data Protect Function** for details.)
In EW1 mode, do not execute this 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.

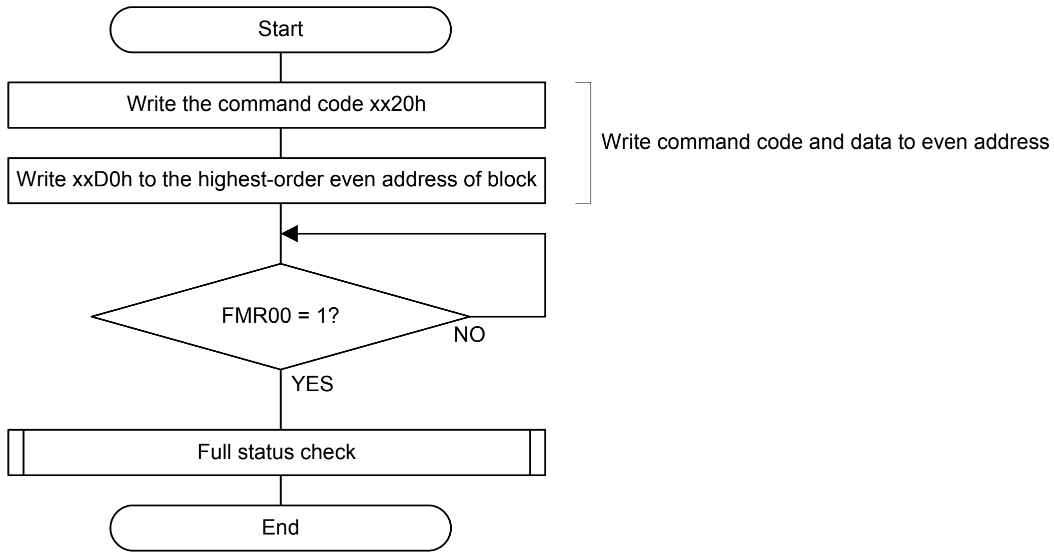


Figure 26.10 Block Erase Command

26.3.2.6 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 26.11 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 **26.3.3 Data Protect Function** for information on lock bit functions and how to set it to 1 (unlocked).
In EW1 mode, do not execute this 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.

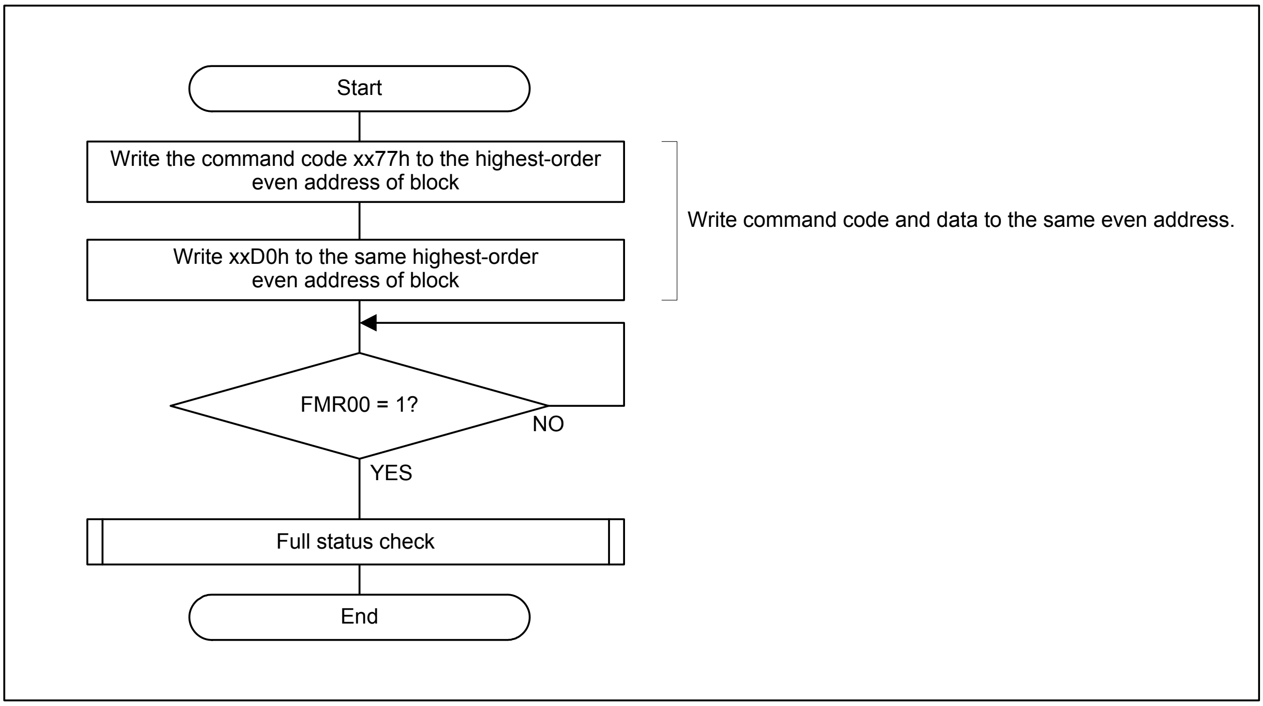


Figure 26.11 Lock Bit Program Command

26.3.2.7 Read Lock Bit Status Command

The read lock bit status command reads a lock bit status of a given block.
By writing the command code xx71h in the first bus cycle and xxD0h to the highest-order even address of a block in the second bus cycle, the FMR16 bit in the FMR1 register stores information on whether the lock bit of the block is locked or not. Read the FMR16 bit after the FMR00 bit in the FMR0 register becomes 1 (ready).
Figure 26.12 shows a flow chart of read lock bit status command.

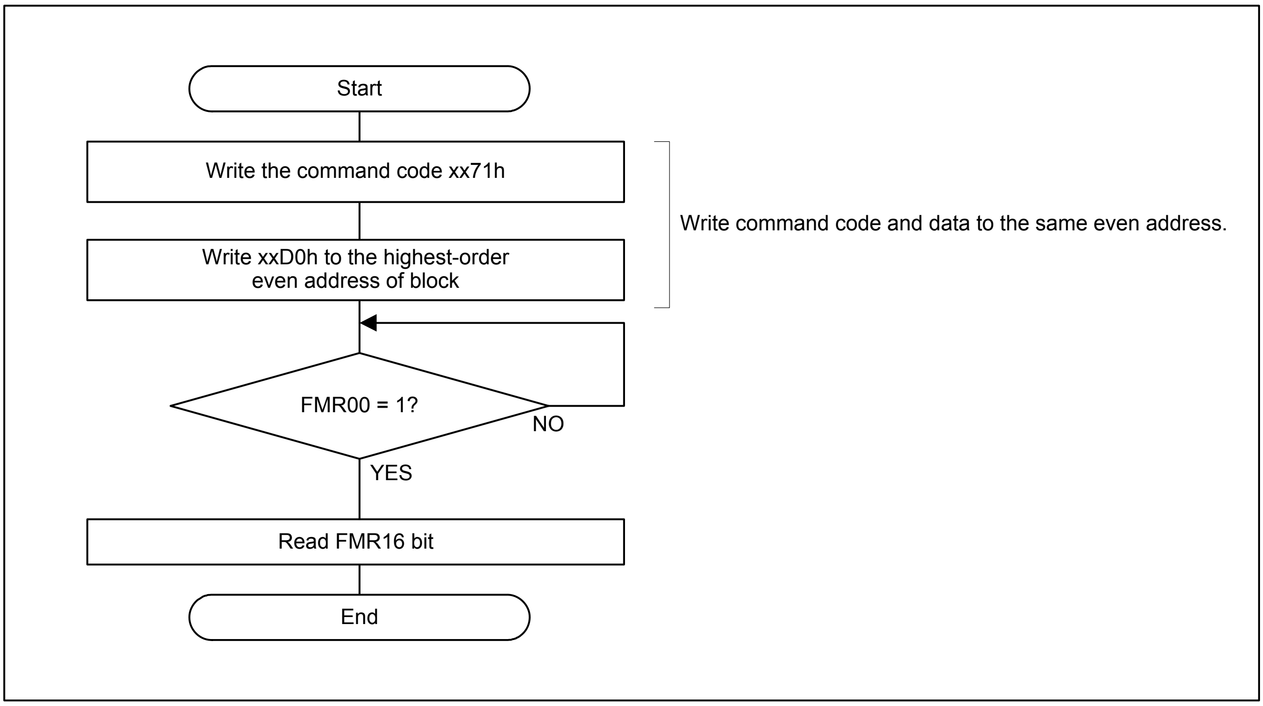


Figure 26.12 Read Lock Bit Status Command

26.3.3 Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit protects (locks) each block individually against erasing and programming. This prevents data from being inadvertently erased from or programmed to the flash memory. The following is the block conditions controlled by the lock bit.

When the FMR02 bit in the FMR0 register is set to 0 (lock bit enabled);

- If lock bit data is set to 0, the block is locked (block is protected against erasing and programming).
- If lock bit data is set to 1, the block is unlocked (block can be erased or programmed).

When the FMR02 bit in the FMR0 register is set to 1 (lock bit disabled);

- The block is unlocked regardless of the lock bit data status (block can be erased or programmed).

When the block erase command is executed while the FMR02 bit is set to 1, the target block is erased regardless of the lock bit data status. The lock bit data of the target block becomes 1 when the block erase operation is completed.

26.3.4 Status Register (SRD Register)

In EW0 mode, the Status Register value is returned by reading the flash memory after executing the commands shown below.

- Read status register command
- Program command
- Block erase command
- Lock bit program command

The Status Register 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.

26.3.4.1 Sequencer Status (SR7 Bit, FMR00 Bit)

The sequencer status bit indicates the operating status of the flash memory. It becomes 0 while the program command, block erase command, lock bit program command, or read lock bit status command is being executed; otherwise, it is 1.

26.3.4.2 Erase Status (SR5 Bit, FMR07 Bit)

Refer to **26.3.5 Full Status Check**.

26.3.4.3 Program Status (SR4 Bit, FMR06 Bit)

Refer to **26.3.5 Full Status Check**.

Table 26.5 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, and read lock bit status command cannot be accepted by the flash memory.

26.3.5 Full Status Check

If an error occurs, bits FMR07 and FMR06 in the FMR0 register become 1, indicating the occurrence of an error. Therefore, by checking these status bits (full status check), the execution result can be confirmed.
Table 26.6 lists error types and FMR0 register values. Figure 26.13 shows a flow chart of the full status check and handling procedure for each error.

Table 26.6 Errors and FMR0 Register Values

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 or block erase 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

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.

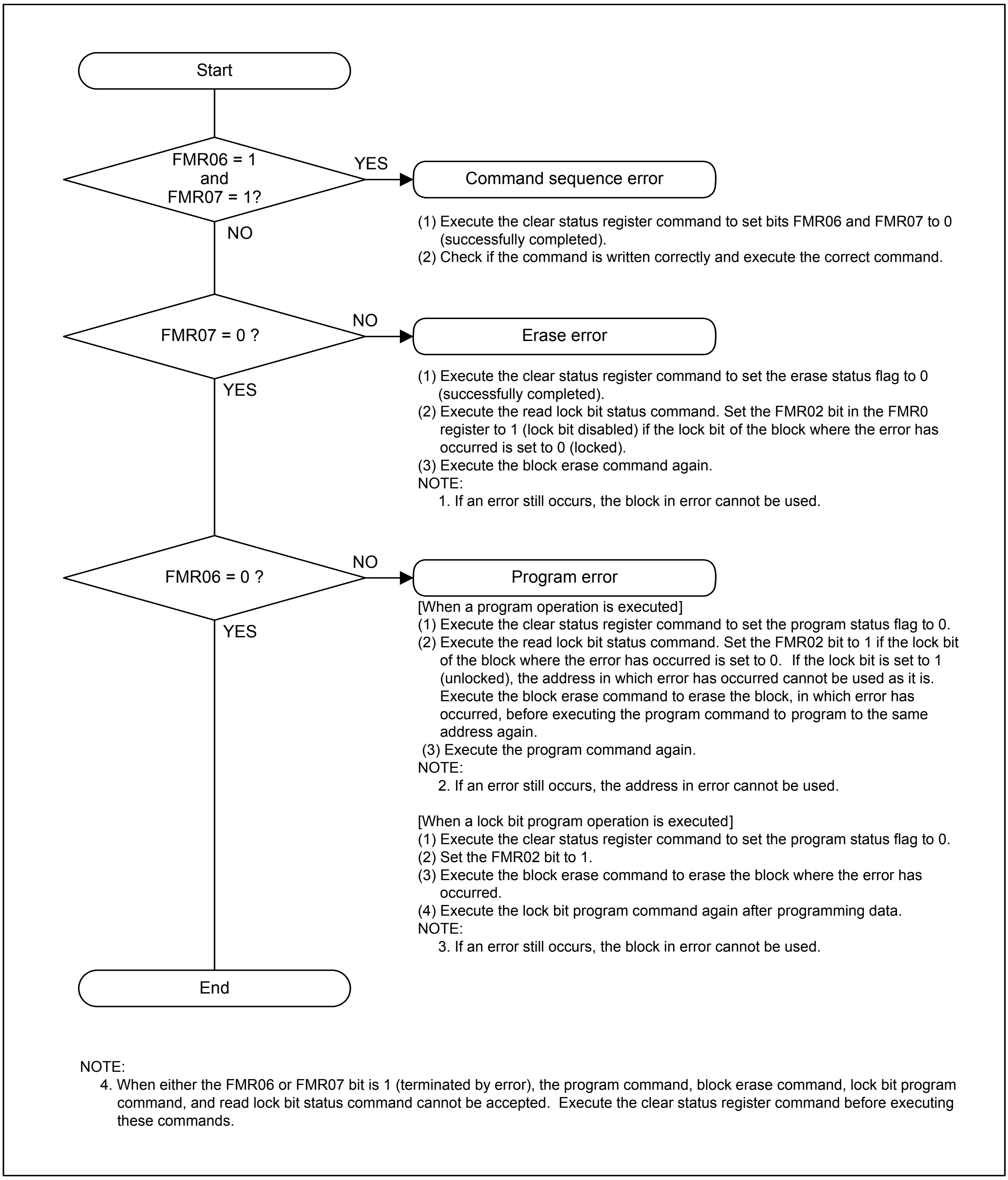


Figure 26.13 Full Status Check and Handling Procedure for Each Error

26.4 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/87 Group (M32C/87, M32C/87A, M32C/87B).
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 26.7 lists pin functions for flash memory standard serial I/O mode. Figures 26.14 to 26.16 show pin connections for standard serial I/O mode.

Table 26.7 Pin Functions for Flash Memory Standard Serial I/O Mode

Pin Name	Function	I/O Type	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)

I: Input O: Output I/O: Input and output

NOTE:

1. These pins are provided in the 144-pin package only.

Mode setting

Signal	Value
CNVSS	VCC1
EPM	VSS
RESET	VSS→VCC1
CE	VCC2

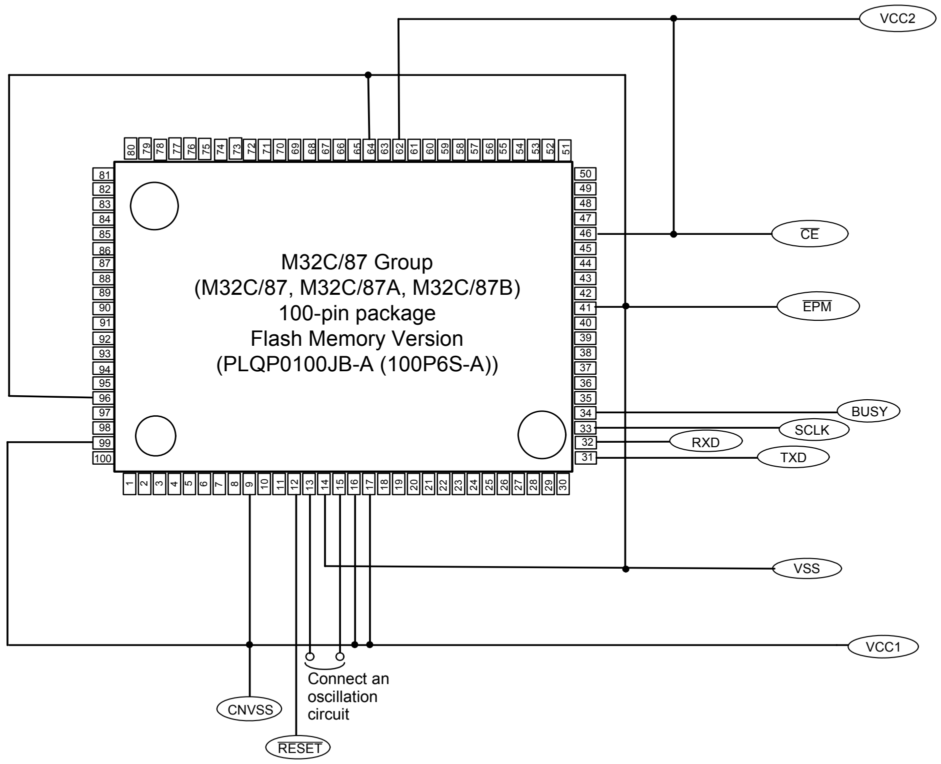


Figure 26.14 Pin Connections in Standard Serial I/O Mode (1/3)

Mode setting

Signal	Value
CNVSS	VCC1
EPM	VSS
RESET	VSS→VCC1
CE	VCC2

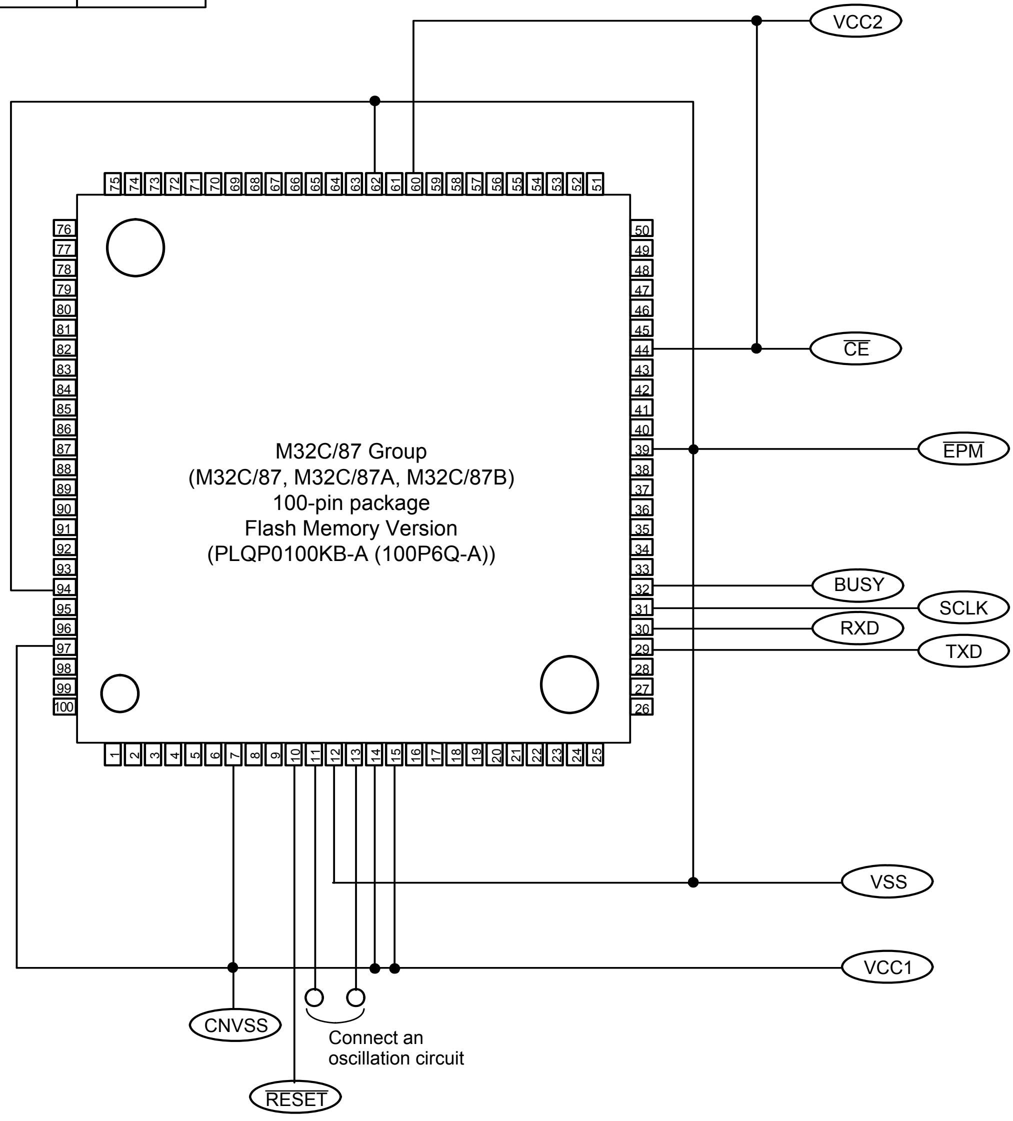


Figure 26.15 Pin Connections in Standard Serial I/O Mode (2/3)

Mode setting

Signal	Value
CNVSS	VCC1
$\overline{\text{EPM}}$	VSS
$\overline{\text{RESET}}$	VSS→VCC1
$\overline{\text{CE}}$	VCC2

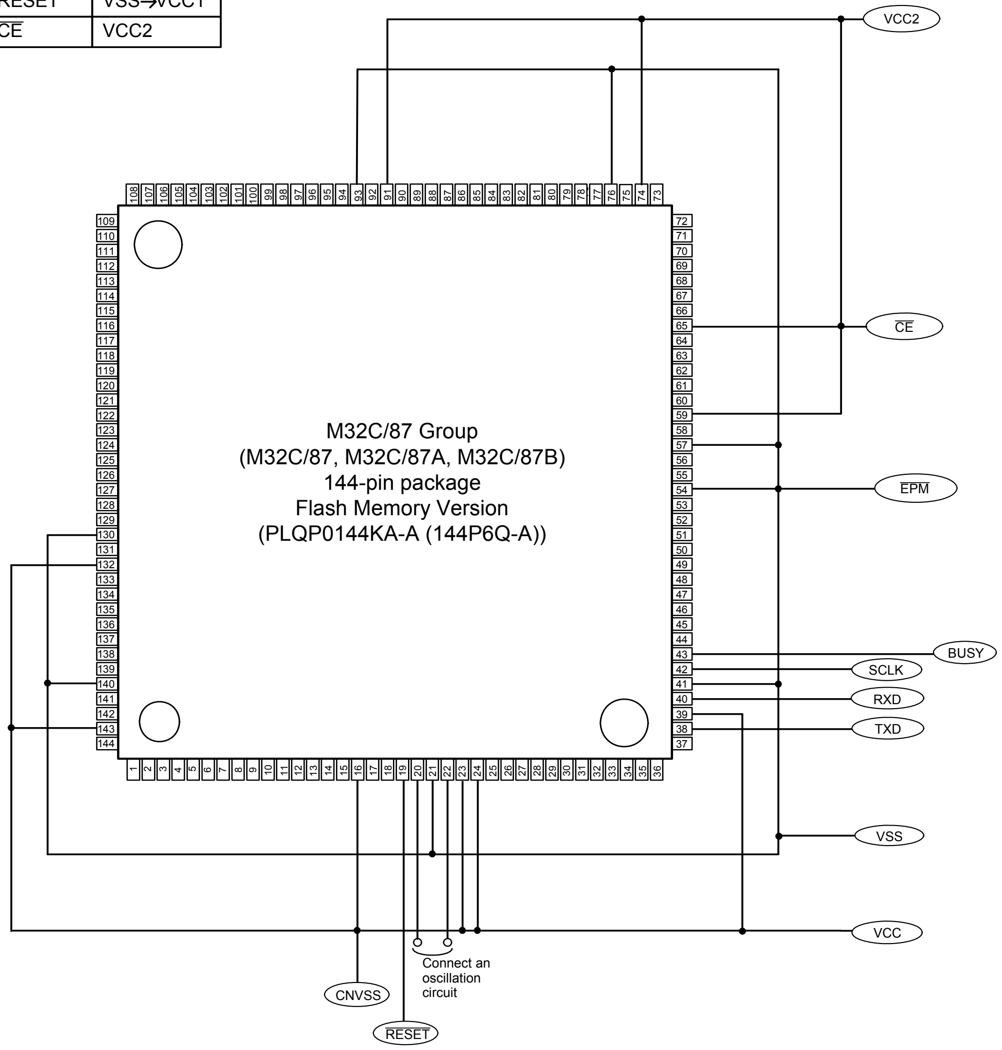
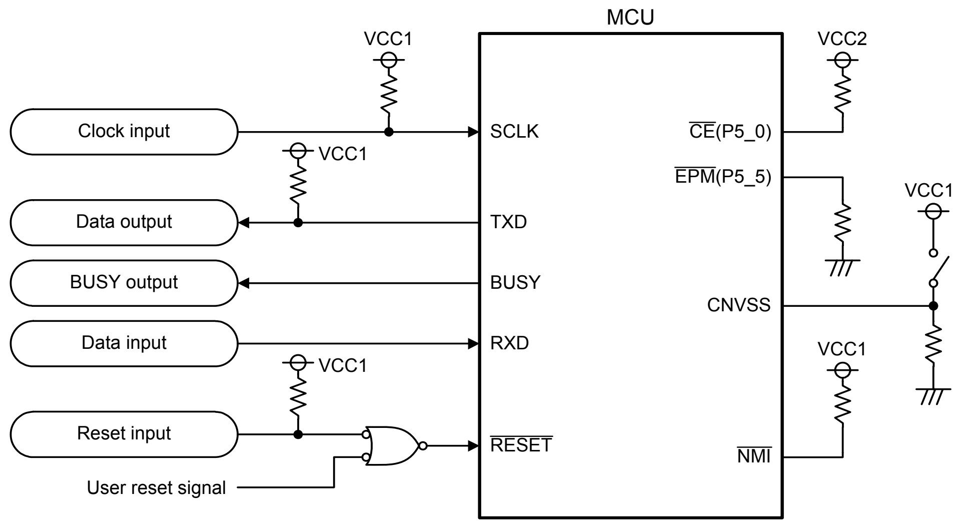


Figure 26.16 Pin Connections in Standard Serial I/O Mode (3/3)

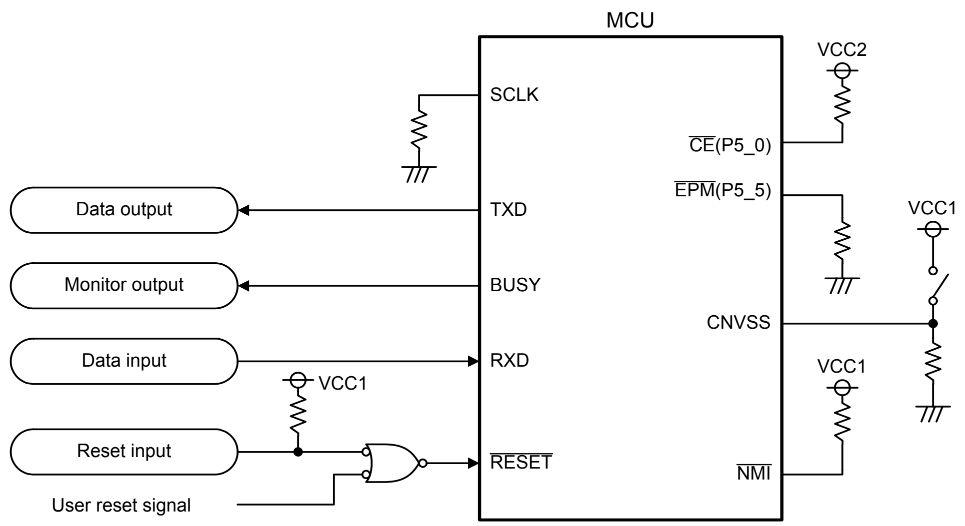
26.4.1 Pin Handling in Standard Serial I/O Mode

Figure 26.17 shows an example of a pin handling in standard serial I/O mode 1. Figure 26.18 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 26.17 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 26.18 Pin Handling in Standard Serial I/O Mode 2

26.5 Parallel I/O Mode

In parallel I/O mode, the user ROM area and the boot ROM area can be programmed by using a parallel programmer supporting the M32C/87 Group (M32C/87, M32C/87A, M32C/87B). 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.

26.5.1 Boot ROM Area

The boot ROM area has one 4K-byte block. The rewrite control program for standard serial I/O mode is stored in the boot ROM area in factory default configuration. Do not rewrite the boot ROM area to use the serial programmer.

In parallel I/O mode, the boot ROM area is allocated in addresses FFF000h to FFFFFFh. Rewrite only this address block if it is necessary to rewrite the boot ROM area. (Do not access other than addresses FFF000h to FFFFFFh.)

27. Electrical Characteristics

Table 27.1 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	$\overline{RESET}$, 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 27.2 Recommended Operating Conditions (1/3)
(VCC1 = VCC2 = 3.0 to 5.5 V, Topr = -20 to 85°C unless otherwise specified)

Symbol	Parameter		Standard Min.	Standard Typ.	Standard Max.	Unit
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, $\overline{RESET}$, 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, $\overline{RESET}$, 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 27.3 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 27.4 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 = 4.2 to 5.5V	0		32	MHz
		VCC1 = 3.0 to 5.5V	0		24	MHz
f(XIN)	Main clock input oscillation frequency	VCC1 = 4.2 to 5.5V	0		32	MHz
		VCC1 = 3.0 to 5.5V	0		24	MHz
f(XCIN)	Sub clock frequency			32.768	50	kHz
f(Ring)	On-chip oscillator frequency			1		MHz
f(VCO)	VCO clock frequency (PLL frequency synthesizer)		20		80	MHz
f(PLL)	PLL clock frequency	VCC1 = 4.2 to 5.5V	10		32	MHz
		VCC1 = 3.0 to 5.5V	10		24	MHz
tsu(PLL)	Wait time to stabilize PLL frequency synthesizer	VCC1 = 5.0V			5	ms
		VCC1 = 3.3V			10	ms

VCC1 = VCC2 = 5V

Table 27.5 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			Measurement Condition	Standard Min.	Standard Typ.	Standard Max.	Unit
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		1.6		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	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT8, ADTRG, CTS0 to CTS6, CLK0 to CLK6, TA0OUT to TA4OUT, NMI, KI0 to KI3, RXD0 to RXD6, SCL0 to SCL4, SDA0 to SDA4, INPC1_0 to INPC1_7, ISCLK0 to ISCLK2, ISRXD0 to ISRXD2, IEIN, CAN0IN, CAN1IN, CAN1WU			0.2		1.0	V
		RESET			0.2		1.8	V

NOTE:
1. P11 to P15 are provided in the 144-pin package only.

VCC1 = VCC2 = 5V

Table 27.6 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		Measurement 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, $\overline{RESET}$, 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, $\overline{RESET}$, 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	167	kΩ
RfXIN	Feedback resistance	XIN			1.5		MΩ
RfXCIN	Feedback resistance	XCIN			10		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 27.7 Electrical Characteristics (3/3)
(VCC1 = VCC2 = 5.5 V, VSS = 0 V, Topr = 25°C)

Symbol	Parameter	Measurement Condition[1]		Standard Min.	Standard Typ.	Standard Max.	Unit
ICC	Power supply current	Flash memory version	f(CPU) = 32 MHz		32	45	mA
			f(CPU) = 16 MHz		19		mA
			f(CPU) = 8 MHz		12		mA
			f(CPU) = f(Ring) In on-chip oscillator low-power consumption mode		2.6		mA
			f(CPU) = 32 kHz In low-power consumption mode While flash memory is operating		430		µA
			f(CPU) = 32 kHz In low-power consumption mode While flash memory is stopped[2]		30		µA
			Wait mode: f(CPU) = f(Ring) After entering wait mode from on-chip oscillator low-power consumption mode		50		µA
			Stop mode (while clock is stopped)		0.8	5	µA
			Stop mode (while clock is stopped) Topr = 85°C			50	µA
		Mask ROM version	f(CPU) = 32 MHz		32	45	mA
			f(CPU) = 16 MHz		19		mA
			f(CPU) = 8 MHz		12		mA
			f(CPU) = f(Ring) In on-chip oscillator low-power consumption mode		1		mA
			f(CPU) = 32 kHz In low-power consumption mode		30		µA
			Wait mode: f(CPU) = f(Ring) After entering wait mode from on-chip oscillator low-power consumption mode		50		µA
			Stop mode (while clock is stopped)		0.8	5	µA
			Stop mode (while clock is stopped) Topr = 85°C			50	µA

NOTES:
1. In single-chip mode, leave the output pins open and connect the input pins to VSS.
2. Value is obtained when setting the FMSTP bit in the FMR0 register to 1 (flash memory stopped) and running the program on RAM.

VCC1 = VCC2 = 5V

Table 27.8 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		8		40	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			2		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 27.9 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 27.10 Flash Memory Electrical Characteristics (VCC1 = 4.5 V to 5.5 V, 3.0 to 3.6 V, Topr = 0 to 60°C unless otherwise specified)

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
–	Erase and program endurance[1]		100			times
–	Word program time (16 bits) (VCC1 = 5.0 V, Topr = 25°C)			25	300	µs
–	Lock bit program time			25	300	µs
–	Block erase time (VCC1 = 5.0 V, Topr = 25°C)	4-Kbyte block		0.3	4	s
		8-Kbyte block		0.3	4	s
		32-Kbyte block		0.5	4	s
		64-Kbyte block		0.8	4	s
tps	Wait time to stabilize flash memory circuit				15	µs
–	Data hold time (Topr = -40 to 85°C)		10			years

NOTE:

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 a word data 2,048 times, each to a different address, this counts as one erase and program time. Data can not be programmed to the same address more than once without erasing the block. (rewrite prohibited)

VCC1 = VCC2 = 5V

Table 27.11 Voltage Detection Circuit Electrical Characteristics (VCC1 = VCC2 = 3.0 to 5.5 V, VSS = 0 V, Topr = 25°C unless otherwise specified)

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet4	Vdet4 detection voltage	VCC1 = 3.0 V to 5.5 V	3.3	3.8	4.4	V
Vdet3	Vdet3 detection voltage			3.0		V
Vdet3s	Hardware reset 2 hold voltage				2.0	V
Vdet3r	Hardware reset 2 release voltage			3.1		V

NOTES:
1. Vdet4 > Vdet3
2. Vdet3r > Vdet3 is not guaranteed.

Table 27.12 Power Supply Circuit 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(S-R)	Wait time to release hardware reset 2	VCC1 = Vdet3r to 5.5 V		6(1)	20	ms
td(E-A)	Start-up time for Vdet3 and Vdet4 detection circuit	VCC1 = 3.0 to 5.5 V			20	µs

NOTE:
1. When VCC1 = 5 V

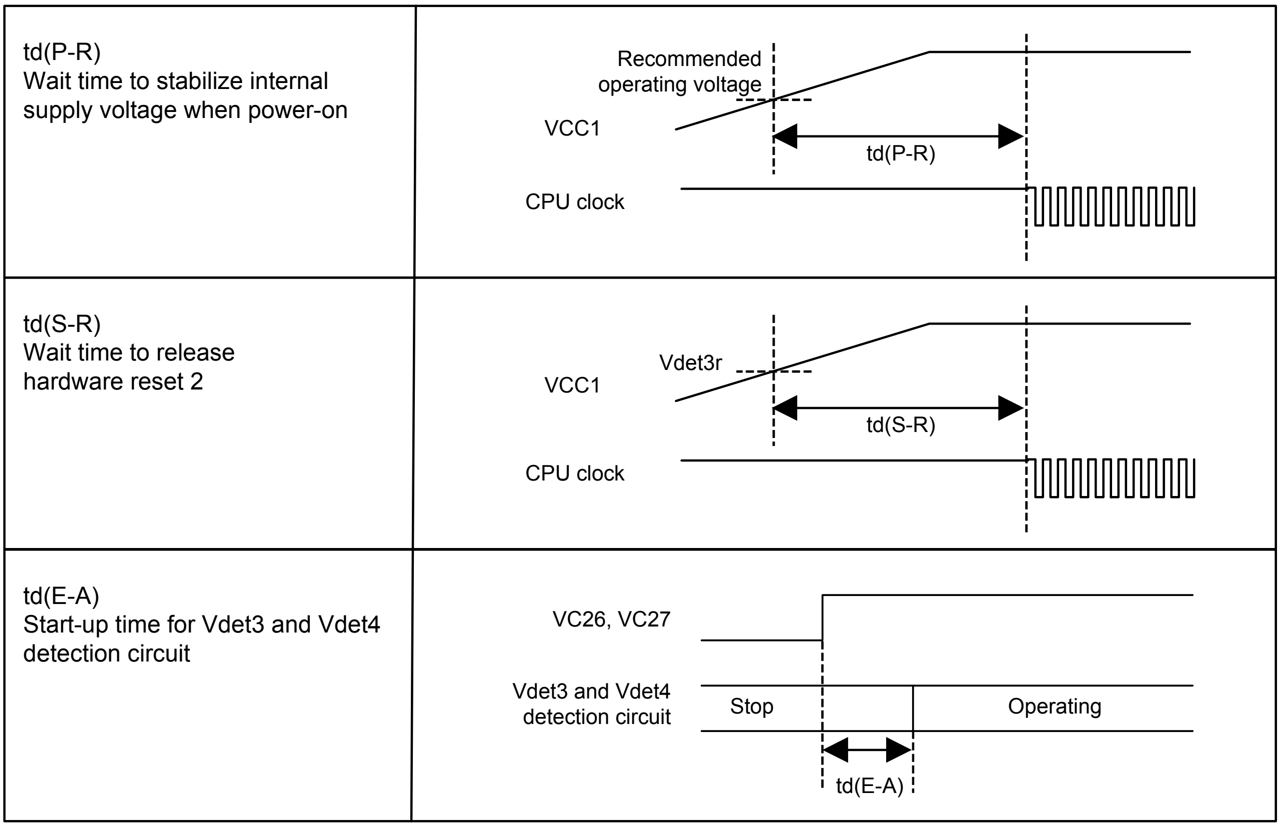


Figure 27.1 Power Supply Timing Diagram

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 27.13 External Clock Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External clock input cycle time	31.25		ns
tw(H)	External clock input high (“H”) pulse width	13.75		ns
tw(L)	External clock input low (“L”) pulse width	13.75		ns
tr	External clock rise time		5	ns
tf	External clock fall time		5	ns

Table 27.14 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 27.15 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 27.16 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 27.17 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 27.18 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 27.19 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 27.20 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 27.21 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 27.22 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 27.23 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

Table 27.24 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	70		ns
th(C-D)	RXDi input hold time	90		ns

i = 0 to 6

Table 27.25 Intelligent I/O Communication Function (Groups 0 and 1)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	ISCLKi input cycle time	600		ns
tw(CKH)	ISCLKi input high ("H") pulse width	300		ns
tw(CKL)	ISCLKi input low ("L") pulse width	300		ns
td(C-Q)	ISTXDi output delay time		100	ns
th(C-Q)	ISTXDi output hold time	0		ns
tsu(D-C)	ISRXDi input setup time	100		ns
th(C-D)	ISRXDi input hold time	100		ns

i = 0, 1

Table 27.26 Intelligent I/O Communication Function (Group 2)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	ISCLK2 input cycle time	600		ns
tw(CKH)	ISCLK2 input high ("H") pulse width	300		ns
tw(CKL)	ISCLK2 input low ("L") pulse width	300		ns
td(C-Q)	ISTXD2 output delay time		180	ns
th(C-Q)	ISTXD2 output hold time	0		ns
tsu(D-C)	ISRXD2 input setup time	150		ns
th(C-D)	ISRXD2 input hold time	100		ns

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 27.27 External Interrupt $\overline{INTi}$ Input (Edge Sensitive)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(INH)	$\overline{INTi}$ input high ("H") pulse width	250		ns
tw(INL)	$\overline{INTi}$ input low ("L") pulse width	250		ns

i = 0 to 8(1)
NOTE:
1. $\overline{INT6}$ to $\overline{INT8}$ are provided in the 144-pin package only.

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 27.28 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.

$$tac1(RD\text{-}DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) + 1)$$

$$tac1(AD\text{-}DB) = \frac{10^9 \times n}{f(BCLK)} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ n = a + b)$$

$$tac2(RD\text{-}DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) - 1)$$

$$tac2(AD\text{-}DB) = \frac{10^9 \times p}{f(BCLK) \times 2} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ p = \{(a + b - 1) \times 2\} + 1)$$

VCC1 = VCC2 = 5V

Switching Characteristics
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 27.29 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 27.2		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.

$$th(WR\text{-}DB) = \frac{10^9}{f(BCLK) \times 2} - 15 \text{ [ns]}$$

$$th(WR\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

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 = 5V

Switching Characteristics
(VCC1 = VCC2 = 4.2 to 5.5 V, VSS = 0 V, Topr = -20 to 85°C unless otherwise specified)

Table 27.30 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 27.2		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.

$$th(RD\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(RD\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}DB) = \frac{10^9}{f(BCLK) \times 2} - 15 \text{ [ns]}$$

2. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$td(DB\text{-}WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ m = (b \times 2) - 1)$$

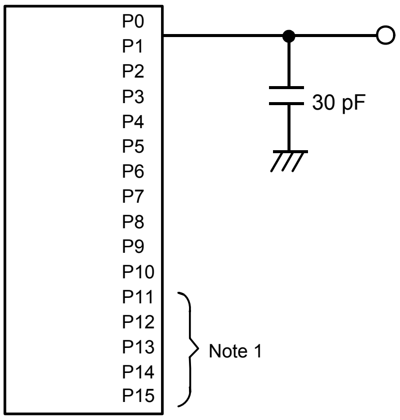
3. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$td(AD\text{-}ALE) = \frac{10^9 \times n}{f(BCLK) \times 2} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ n = a)$$

4. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$th(ALE\text{-}AD) = \frac{10^9 \times n}{f(BCLK) \times 2} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi,\ n = a)$$

5. tc [ns] is added when recovery cycle is inserted.



NOTE:
1. P11 to P15 are provided in the 144-pin package only.

Figure 27.2 P0 to P15 Measurement Circuit

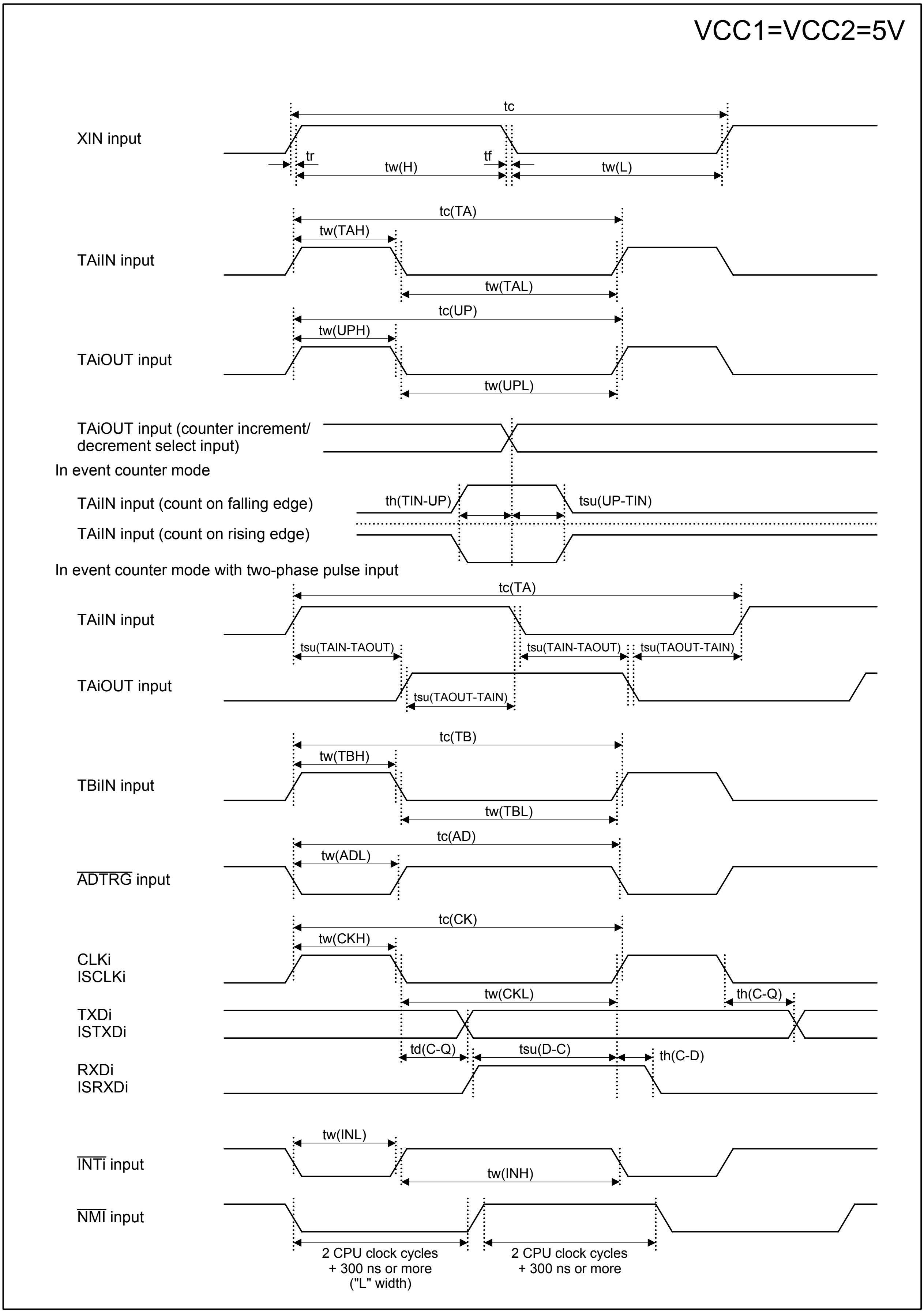
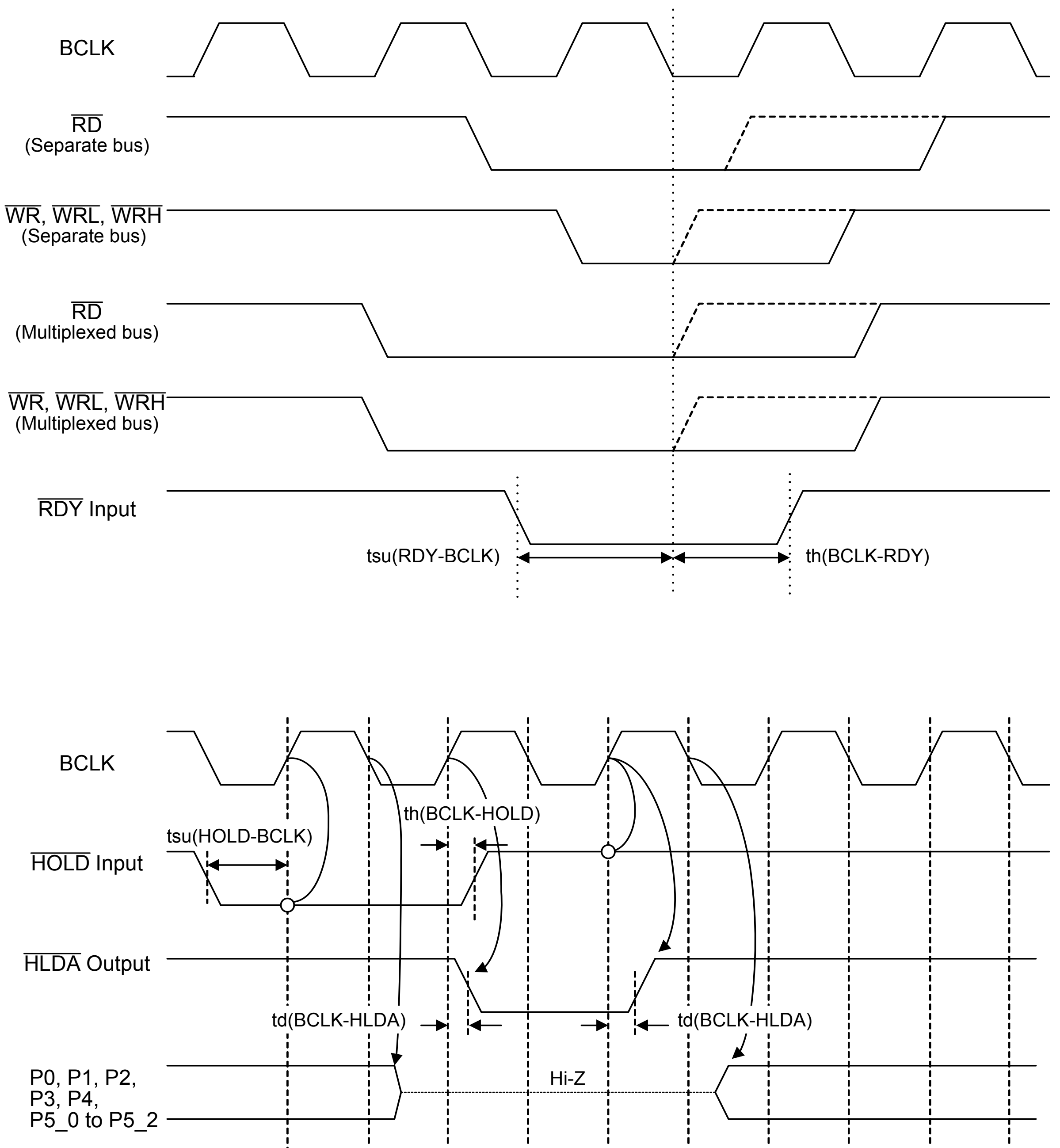


Figure 27.3 VCC1 = VCC2 = 5 V Timing Diagram (1/4)

VCC1=VCC2=5V

Memory Expansion Mode and Microprocessor Mode



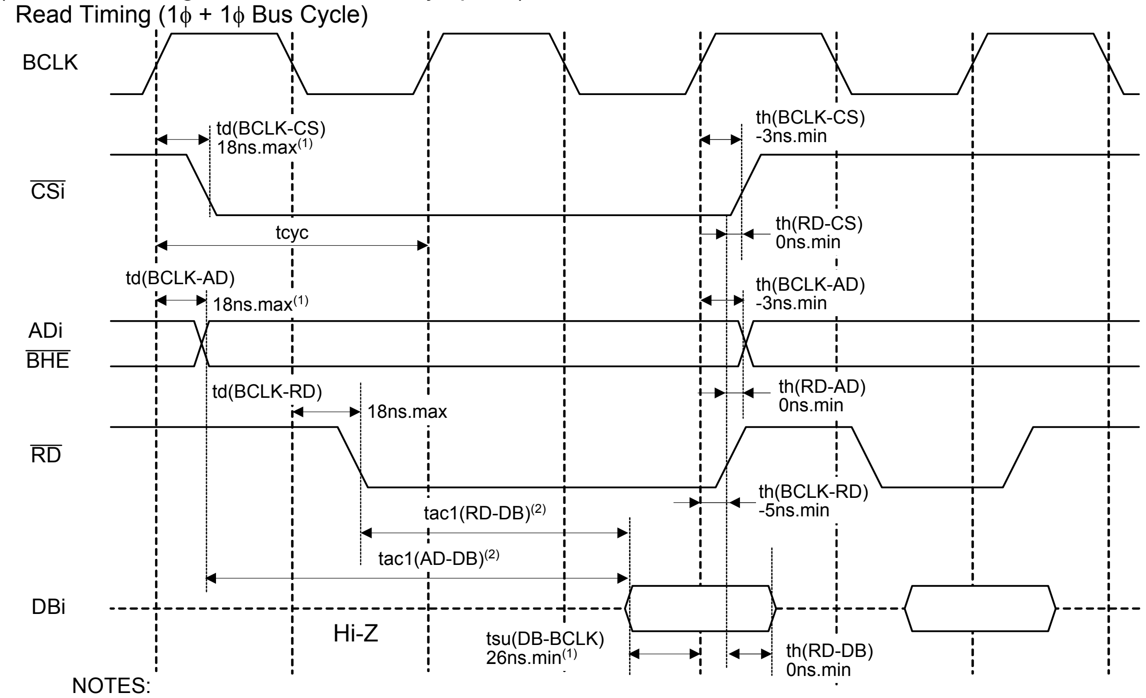
Measurement Conditions
- VCC1 = VCC2 = 4.2 to 5.5 V
- Input high and low voltage: VIH = 4.0 V, VIL = 1.0 V
- Output high and low voltage: VOH = 2.5 V, VOL = 2.5 V

Figure 27.4 VCC1 = VCC2 = 5 V Timing Diagram (2/4)

Memory Expansion Mode and Microprocessor Mode
(when accessing an external memory space)

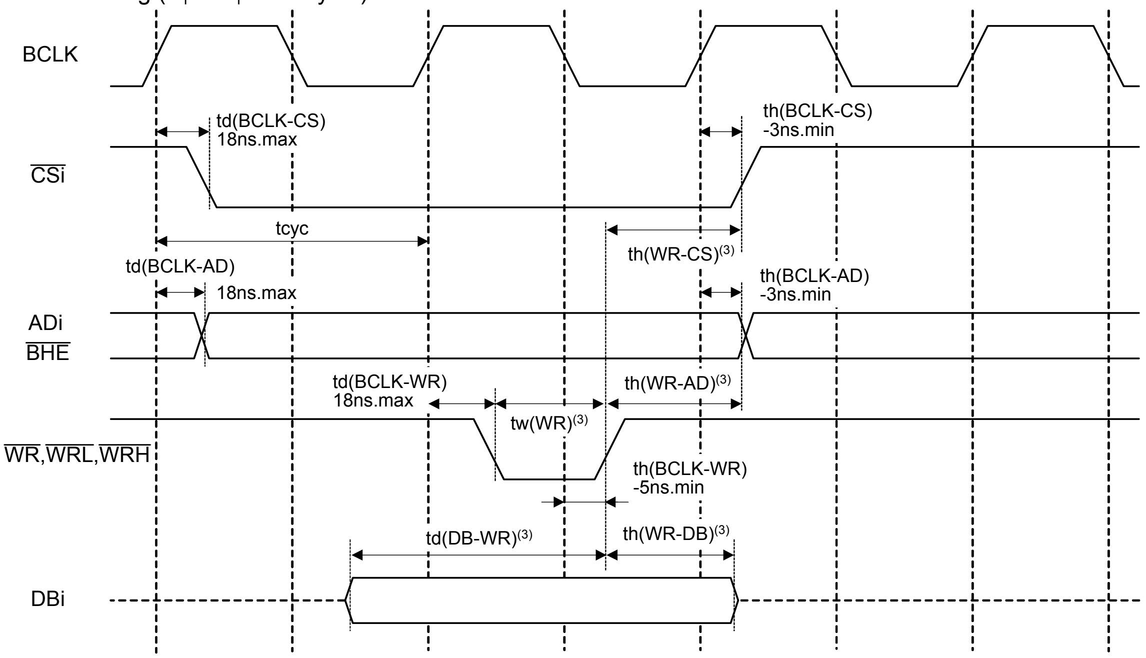
VCC1=VCC2=5V

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 = 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

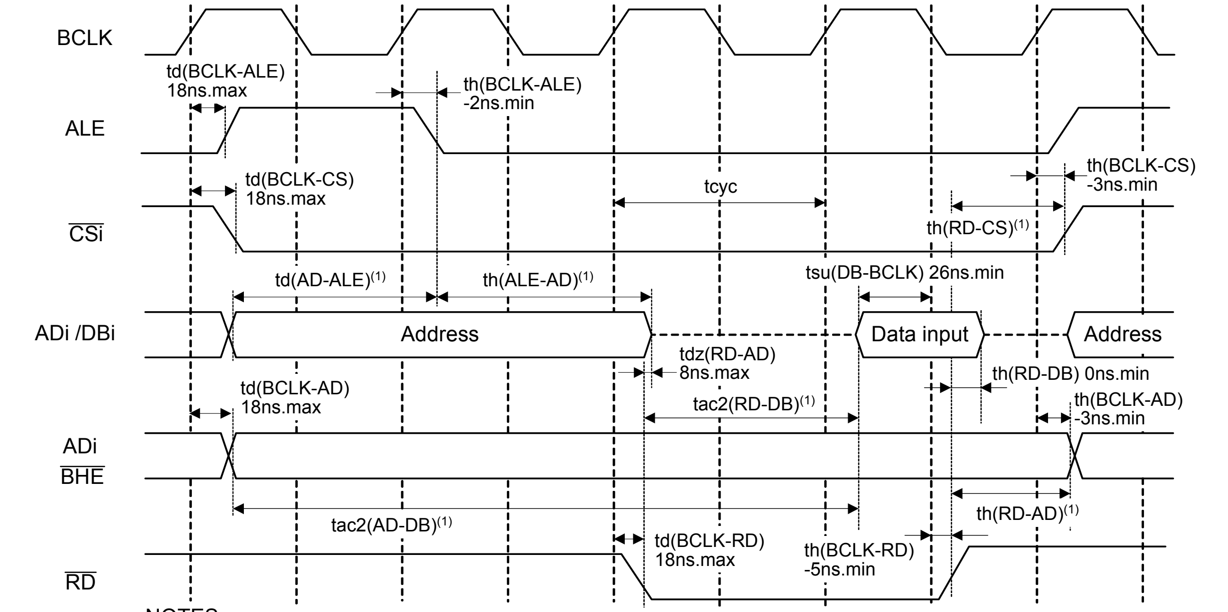
$$tcyc = \frac{10^9}{f(BCLK)}$$

Figure 27.5 VCC1 = VCC2 = 5 V Timing Diagram (3/4)

Memory Expansion Mode and Microprocessor Mode
(when accessing an external memory space with the multiplexed bus)

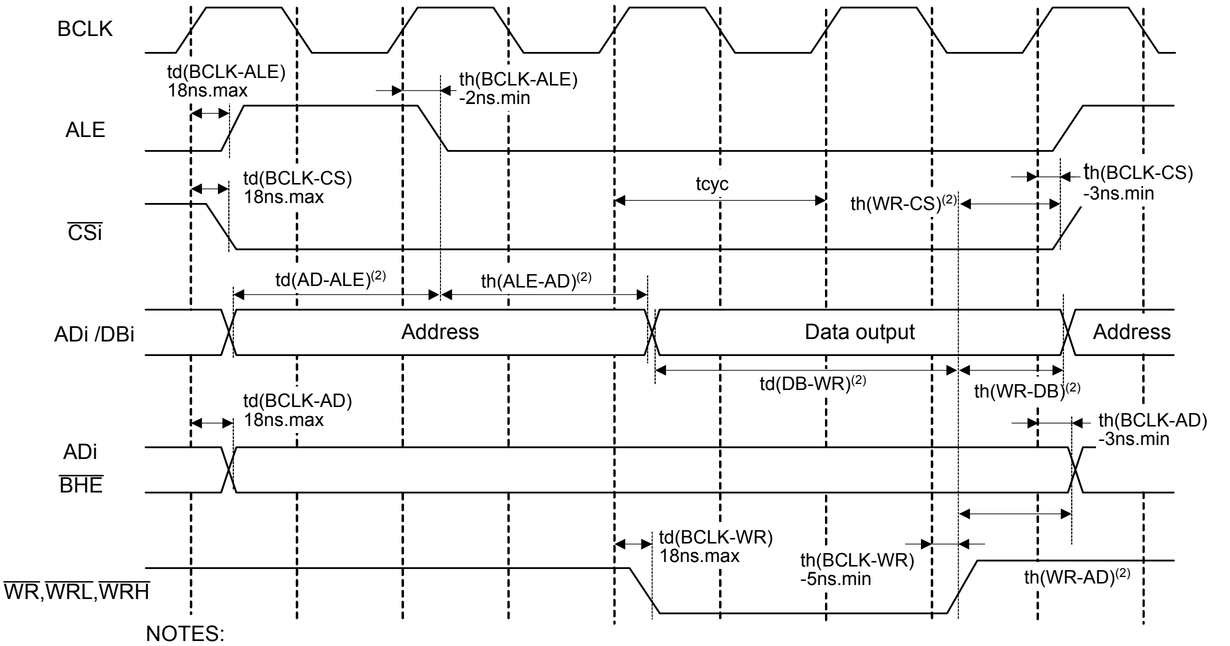
VCC1=VCC2=5V

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 - 20) 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 - 20) 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)

$$tcyc = \frac{10^9}{f(BCLK)}$$

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 27.6 VCC1 = VCC2 = 5 V Timing Diagram (4/4)

VCC1 = VCC2 = 3.3 V

Table 27.31 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			Measurement Condition	Standard Min.	Standard Typ.	Standard Max.	Unit
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		1.6		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	HOLD, RDY, TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT8, ADTRG, CTS0 to CTS6, CLK0 to CLK6, TA0OUT to TA4OUT, NMI, KI0 to KI3, RXD0 to RXD6, SCL0 to SCL4, SDA0 to SDA4, INPC1_0 to INPC1_7, ISCLK0 to ISCLK2, ISRXD0 to ISRXD2, IEIN, CAN0IN, CAN1IN, CAN1WU			0.2		1.0	V
		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 27.32 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		Measurement 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, $\overline{RESET}$, 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, $\overline{RESET}$, 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	40	90	500	kΩ
RfXIN	Feedback resistance	XIN			3.0		MΩ
RfXCIN	Feedback resistance	XCIN			20.0		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 27.33 Electrical Characteristics (3/3)
(VCC1 = VCC2 = 3.3 V, VSS = 0 V, Topr = 25°C)

Symbol	Parameter	Measurement Condition[1]		Standard			Unit
				Min.	Typ.	Max.	
ICC	Power supply current	Flash memory version	f(CPU) = 24 MHz		23	33	mA
			f(CPU) = 16 MHz		17		mA
			f(CPU) = 8 MHz		11		mA
			f(CPU) = f(Ring) In on-chip oscillator low-power consumption mode		2.6		mA
			f(CPU) = 32 kHz In low-power consumption mode While flash memory is operating		430		µA
			f(CPU) = 32 kHz In low-power consumption mode While flash memory is stopped[2]		30		µA
			Wait mode: f(CPU) = f(Ring) After entering wait mode from on-chip oscillator low-power consumption mode		45		µA
			Stop mode (while clock is stopped)		0.8	5	µA
			Stop mode (while clock is stopped) Topr = 85°C			50	µA
		Mask ROM version	f(CPU) = 24 MHz		23	33	mA
			f(CPU) = 16 MHz		17		mA
			f(CPU) = 8 MHz		11		mA
			f(CPU) = f(Ring) In on-chip oscillator low-power consumption mode		1		mA
			f(CPU) = 32 kHz In low-power consumption mode		30		µA
			Wait mode: f(CPU) = f(Ring) After entering wait mode from on-chip oscillator low-power consumption mode		45		µA
			Stop mode (while clock is stopped)		0.8	5	µA
			Stop mode (while clock is stopped) Topr = 85°C			50	µA

NOTES:
1. In single-chip mode, leave the output pins open and connect the input pins to VSS.
2. Value is obtained when setting the FMSTP bit in the FMR0 register to 1 (flash memory stopped) and running the program on RAM.

VCC1 = VCC2 = 3.3 V

Table 27.34 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	8		40	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. Sample and hold function is not available.

Table 27.35 D/A Conversion Characteristics
(VCC1 = VCC2 = 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				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 27.36 External Clock Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc	External clock input cycle time	41		ns
tw(H)	External clock input high (“H”) pulse width	18		ns
tw(L)	External clock input low (“L”) pulse width	18		ns
tr	External clock rise time		5	ns
tf	External clock fall time		5	ns

Table 27.37 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 27.38 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 27.39 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 27.40 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 27.41 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 27.42 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 27.43 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 27.44 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 27.45 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 27.46 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

Table 27.47 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	70		ns
th(C-D)	RXDi input hold time	90		ns

i = 0 to 6

Table 27.48 Intelligent I/O Communication Function (Groups 0 and 1)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	ISCLKi input cycle time	600		ns
tw(CKH)	ISCLKi input high (“H”) pulse width	300		ns
tw(CKL)	ISCLKi input low (“L”) pulse width	300		ns
td(C-Q)	ISTXDi output delay time		100	ns
th(C-Q)	ISTXDi output hold time	0		ns
tsu(D-C)	ISRXDi input setup time	100		ns
th(C-D)	ISRXDi input hold time	100		ns

i = 0, 1

Table 27.49 Intelligent I/O Communication Function (Group 2)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(CK)	ISCLK2 input cycle time	600		ns
tw(CKH)	ISCLK2 input high (“H”) pulse width	300		ns
tw(CKL)	ISCLK2 input low (“L”) pulse width	300		ns
td(C-Q)	ISTXD2 output delay time		180	ns
th(C-Q)	ISTXD2 output hold time	0		ns
tsu(D-C)	ISRXD2 input setup time	150		ns
th(C-D)	ISRXD2 input hold time	100		ns

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 27.50 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 8[1]

NOTE:
1. $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ are provided in the 144-pin package only.

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 27.51 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	30		ns
tsu(RDY-BCLK)	$\overline{\text{RDY}}$ input setup time	40		ns
tsu(HOLD-BCLK)	$\overline{\text{HOLD}}$ input setup time	60		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.

$$tac1(RD\text{-}DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } m = (b \times 2) + 1)$$

$$tac1(AD\text{-}DB) = \frac{10^9 \times n}{f(BCLK)} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } n = a + b)$$

$$tac2(RD\text{-}DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } m = (b \times 2) - 1)$$

$$tac2(AD\text{-}DB) = \frac{10^9 \times p}{f(BCLK) \times 2} - 35 \text{ [ns] (if external bus cycle is } a\phi + b\phi\text{, } p = \{(a + b - 1) \times 2\} + 1)$$

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 27.52 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 27.2		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		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.

$$th(WR\text{-}DB) = \frac{10^9}{f(BCLK) \times 2} - 20 \text{ [ns]}$$

$$th(WR\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 15 \text{ [ns]}$$

$$th(WR\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

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 27.53 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 27.2		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		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.

$$th(RD\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}AD) = \frac{10^9}{f(BCLK) \times 2} - 15 \text{ [ns]}$$

$$th(RD\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \text{ [ns]}$$

$$th(WR\text{-}DB) = \frac{10^9}{f(BCLK) \times 2} - 20 \text{ [ns]}$$

2. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$td(DB\text{-}WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \text{ [ns] (if external bus cycle is } a\phi + b\phi, m = (b \times 2) - 1)$$

3. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$td(AD\text{-}ALE) = \frac{10^9 \times n}{f(BCLK) \times 2} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi, n = a)$$

4. Values, which depend on BCLK frequency and external bus cycles, can be obtained from the following equation.

$$th(ALE\text{-}AD) = \frac{10^9 \times n}{f(BCLK) \times 2} - 20 \text{ [ns] (if external bus cycle is } a\phi + b\phi, n = a)$$

5. tc [ns] is added when recovery cycle is inserted.

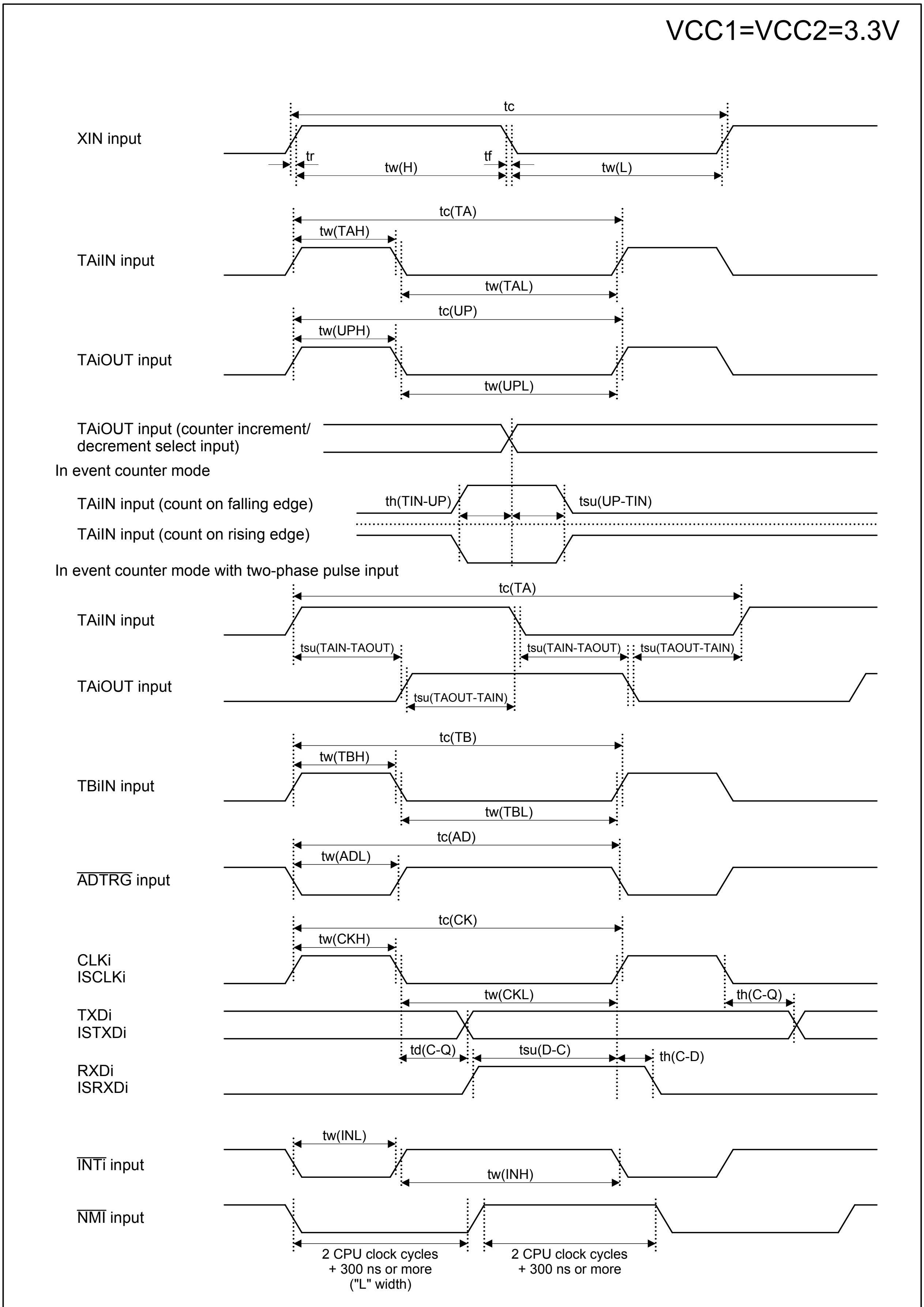


Figure 27.7 VCC1 = VCC2 = 3.3 V Timing Diagram (1/4)

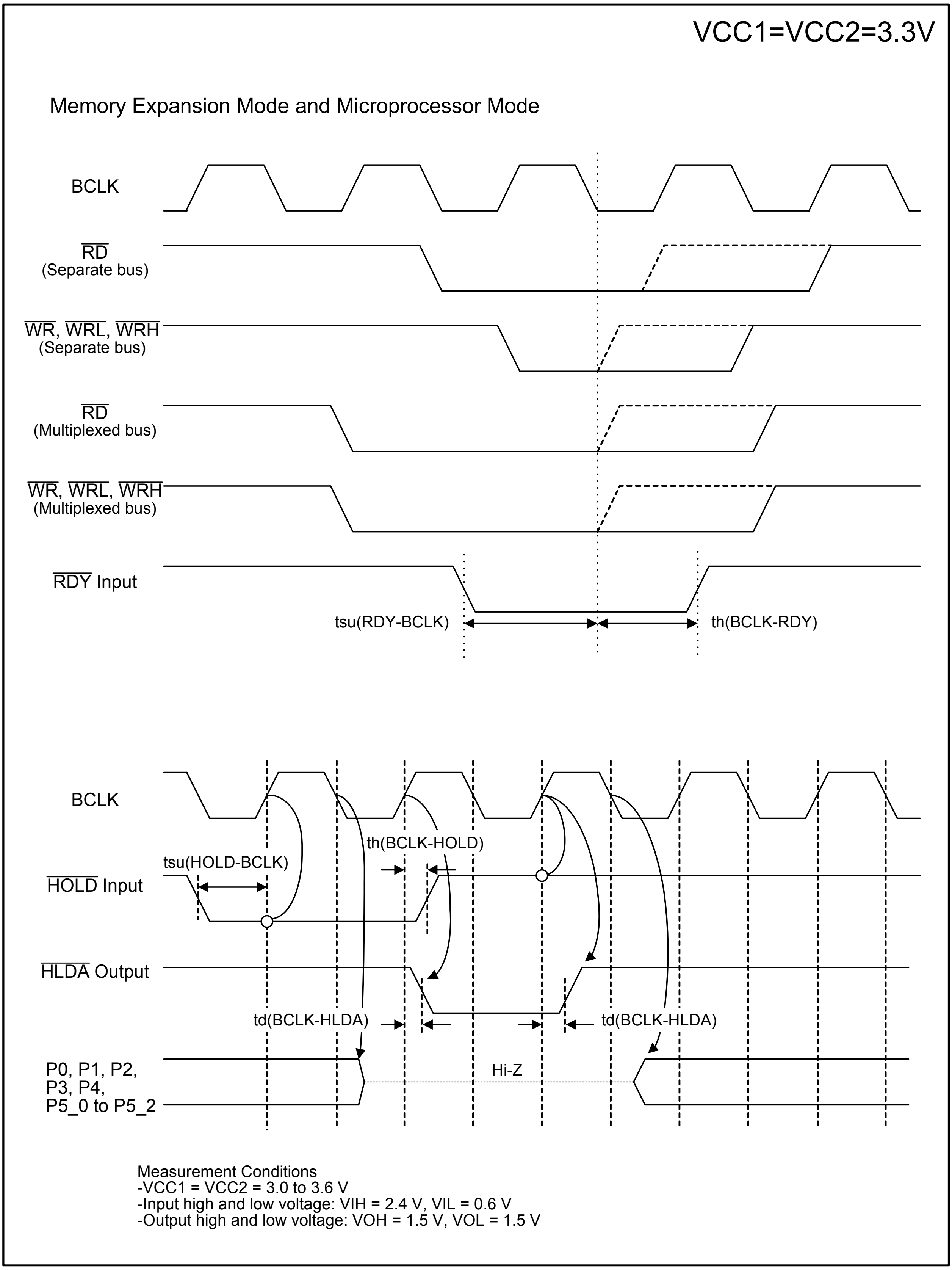
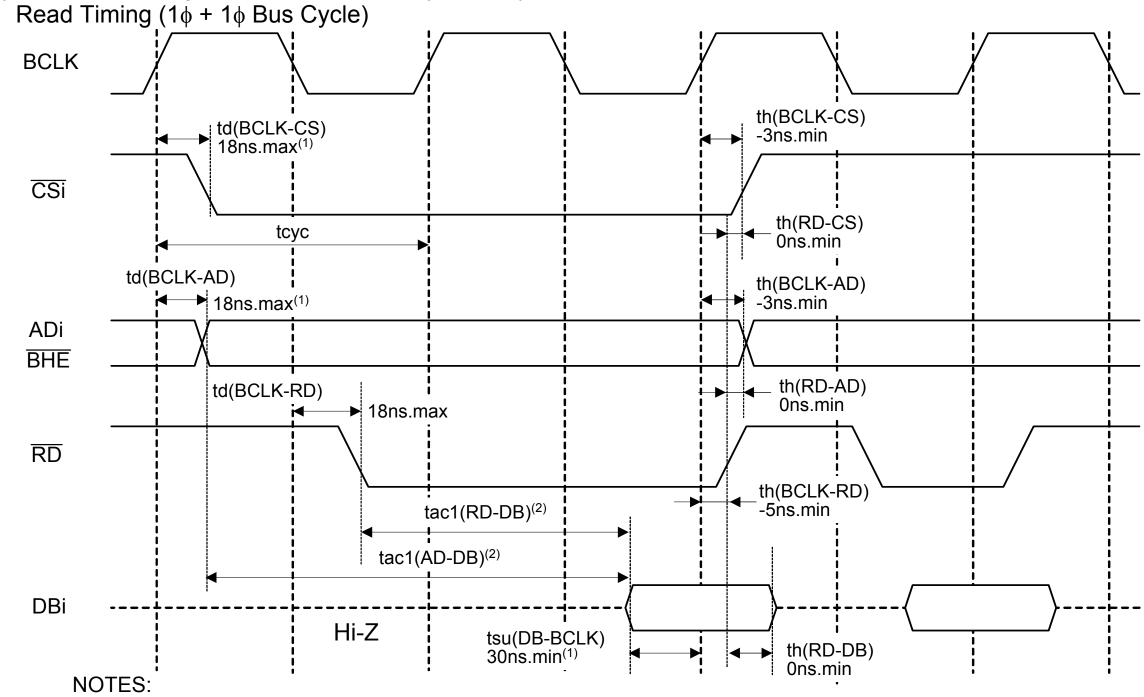


Figure 27.8 VCC1 = VCC2 = 3.3 V Timing Diagram (2/4)

Memory Expansion Mode and Microprocessor Mode
(when accessing an external memory space)

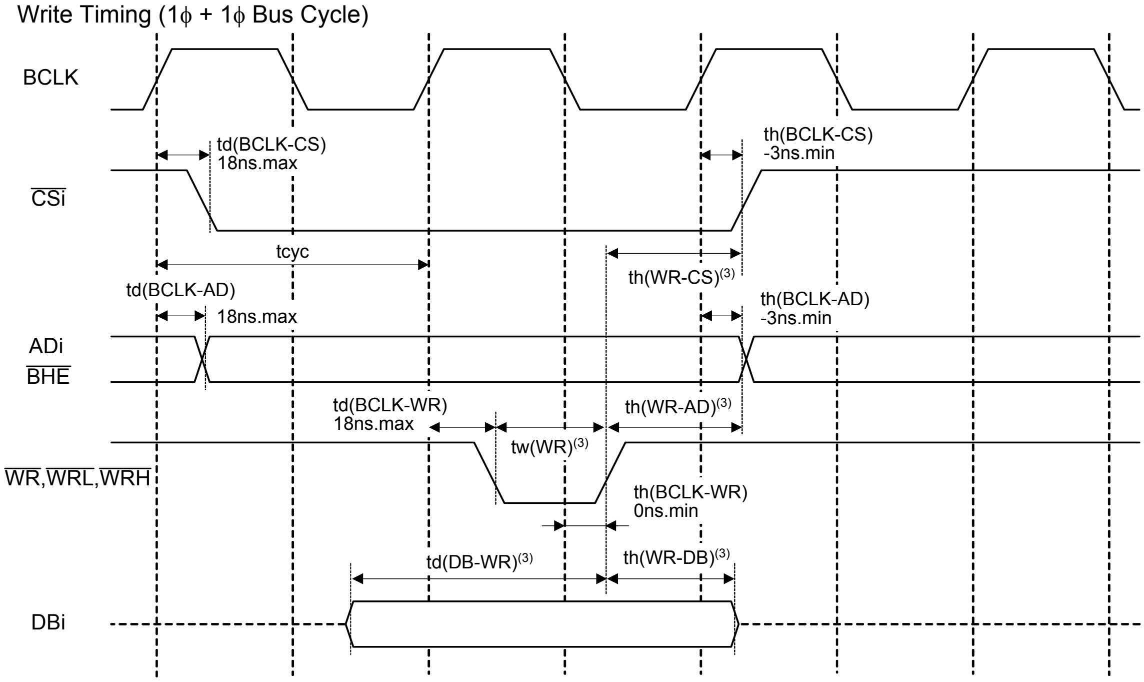
VCC1=VCC2=3.3V

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 - 20) ns.min
   th(WR-AD) = (tcyc / 2 - 15) 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

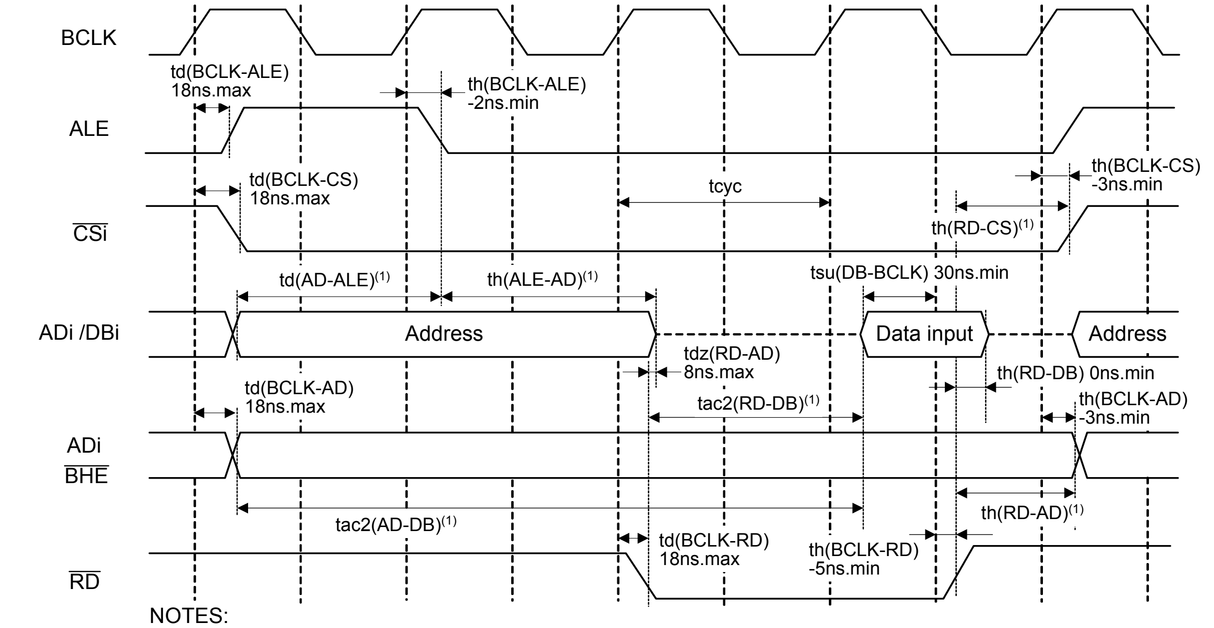
$$tcyc = \frac{10^9}{f(BCLK)}$$

Figure 27.9 VCC1 = VCC2 = 3.3 V Timing Diagram (3/4)

Memory Expansion Mode and Microprocessor Mode
(when accessing an external memory space with the multiplexed bus)

VCC1=VCC2=3.3V

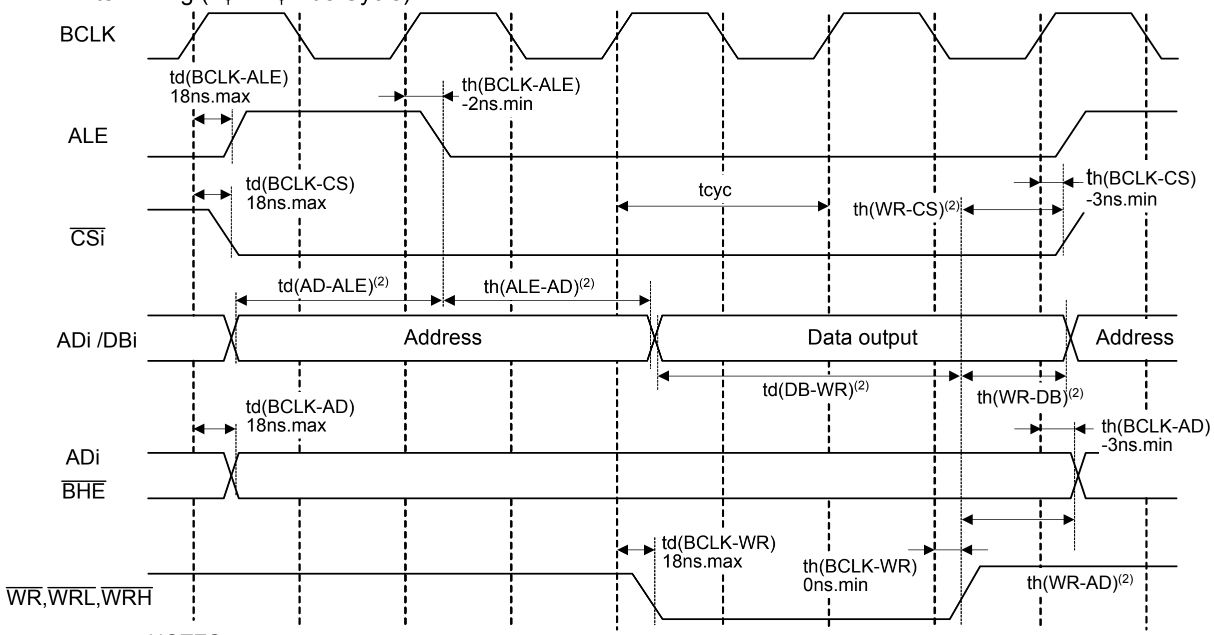
Read Timing (2ϕ + 2ϕ Bus Cycle)



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 - 20) 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 - 20) ns.min (if external bus cycle aϕ + bϕ, n = a)
   th(WR-AD) = (tcyc / 2 - 15) ns.min, th(WR-CS) = (tcyc / 2 - 10) ns.min
   th(WR-DB) = (tcyc / 2 - 20) 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 = 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 27.10 VCC1 = VCC2 = 3.3 V Timing Diagram (4/4)

28. Usage Notes

28.1 Power Supply

28.1.1 Power-on

At power-on, supply voltage applied to the VCC1 must meet the SVCC standard.
(Technical update: TN-M16C-116-0311)

Table 28.1 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

SVCC

Supply voltage power-up slope (VCC1)

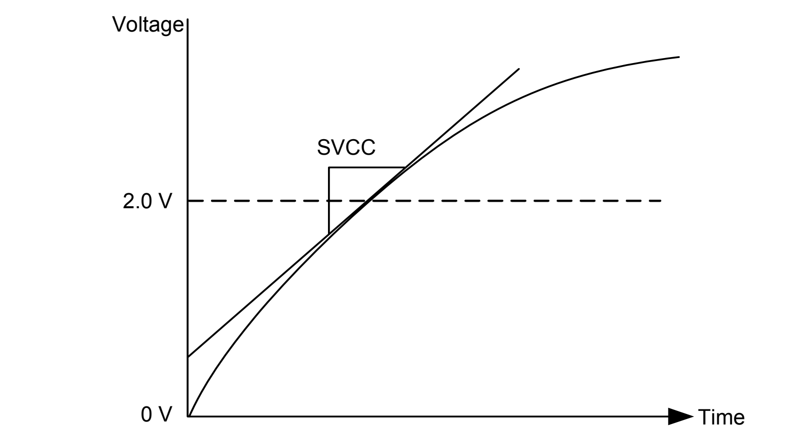


Figure 28.1 SVCC Timing

28.1.2 Power Supply Ripple

Stabilize supply voltage to meet the power supply standard listed in Table 28.2.

Table 28.2 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

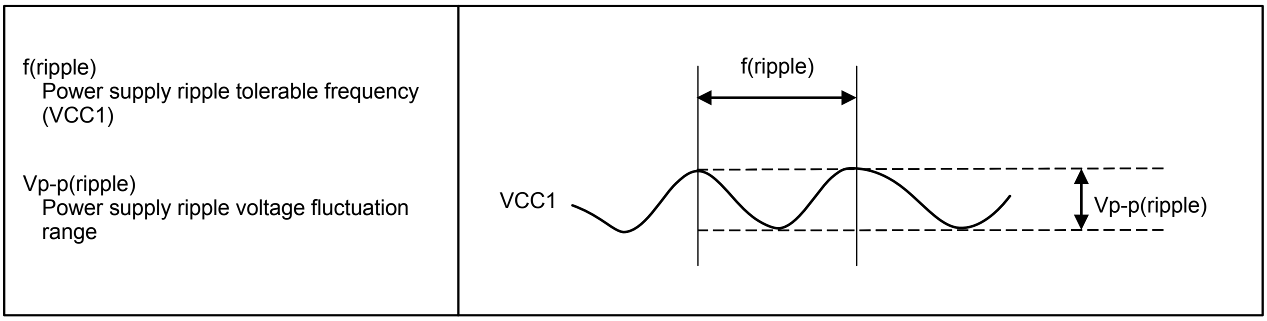


Figure 28.2 Power Supply Fluctuation Timing

28.1.3 Noise

Use thick and shortest possible wiring to connect a bypass capacitor (0.1 µF or more) between VCC and VSS.

28.2 Special Function Registers (SFRs)

28.2.1 100 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.

28.2.2 Register Settings

Table 28.3 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 28.4 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 28.3 Registers with Write-Only Bits

Register	Address	Register	Address
WDTS register	000Eh	TA41 register	0307h to 0306h
G0TB register	00EAh	DTT register	030Ch
G0RI register	00ECh	ICTB2 register	030Dh
G1TB register	012Ah	U3BRG register	0329h
G1RI register	012Ch	U3TB register	032Bh to 032Ah
G2TB register	016Dh to 016Ch	U2BRG register	0339h
U5BRG register	01C1h	U2TB register	033Bh to 033Ah
U5TB register	01C3h to 01C2h	UDF register	0344h
U6BRG register	01C9h	TA0 register(1)	0347h to 0346h
U6TB register	01CBh to 01CAh	TA1 register(1)	0349h to 0348h
U1BRG register	02E9h	TA2 register(1)	034Bh to 034Ah
U1TB register	02EBh to 02EAh	TA3 register(1)	034Dh to 034Ch
U4BRG register	02F9h	TA4 register(1)	034Fh to 034Eh
U4TB register	02FBh to 02FAh	U0BRG register	0369h
TA11 register	0303h to 0302h	U0TB register	036Bh to 036Ah
TA21 register	0305h to 0304h		

NOTE:
1. In one-shot timer mode and pulse width modulation mode only.

Table 28.4 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

28.3 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.

28.4 Bus

28.4.1 $\overline{\text{HOLD}}$ 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)

28.5 Clock Generation Circuits

28.5.1 Main Clock

- When the CPU operating frequency is required 24 MHz or higher, make an oscillator connected to the main clock circuit (XIN-XOUT), or a clock applied to the XIN pin have 24 MHz or lower frequency, and then multiply the main clock with the PLL frequency synthesizer. By using this procedure, a better EMC (Electromagnetic Compatibility) performance can be achieved than connecting a 24 MHz or higher frequency oscillator or using 24 MHz or higher input clock applied to the XIN pin.

- 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).

28.5.2 Sub Clock

28.5.2.1 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)

28.5.2.2 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.

28.5.3 Clock Dividing Ratio

To change bits MCD4 to MCD0, set the PM12 bit in the PM1 register to 0 (no wait state).

28.5.4 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.

28.5.4.1 Wait Mode

- When entering wait mode with setting the CM02 bit in the CM0 register to 1 (peripheral clocks stop in wait mode), set bits MCD4 to MCD0 in the MCD register for CPU clock frequency to be 10-MHz or lower after dividing the main clock.

- 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.

28.5.4.2 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
              .
              .
              .
```

28.5.4.3 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 registers DACON1 and DAi 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, do not set the CM02 bit to 1 (peripheral clock stops in wait mode) when entering wait mode.
  (Technical update: TN-M16C-69-0104)

28.6 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.

28.7 Interrupts

28.7.1 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 $\overline{\text{NMI}}$ interrupt, set ISP at the very beginning of the program. The $\overline{\text{NMI}}$ interrupt can be acknowledged after the first instruction has been executed after reset.

28.7.2 $\overline{\text{NMI}}$ Interrupt

- The $\overline{\text{NMI}}$ interrupt cannot be disabled. Connect the $\overline{\text{NMI}}$ 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 $\overline{\text{NMI}}$ pin. Read the P8_5 bit only to determine the pin level after the $\overline{\text{NMI}}$ interrupt occurs.

28.7.3 $\overline{\text{INT}}$ Interrupt

- Edge Sensitive
  Each “H” or “L” width of the signal applied to pins $\overline{\text{INT0}}$ to $\overline{\text{INT8}}$ 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 $\overline{\text{INT0}}$ to $\overline{\text{INT5}}$ 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 $\overline{\text{INT0}}$ to $\overline{\text{INT5}}$ are changed. Set the IR bit to 0 (interrupt not requested) after the polarity setting is changed.

Figure 28.3 shows a procedure to set the $\overline{\text{INTi}}$ interrupt source (i = 0 to 5).

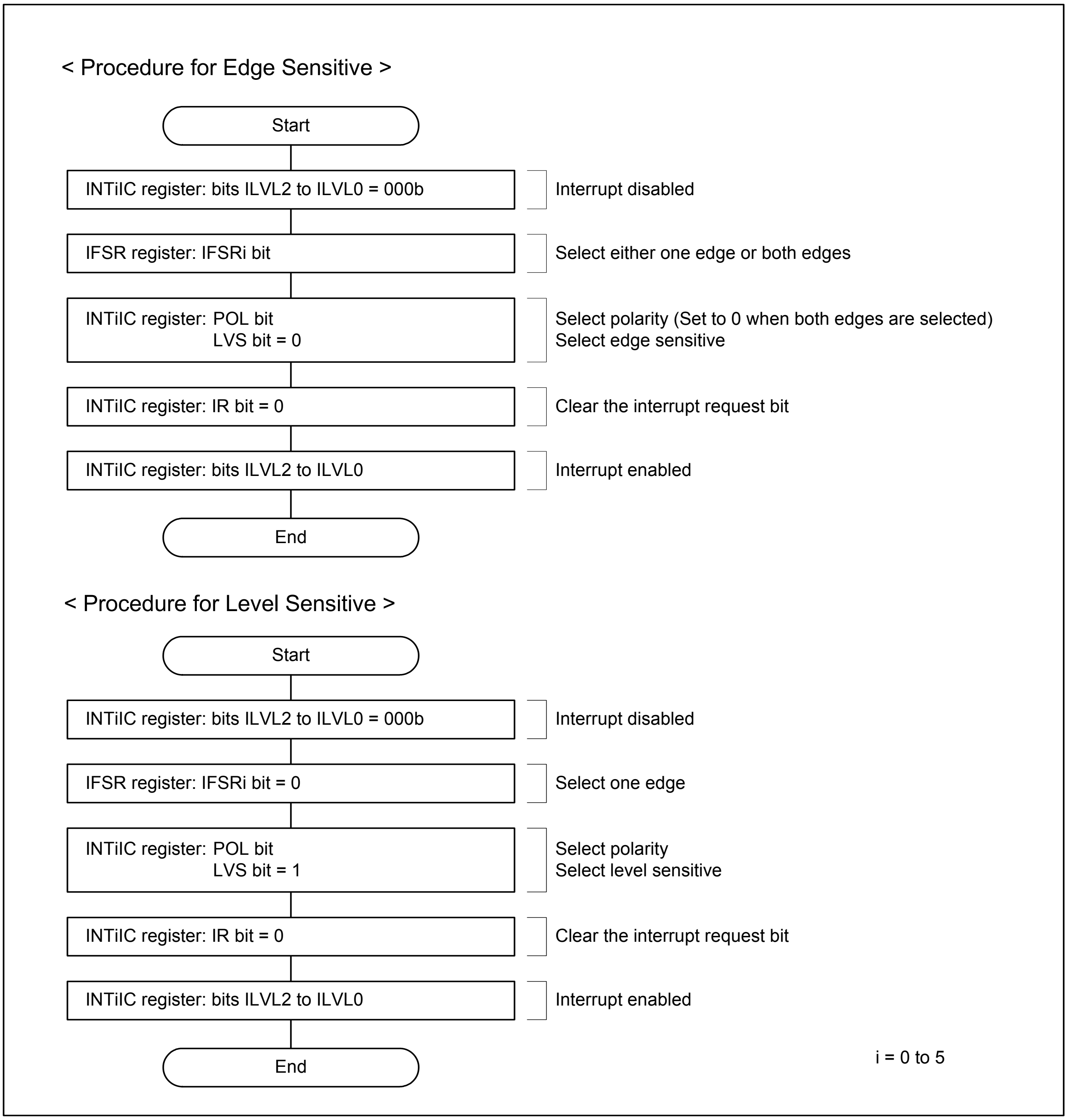


Figure 28.3 Procedure to set the $\overline{\text{INTi}}$ Interrupt Source (i = 0 to 5)

- The INTiR bit (i = 6 to 8) in the IIOjIR register (j = 9 to 11) may become 1 (interrupt requested) when the polarity settings of pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ are changed. Set the INTiR bit to 0 (interrupt not requested) after the polarity setting is changed.

Figure 28.4 shows an example of the switching procedure for an $\overline{\text{INTi}}$ interrupt source. (i = 6 to 8)

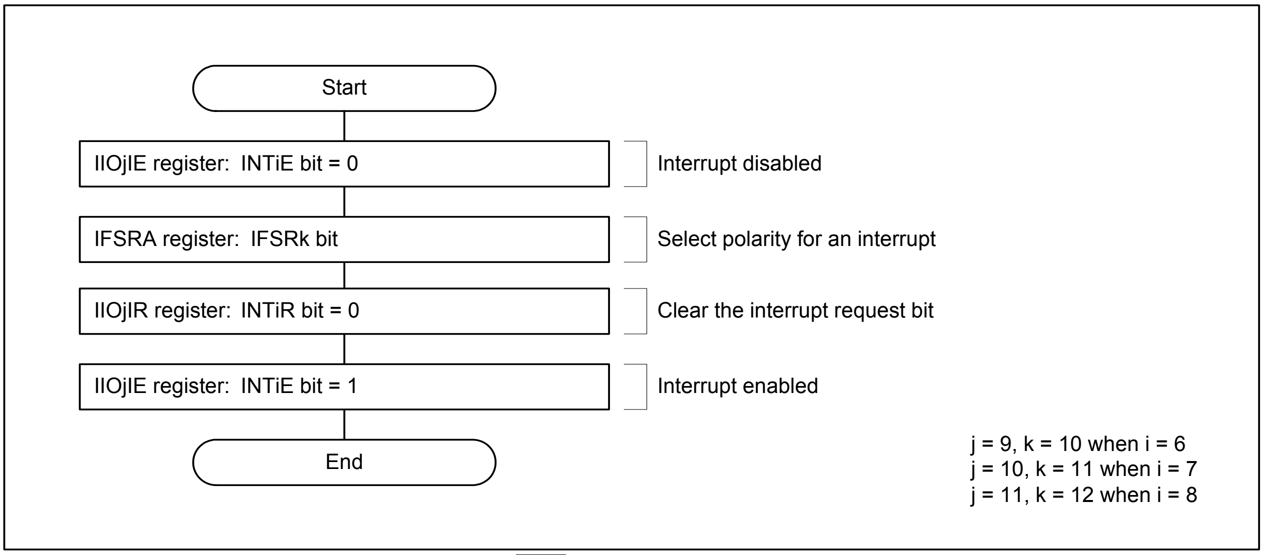


Figure 28.4 Switching Procedure for $\overline{\text{INTi}}$ (i = 6 to 8) Interrupt Source

28.7.4 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

28.7.5 Changing IIOiIR Register (i = 0 to 11)

- Interrupt request flag
  An interrupt request flag becomes 1 (interrupt requested) when an interrupt request is generated. This flag does not automatically become 0 when the interrupt request is acknowledged. Use AND or BCLR instruction to set it to 0 (interrupt not requested) in the interrupt routine. If any of these flags remains 1, the IR bit in the IIOiIC (CANjIC (j = 0 to 5)) register does not become 1 when an interrupt request is generated in the same register. (Interrupt does not occur.)
  If an interrupt request is generated while writing a 0 to the corresponding interrupt request flag, the flag may not be cleared to 0. In this case, keep writing a 0 until 0 is read.

28.7.6 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.

28.8 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).

28.9 Timers

28.9.1 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

28.9.2 Timer A

28.9.2.1 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.

28.9.2.2 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.

28.9.2.3 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)

28.9.2.4 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.

28.9.3 Timer B

28.9.3.1 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.

28.9.3.2 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.

28.10 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)

28.11 Serial Interfaces

28.11.1 Changing UiBRG Register (i = 0 to 6)

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.

28.11.2 Clock Synchronous Mode

28.11.2.1 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 6) 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.

28.11.2.2 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 SmRIC register (m = 0 to 4), the U5RR in the IIO0IR register, or U6RR bit in the IIO9IR register is not changed to 1.

- 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.

28.11.3 UART Mode

Set the UmERE bit in the UmC1 register after setting the UmMR register.

28.11.4 Special Mode 1 (I²C Mode)

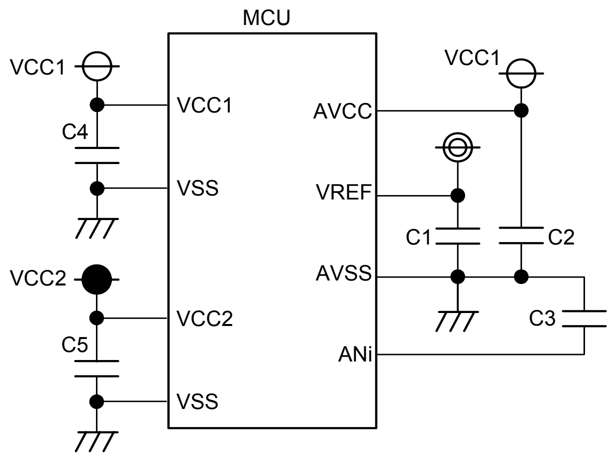
To generate the start condition, stop condition, or restart condition, set the STSPSEL bit in the UmSMR4 register to 0. Then, wait for a half clock cycle of the serial clock or more to change individual condition generation bit (the STAREQ bit, STPREQ bit, or RSTAREQ bit) from 0 to 1.
(Technical update: TN-16C-130A/EA)

28.12 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 28.5 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 28.5 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 AD0i 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 AD0i registers.

- External triggers cannot be used in DMAC operating mode. 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)

28.13 Intelligent I/O

28.13.1 Register Setting

- Each value written to the following registers is reflected in synchronization with the count source (fBTi) (i = 1, 2) set using bits BCK1 and BCK0 in the GiBR0 register. Set bits BCK1 and BCK0 before setting these registers.

  Group 1:
  G1BT, G1BCR1, G1TMCR0 to G1TMCR7, G1TPR6, G1TPR7, G1TM0 to G1TM7, G1POCR0 to G1POCR7, G1PO0 to G1PO7, G1FS, G1FE
  Group 2:
  G2BT, G2BCR1, G2POCR0 to G2POCR7, G2PO0 to G2PO7, G2FS, G2RTP, BTSR

- When interrupts are used in time measurement function and waveform generation function, use the following procedure. (Refer to a flowchart of register settings for each function.)
  (1) Configure for time measurement function or waveform generation function
  (2) Set the IFEj bit (j = 0 to 7) in the GiFE register to 1
  (3) Wait 2 fBTi clock cycles or more
  (4) Set for the intelligent I/O interrupt

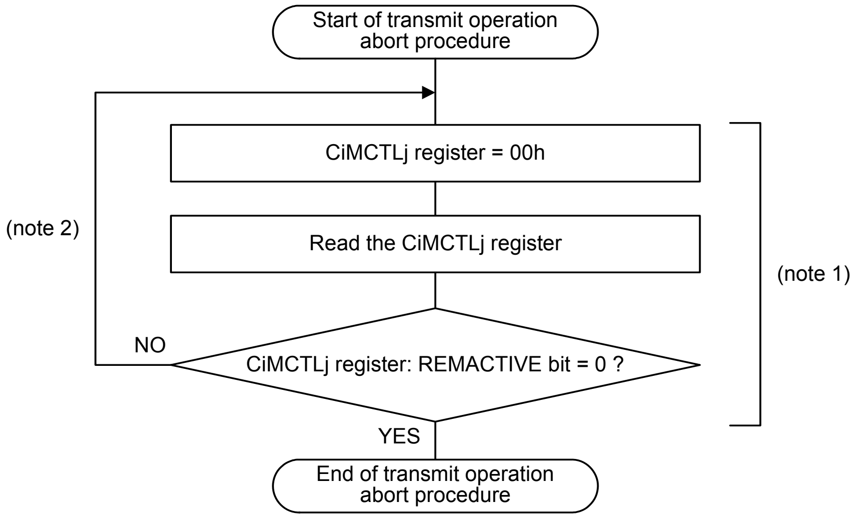
- Each value written to the following registers is reflected in synchronization with the serial clock. Wait for one clock cycle of the serial clock or more after selecting the serial clock, and then set these registers.

  Group 0 and 1:
  GmMR (m = 0, 1), GmCR, GmEMR, GmETC, GmERC, GmIRF, GmTB (GmDR), GmCMP0 to GmCMP3, GmMSK0 to GmMSK1, GmTCRC, GmRCRC, GmRB, GmRI, GmTO
  Group 2:
  G2TB, G2RB, G2MR, G2CR, IECR, IEAR, IETIF, IERIF

- If the IVL or INV bit in the GiPOCRj register is written while outputting waveform, the value written takes effect immediately on the output waveform.

28.14 CAN

Use the following procedures to abort a remote frame transmit operation or to cancel a remote frame receive operation. (Technical update: TN-16C-126A/EA)

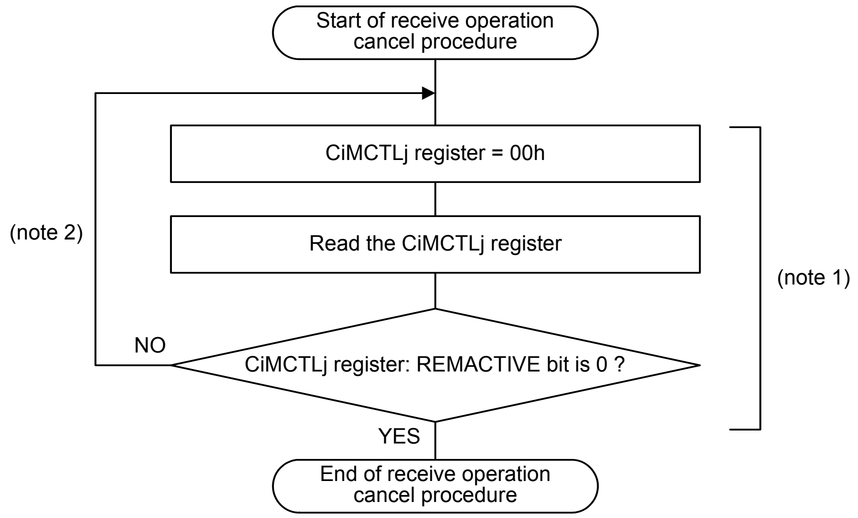


i = 0, 1, j = 0 to 15

NOTES:

1. The CiMCTLj register can be accessed when the BANKSEL bit the CiCTLR1 register is set to 0 (registers CiMCTLj, CiSSCTLR, and CiSSSTR are selected).
2. There is an infinite loop set in the flowchart to simplify the description. Set the time limit to break out of the infinite loop for practical use.

Figure 28.6 Procedure to Abort a Remote Frame Transmit Operation



i = 0, 1, j = 0 to 15

NOTES:

1. The CiMCTLj register can be accessed when the BANKSEL bit the CiCTLR1 register is set to 0 (registers CiMCTLj, CiSSCTLR, and CiSSSTR are selected).
2. There is an infinite loop set in the flowchart to simplify the description. Set the time limit to break out of each infinite loop for practical use.

Figure 28.7 Procedure to Cancel a Remote Frame Receive Operation

28.15 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, intelligent I/O output, RTP output), they are affected by the three-phase motor control timer function and the $\overline{\text{NMI}}$ pin setting. Table 28.5 shows the INVC0 register setting, $\overline{\text{NMI}}$ pin input level, and output pin states.

Table 28.5 INVC0 Register Setting, $\overline{\text{NMI}}$ Pin Level, and Output Pin Status

<table>
<tr><th colspan="2">Setting Value of the INVC0 Register</th><th rowspan="2">$\overline{\text{NMI}}$ Pin Input Level</th><th rowspan="2">Pin States of P7_2 to P7_5, P8_0, P8_1 (when set in output mode)</th></tr>
<tr><th>INV02 Bit</th><th>INV03 Bit</th></tr>
<tr><td>0
(three-phase motor control timer function not used)</td><td>–</td><td>–</td><td>Output functions selected using registers PS1, PSL1, PSC, PS2, and PSL2</td></tr>
<tr><td rowspan="3">1
(three-phase motor control timer function used)</td><td>0
(three-phase motor control timer output disabled)</td><td>–</td><td>High-impedance states</td></tr>
<tr><td rowspan="2">1
(three-phase motor control timer output enabled)(1)</td><td>H</td><td>Output functions selected using registers PS1, PSL1, PSC, PS2, and PSL2</td></tr>
<tr><td>L
(forcibly terminated)</td><td>High-impedance states</td></tr>
</table>

–: 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)

28.16 Flash Memory

28.16.1 Operating Speed

Prior 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).

28.16.2 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.

28.16.3 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, Vdet4 detection function, or oscillation stop detection function, registers FMR0 and FMR1 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.

28.16.4 Interrupts (EW1 Mode)

- When an interrupt request is generated by the peripheral function or watchdog timer (when the PM22 bit in the PM2 register is set to 0) during the erase or program operation, the interrupt is acknowledged after the erase or program operation is completed.
- When an interrupt request is generated by the $\overline{\text{NMI}}$, watchdog timer (when the PM22 bit is set to 1), Vdet4 detection function, or oscillation stop detection function, registers FMR0 and FMR1 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.

28.16.5 How to Access

To set the FMR01 or FMR02 bit in the FMR0 register, or the FMR11 bit in the FMR1 register to 1, write a 1 immediately after writing a 0 to the bit. Write to the FMR0 or FMR1 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.

28.16.6 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.

28.16.7 Rewriting User ROM Area (EW1 Mode)

Do not rewrite a block where the rewrite control program is stored.

28.16.8 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)

28.16.9 Writing Command and Data

Write command codes and data to even addresses in the user ROM area.

28.16.10 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 of the block which has been erased may become 0 (locked). 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.

28.16.11 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.

28.16.12 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
```

28.16.13 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:

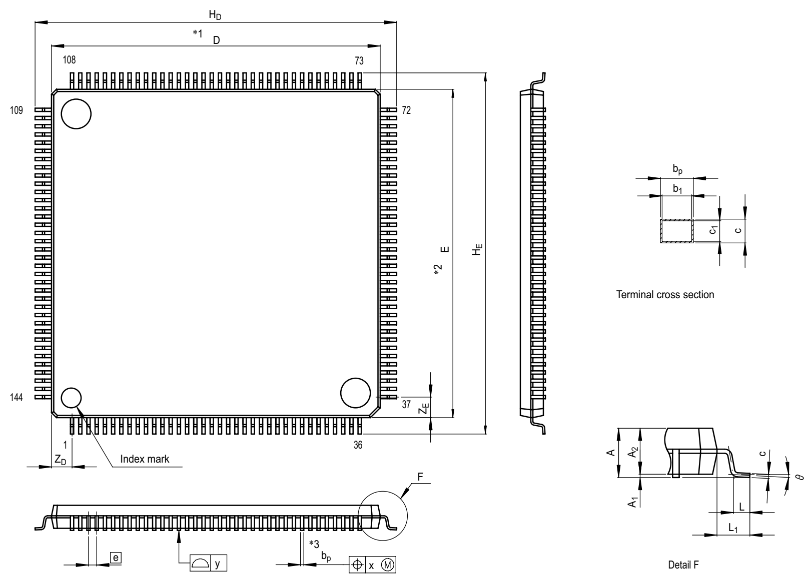
- Program command
- Block erase command
- Lock bit program command
- Read lock bit status command

28.17 Difference Between Flash Memory Version and Mask ROM Version

Due to differences in internal ROM type and the layout pattern, flash memory version and mask ROM version may vary in characteristic values, performance margin, noise endurance, noise radiation, and so on, within a range provided in the chapter, Electrical Characteristics. When switching to mask ROM version, perform system evaluation tests equal to those held on the flash memory version.

Appendix 1. Package Dimensions

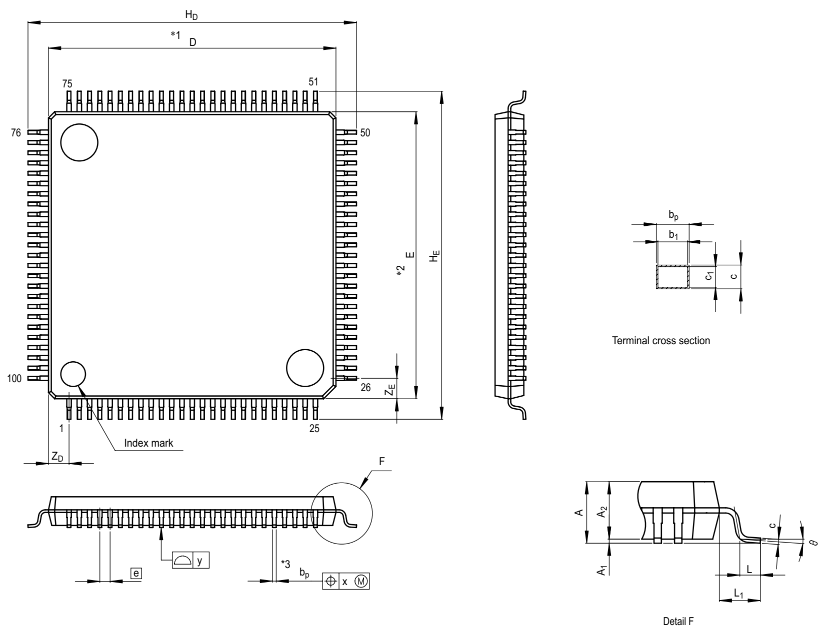
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g



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°
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	—

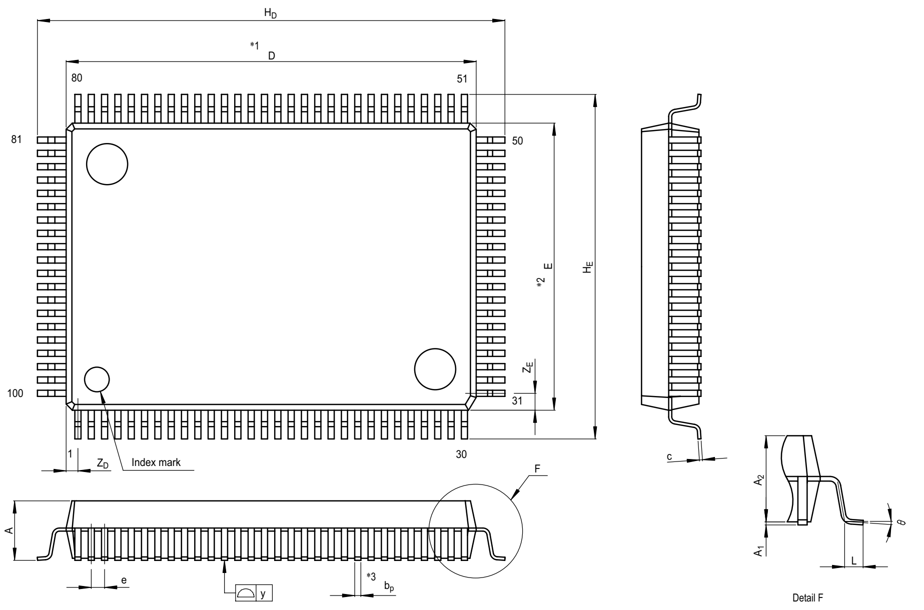
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g



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°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP100-14x20-0.65	PRQP0100JB-A	100P6S-A	1.6g



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	19.8	20.0	20.2
E	13.8	14.0	14.2
A_2	—	2.8	—
H_D	22.5	22.8	23.1
H_E	16.5	16.8	17.1
A	—	—	3.05
A_1	0	0.1	0.2
b_p	0.25	0.3	0.4
c	0.13	0.15	0.2
θ	0°	—	10°
e	0.5	0.65	0.8
y	—	—	0.10
Z_D	—	0.575	—
Z_E	—	0.825	—
L	0.4	0.6	0.8

Index

[U]

[V]

[W]

[X]

[Y]

REVISION HISTORY	M32C/87 Group Hardware Manual

Rev.	Date	Description: Page	Description: Summary
0.20	Dec 16, 2004	–	First Edition issued
1.00	Aug 10, 2005	–	**M32C/87A and M32C/87B** added
		–	Package code changed: 144P6Q-A to PLQP0144KA-A, 100P6Q-A to PLQP0100KB-A, 100P6S-A to PRQP0100JB-A
		–	Function description for the reserved bits on register diagram modified
		–	"Low Voltage Detection Reset" changed to "Brown-out Detection Reset"
			Overview
		2, 3	• **Tables 1.1 and 1.2 M32C/87 Group (M32C/87, M32C/87A, M32C/87B) Performance** Performance for CAN and Electrical Characteristics modified
		4	• **Figure 1.1 M32C/87 Group (M32C/87, M32C/87A, M32C/87B) Block Diagram** Diagram modified, note 5 added
		5	• **Table 1.3 M32C/87 Group** Product information updated
		6	• **Figure 1.2 Product Numbering System** ROM capacity modified
		7	• **Figure 1.3 Pin Assignment for 144-Pin Package** Note 15 added
		8	• **Table 1.4 Pin Characteristics** Note 1 added
		11	• **Figure 1.4 Pin Assignment for 100-Pin Package** Note 19 added
		12	• **Figure 1.5 Pin Assignment for 100-Pin Package** Note 15 added
		13	• **Table 1.5 Pin Characteristics** Note 1 added
		17	• **Table 1.6 Pin Description** Note 2 added
			Memory
		22	• **Figure 3.1 Memory Map** Diagram modified, notes 2 and 3 modified, notes 4 and 5 added
			Special Function Register (SFR)
		23	• The PM0 register Note 1 added
			• The PLC0 register Value after reset modified
		26	• The RLVL and IIO0IR to IIO11IR registers Value after reset modified
		29	• The G1BCR1 and G1RB registers Value after reset modified
		32-37	• Note "The CAN-associated registers in M32C/87B cannot be used. Only CAN0-associated registers in M32C/87A can be used" added
		38	• The IDB1 and IDB0 registers Value after reset modified
		40	• Note 1 added
		41	• The DM0SL to DM3SL and AD00 registers Value after reset modified
		42	• The PSC and PS2 registers Value after reset modified
		43	• The PCR register Value after reset modified
		44	• The PSD1 register Value after reset modified
		45	• The PCR register Value after reset modified
			Reset
		46	• **Section "Voltage Detection Circuit"** deleted to create the new chapter
			Voltage Detection Circuit
		51	• **Section structure and description** modified to create the new chapter
			• **Figure 6.1 Voltage detection Circuit Block Diagram** modified
		52	• **Figure 6.2 WDC Register** Note 3 added
		53	• **Figure 6.3 VCR1 Register** Note 1 deleted
			• **Figure 6.3 VCR2 Register** Note 2 deleted, note 5 added
		55	• **Table 6.1 Conditions to Generate Low Voltage Detection Interrupt Request** The D42 bit setting modified
			• **Table 6.2 Sampling Periods** Sampling period modified
		57	• **6.2 Cold Start-up/Warm Start-up Determine Function** added

Rev.	Date	Page	Summary
			Processor Mode
		59	• **Section structure and description** modified
		61	• **Figure 6.3 Memory Map in Each Processor Mode** Note 3 modified
			Bus
		65	• **Table 8.3 Processor Mode and Port Function** Note 3 modified
			Clock Generation Circuit
		84	• **Figure 9.4 MCD Register** Note 4 added
		86	• **Figure 9.6 TCSPR Register** Note 2 added
		88	• **Figure 9.8 PM2 Register** The PM24 and PM25 bit functions modified
		89	• **Figure 9.9 Main Clock Circuit Connection** Diagram modified
		90	• **Figure 9.10 Sub Clock Circuit Connection** Diagram modified
		91	• **Table 9.2 Bit Settings for On-Chip Oscillator Start Condition** added
		94	• **Table 9.4 CPU Clock Source and Bit Settings** Table modified, note 1 added
		95	• **9.3.4 fCAN** added
		97	• **9.5.2 Wait Mode** Structure and description modified
		99	• **9.5.3 Stop Mode** Structure and description modified
		102	• **Figure 9.13 Status Transition in Wait Mode and Stop Mode** Diagram modified, note 2 deleted
		103	• **Figure 9.14 Status Transition** Note 5 added
			Interrupt
		111	• **Table 11.2 Relocatable Vector Table** “Fault error” as interrupt source deleted, note 4 deleted, note 5 added
		114	• **Figure 11.3 Interrupt Control register (1)** Note 3 modified
		116	• **Figure 11.5 RLVL Register** Value after reset modified, note 3 modified, note 4 added
		120	• **11.6.6 Saving a Register** Description modified
		125	• **Figure 11.12 Key Input Interrupt** Diagram modified
		126	• **Figure 11.13 AIER Register** Value after reset revised
		127	• **Figure 11.14 Intelligent I/O Interrupt and CAN Interrupt** Notes 1 and 2 revised
		128	• Description revised, note 1 added
			Watchdog Timer
		132	• **Figure 12.2 WDC Register** Note 3 added
			DMAC
		136	• **Table 13.1 DMAC Specifications** DMA Transfer Cycle specification modified, note 2 added
		137	• **Figure 13.2 DM0SL to DM3SL Registers** Value after reset modified
			DMAC II
		147	• **Figure 14.1 RLVL Register** Value after reset modified, note 3 modified, note 4 added
		153	• **Figure 14.5 Transfer Cycle** Values in the diagram modified
			Timer
		154	• **Figure 15.1 Timer A Configuration** Diagram modified
		155	• **Figure 15.2 Timer B Configuration** Diagram modified
		160	• **Figure 15.7 TCSPR Register** Note 2 added
		178	• **Figure 15.21 TB0MR to TB5MR Registers** The TCK1 bit function modified
			Three-Phase Motor Control Timer Functions
		187	• **Figure 16.4 IDB0 and IDB1 Registers** Value after reset modified

Rev.	Date	Description: Page	Description: Summary
			Serial I/O
		194	• **Figure 17.1 UART0 to UART4 Block Diagram** Diagram modified
		195	• **Figure 17.2 UART5 to UART6 Block Diagram** Diagram modified
		200	• **Figure 17.7 U0C0 to U6C0 Registers** Note 3 added
		201	• **Figure 17.8 U0C1 to U4C1 Registers** Note 1 added
		202	• **Figure 17.9 U5C1 to U6C1 Registers** Value after reset modified
		204	• **Figure 17.11 U0SMR2 to U4SMR2 Registers** Value after reset modified
		212	• **Table 17.4 Pin Settings in Clock Synchronous Serial I/O Mode** The PSL0 register settings modified
		213	• **Table 17.7 Pin Settings in Clock Synchronous Serial I/O Mode** The PSL3 register settings modified
			• **Table 17.8 Pin Settings** The PS6 register settings modified
			• **Table 17.9 Pin Settings** The PS6 register settings modified
		220	• **Table 17.13 Pin Settings in UART Mode** The PSL0 register settings modified
		221	• **Table 17.17 Pin Settings** The PS6 register settings modified
			• **Table 17.18 Pin Settings** The PS6 register settings modified
		222	• **Figure 17.20 Transmit Operation** Diagram modified
		223	• **Figure 17.21 Receive Operation** Notes 1 and 2 revised
			• **17.2.1 Bit Rate** added
		231	• **Table 17.23 Pin Settings in I^2C Mode** The PSL0 register settings modified
			• **Table 17.25 Pin Settings** The PSL3 register settings modified
		238	• **Table 17.29 Pin Settings in Special Mode 2** The PSL0 register settings modified
			• **Table 17.31 Pin Settings** The PSL3 register settings modified
		239	• **17.4.1 SSi Input Pin Function** Description modified
		242	• **Table 17.32 GCI Mode Specifications** Transmit/Receive Start Condition modified
		244	• **Table 17.34 Pin Settings in GCI Mode** The PSL0 register settings deleted
			• **Table 17.36 Pin Settings** The PSL3 register settings modified
		247	• **Table 17.39 Pin Settings in IE Mode** The PSL0 register settings deleted
			• **Table 17.41 Pin Settings** The PSL3 register settings modified
		248	• Description in section 17.6 modified
		252	• **Table 17.44 Pin Settings in SIM Mode** The PSL0 register settings deleted
		253	• **Figure 17.35 SIM Interface Operation** Diagram modified
		257	• **Figure 17.40 IRCON Register** Address revised, the IRTPOL and IRRPOL function revised
			A/D Converter
		261	• **Table 18.1 A/D Converter Specifications** Notes 2 and 3 modified
		263	• **Figure 18.2 AD0CON0 Register** Notes 3, 5, and 7 modified, note 8 added
		264	• **Figure 18.3 AD0CON1 Register** Note 10 modified, note 11 and 12 added
		277	• **18.2.8 Analog Input Pin and External Sensor Equivalent Circuit** deleted
			• **18.2.8 Output Impedance of Sensor Equivalent Circuit under A/D Conversion** added
			D/A Converter
		281	• **Figure 19.4 D/A Converter Equivalent Circuit** Notes 3 and 4 modified

Rev.	Date	Description: Page	Description: Summary
			Intelligent I/O
		292	• **Figure 22.5 G1BCR1 Register** The RST2 bit function modified, note 2 modified
		293	• **Figure 22.6 G2BCR1 Register** Note 3 deleted
		295	• **Figure 22.8 G1TPR6 and G1TPR7 Registers** Note 1 modified
		296	• **Figure 22.9 G1POCR0 to G1POCR7 Registers** Note 6 and 7 modified
		301	• **Table 22.2 Base Timer Specifications** Base Timer Reset Condition modified, Interrupt Request modified
		303	• **Figure 22.14 Base Timer Block Diagram** Diagram modified
		304	• **Table 22.3 Base Timer Associated Register Settings** Table layout modified
		314	• **Table 22.8 Waveform Generating Function Associated Register Settings** Table layout modified
		330	• **Figure 22.31 G0RB and G1RB Registers** Bit 14 modified as the PER bit
		331	• **Figure 22.32 G1MR Register** Bit 5 and 4 modified as the PRY and PRYE bits
		332	• **Figure 22.33 G0EMR Register** The RXSL and TXSL bit names and functions modified
			• **Figure 22.33 G1EMR Register** The RXSL and TXSL bit functions modified
		333	• **Figure 22.34 G0ETC Register** Note 1 modified
			• **Figure 22.34 G1ETC Register** Bits 2 to 0 functions modified, note 1 modified
		334	• **Figure 22.35 G0ERC and G1ERC Registers** Note 1 modified
		336	• **Figure 22.37 G1IRF Register** Note 1 modified
		339	• **Table 22.16 Clock Synchronous Serial I/O Mode Specifications** Error Detection specification modified
		340	• **Table 22.18 Clock Settings in Clock Synchronous Serial I/O Mode** Setting value of the G1PO0 register revised
		341	• **Table 22.20 Pin Settings in Clock Synchronous Serial I/O Mode** Register settings modified
			• **Table 22.23 Pin Settings** Register settings deleted
		343	• **Table 22.24 UART Mode Specifications** Data Transfer Format, Error Detection and Selectable Function specifications modified, note 2 modified
		344	• **Table 22.25 Clock Settings in UART Mode** “Input form ISCLK1” setting deleted, note 3 modified
			• **Table 22.26 Register Settings in UART Mode** The PRY and PRYE bit functions added
		345	• **Figure 22.41 Transmit Operation** Conditions modified
			• **Figure 22.42 Receive Operation** Conditions modified
		346	• **22.4.3 HDLC Data Processing Mode** Description modified
			• **Table 22.29 HDLC Data Processing Mode Specifications** Items modified, Interrupt Request specifications modified
		347	• **Table 22.31 Clock Settings in HDLC Processing Mode** Setting value of the G1PO1 register modified
		348	• **Table 22.32 Register Settings in HDLC Processing Mode** The G1PO1 register function modified
		355	• **Table 22.35 Pin Settings in Variable Clock Synchronous Serial I/O Mode** The PD7 register settings modified

Rev.	Date	Description	
		Page	Summary
			CAN Module
		360	• NOTE added
			Real-Time Port
		415	• **Table 24.2 RTP-Associated Register Settings** The RTP32 bit function revised
			• **Table 24.4 Pin Settings** The PS1 register settings modified, note 1 revised
			• **Table 24.6 Pin Settings** $P10_4$ to $P10_7$ functions revised
			Programmable I/O Port
		432	• **Figure 25.15 PSC Register** Note 1 added
			• **Figure 25.15 PSC2 Register** Note 1 added
		433	• **Figure 25.16 PSC3 Register** Note 1 added
		436	• **Figure 25.19 PUR0 and PUR1 Registers** Note 1 modified
		439	• **Figure 25.22 IPSA Register** Note 1 added
		441	• **Table 25.3 Port P6 Peripheral Function Output Control** Bit 2 and bit 6 settings in the PS0 register modified
			• **Table 25.4 Port P7 Peripheral Function Output Control** Bit 0 setting in the PS1 register revised
			Flash Memory Version
		448	• **26.2.1 ROM Code Protect Function** Description modified
		449	• **Figure 26.2 ROMCP Address** Bits 5 and 4 functions modified, notes 2 to 4 modified, note 5 added
		452	• **Figure 26.5 FMR1 Register** Bits 0, 2 and 3 functions modified
		457	• **26.3.4.5 How to Access** Descriptions modified
		469	• **Table 26.7 Pin Description** $P7_6$ and $P7_7$ functions modified
			Electrical Characteristics
		478	• **Table 27.2 Recommended Operating Conditions** f(BCLK) standard added
		479	• **Table 27.3 Electrical Characteristics** R_{PULLUP} standard modified, Icc standard modified
		484	• **Table 27.10 Memory Expansion Mode and Microprocessor Mode** Formula on note 1 modified
		487	• **Table 27.22 Memory Expansion Mode and Microprocessor Mode** Formula on note 1 modified
		488	• **Table 27.23 Memory Expansion Mode and Microprocessor Mode** Formula on note 1 and 4 modified
		490	• **Figure 27.3 Vcc1=Vcc2=5V Timing Diagram** Values in the diagram modified, note 2 modified
		490	• **Figure 27.4 Vcc1=Vcc2=5V Timing Diagram** Values in the diagram modified, note 1 modified
		494	• **Table 27.24 Electrical Characteristics** R_{PULLUP} standard modified, Icc standard modified
		495	• **Table 27.25 A/D Conversion Characteristics** tCONV standard modified
		496	• **Table 27.28 Memory Expansion Mode and Microprocessor Mode** Formula on note 1 modified
		499	• **Table 27.40 Memory Expansion Mode and Microprocessor Mode** Formula on note 1 modified
		500	• **Table 27.41 Memory Expansion Mode and Microprocessor Mode** Formula on note 1 and 4 modified

Rev.	Date	Description: Page	Description: Summary
		 501 502	**Electrical Characteristics** • **Figure 27.7 Vcc1=Vcc2=3.3V Timing Diagram** Values in the diagram modified, note 2 modified • **Figure 27.8 Vcc1=Vcc2=3.3V Timing Diagram** Values in the diagram modified, note 1 modified
		 – 505 508 513 516 520 – 521	**Precautions** • **Processor Mode** Section deleted • **28.1 Reset** section added • **28.4 Clock Generation Circuit** Section structure and description modified • **28.6.3 INT Interrupt** Description modified • **28.8 Timer** Section structure and description modified • **28.9.2 UART Mode** Description modified • **Special Mode 2** Subsection deleted • **28.9.3 Special Mode 1 (I^2C Mode)** added • **Figure 28.4 Use of Capacitors to Reduce Noise** Diagram modified
1.50	Oct 20, 2007	All	**All in this manual** • Descriptions and formats unified • Notation of numbers changed (e.g. 00_2 → 00b, FF_{16} → FFh) • Notation of pin name changed (e.g. RTP00 → RTP_0, A15(/D15) → [A15/D15]) • Columns in pin settings tables are rearranged by the order of the setting procedure • [Term changed] Serial I/O → Serial interface Clock synchronous serial I/O mode → Clock synchronous mode Clock asynchronous serial I/O mode → Clock asynchronous mode Clock synchronous variable length → Variable data length clock synchronous Voltage detection circuit → Power supply voltage detection function Low voltage detection interrupt → Vdet4 detection interrupt Brown-out detection reset → Vdet3 detection function • [NOTE changed] -"Nothing is assigned. If necessary, set to 0. When read, the content is undefined."→ "Unimplemented. Write 0. Read as undefined value." -"Set the PD9 and PS3 registers immediately after the PRC2 bit in the PRCR register is set to "1" (write enable). Do not generate an interrupt or a DMA transfer between the instruction to set to the PRC2 bit to "1" and the instruction to set the PD9 and PS3 registers." → "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."
		Cover	**RENESAS 16/32-BIT SINGLE-CHIP MICROCOMPUTER** → **RENESAS MCU**
		–	**Keep safety first in your circuit designs!** deleted **Notes regarding these materials** partially modified
		–	**General Precautions in the Handling of MPU/MCU Products** added
		–	**How to Use This Manual** (revised overall) Sections **Purpose and Target Readers**, **Notation of Numbers and Symbols**, and **List of Abbreviations and Acronyms** added

Rev.	Date	Description: Page	Description: Summary
			Overview
		1	• Header **SINGLE-CHIP 16/32-BIT CMOS MICROCOMPUTER** → **RENESAS MCU**
			• **1.1 Features** title added, **1.1 Applications** changed to **1.1.1 Applications**
		2	• **1.2 Performance Overview** changed to **1.1.2 Specifications**
		2-5	• **Tables 1.1** to **1.4** Structure, descriptions in Specification field, NOTE, and value partially revised or deleted
			• **Real-Time Port** Item deleted, **ROM Correction Function** Item added
		8	• **1.3 Block Diagram** moved following the **1.2 Product List**
		6-7	• **1.2 Product List** Tables revised, NOTE 1 added
		9, 14, 15	• **Figures 1.3** to **1.5** Arrows for VSS and VCC deleted, NOTES partially modified
		11,17	• **Tables 1.9** and **1.13** CLKOUT pin moved from Bus Control Pin column to Control Pin column
		19-22	• **Tables 1.15** to **1.19** Descriptions revised, NOTE 1 added
			Memory
		26	• Text partially modified
			SFR
		34-39	• **Tables 4.8** to **4.13** NOTE “Set the PM13 bit in the PM1 register to 1 (2 wait states for SFR area) before accessing the CAN-associated registers.” added
		45	• **Table 4.19** The PSL5 register added to the Address field of 03BBh item, the PSL7 register added to the Address field of 03BFh item
			• [Register names changed]
		27	002Fh Low Voltage Detection Interrupt Register → Vdet4 Detection Interrupt Register
		34	01C1h UART5 Bit Rate Register → UART5 Baud Rate Register
			01C9h UART6 Bit Rate Register → UART6 Baud Rate Register
			01D0h UART5, UART6 Transmit/Receive Control Register 2 → UART5, UART6 Transmit/Receive Control Register
			01DBh to 01D8h Pulse Output Data Register → RTP Output Buffer Register
		41	0303h to 0302h Timer A1-1 Register → Timer A11 Register
			0305h to 0304h Timer A2-1 Register → Timer A21 Register
			0307h to 0306h Timer A4-1 Register → Timer A41 Register
		42	0340h Count Start Flag → Count Start Register
			0341h Clock Prescaler Reset Flag → Clock Prescaler Reset Register
			0342h One-Shot Start Flag → One-Shot Start Register
			0344h Up-Down Flag → Up/Down Select Register
			• [Value After Reset changed]
		27	000Fh WDC 000X XXX$_2$ → 00XX XXXXb
		27	002Fh D4INT 00$_{16}$ → XX00 0000b
		29	007Bh IIO6IC XX00 X000$_2$ → XXXX X000b
		31	00EFh G0CR XX00 X011$_2$ → 0000 X011b
		31	00FEh G0IRF 00$_{16}$ → 0000 XXXXb
		32	013Eh G1IRF 00$_{16}$ → 0000 XXXXb
		34	01C7h to 01C6h U5RB XXXX XXXX XXXX 0XXX$_2$ → XXXXh
		34	01CFh to 01CEh U6RB XXXX XXXX XXXX 0XXX$_2$ → XXXXh
		44	038Fh to 0382h AD07 to AD01 XXXX$_{16}$ → 00XXh

Rev.	Date	Page	Summary
		 47-50 49 –	**Reset** • Text, diagrams partially modified • **Table 5.1** Title and structure modified, NOTE 4 added • **Figure Brown-Out Detection Reset (Hardware Reset 2)** Figure title changed and moved into the chapter **6. Power Supply Voltage Detection Function**
		    52-54 51 55 56 57 58	**Power Supply Voltage Detection Function** (revised overall) • [Term changed] Reset level → Vdet3 Low voltage → Vdet4 • Order of register figures rearranged, bit name, description in Function field, and NOTE partially modified • **Figure 6.1** Block diagram modified (Block diagrams of voltage detection circuit, low voltage detection interrupt generation circuit, and cold start-up/warm start-up determine function are integrated) • **6.1 Vdet3 Detection Function** and **Figure 6.5** added • **6.2 Vdet4 Detection Function** added and **Table 6.2** 32MHz changed to 24MHz in CPU Clock field, NOTE 1 added • **Figure 6.6** modified • **6.2.1 Usage Notes on Vdet4 Detection Interrupt** Title and text partially modified • **6.3 Cold Start/Warm Start Determine Function** Text partially modified and **Figure 6.7** partially modified
		 59 60-61 62	**Processor Mode** • Boot mode added, **7.2 Setting of Processor Mode** Text partially modified, **Table 7.2** Structure modified and NOTES deleted • **Figures 7.1** and **7.2** NOTE modified • **Figure 7.3** Text and NOTE modified
		   63-79 64 65 67 68 69 70 71-78 77 78	**BUS** • [Term changed] signal → either input or output • Text partially modified • **Table 8.2** added • **Table 8.3** Structure and NOTE partially modified • **Figure 8.2** partially modified • **Tables 8.4** and **8.5** structure and text partially modified • **Figure 8.3** Bit symbol modified and NOTE partially modified • **Table 8.6** Text partially modified and NOTE 1 added • **Figures 8.4** to **8.11** partially modified • **Table 8.7** NOTE 1 deleted, text partially modified • **Table 8.8** CPU state and NOTE 1 added
		   82-88 80 81 82-88 89-104	**Clock Generation Circuits** • [Term changed] Normal Operating Mode → CPU Operating Mode • **Figures 9.2** to **9.8** Order of figures rearranged • **Table 9.1** Text partially modified • **Figure 9.1** Revised overall • **Figures 9.2** to **9.8** Bit Name, description in Function field, and NOTES partially modified • Text partially modified

Rev.	Date	Description	
		Page	Summary
			Clock Generation Circuits
		92	• **Figure 9.11** Flow chart partially modified
		93	• **Table 9.3** Structure partially modified, **Figure 9.12** Flow chart partially modified
		95	• **Table 9.4** Structure partially modified, **Table 9.5** Bit setting value partially modified, NOTE 3 deleted
		96	• **Figure 9.13** added (entering wait mode from Low-power consumption mode disabled)
		97	• “High-speed mode” and “Medium-speed mode” are changed to “Main clock mode”.
		97	• **9.5.1.7 Main Clock Direct Mode** added
		98	• **Table CPU Clock Source and Bit Settings** Structure modified became **Table 9.6 Operation Mode,** NOTES added
		99	• **9.5.2.2 Entering Wait Mode** Procedure became the flow chart
		101	• **Table 9.8** CAN interrupts usage conditions revised
		101-102	• **9.5.3.1 Entering Stop Mode** The program example added, procedures became the flow chart
		–	• **Figure Status Transition in Wait Mode and Stop Mode** deleted
		104	• **9.6 System Clock Protect Function** Procedure became the flow chart
			Protection
		105	• “desired address” changed to “SFR area”
			Interrupt
		106-127	• Text partially modified
		109	• **Figure 11.2** added
		111	• **Table 11.2** Vector table address of the reserved space revised
		112	• **Table 11.3** UART5,6, $\overline{\text{INT}}$, CAN1 wake-up, and reserved space added
		116	• **Figure 11.6** NOTE partially changed
		118	• **Table 11.5** Table changed overall
		119	• **Table 11.6** DMACII end-of-transfer interrupt added, Reset deleted
		120	• **Figure 11.9** modified overall
		123-124	• **Figures 11.11** and **11.12** added
		126	• **Figure 11.15** partially modified
		128	• **11.11 Intelligent I/O Interrupts, CAN Interrupts, UART5 and UART6 Transmit/Receive Interrupts, and INT6 to INT8 Interrupts** Overall structure and text modified, **Figure 11.17** partially modified
		129-130	• **Figure 11.18 IIO0IR to IIO11IR Registers** Bit Name added, **Figure 11.19 IO0IE to IIO11IE Registers** Bit Name changed, NOTE 2 added
		131-133	• **Table 11.7**, **Figures 11.20** and **11.21** added
			Watchdog Timer (revised overall)
		134-137	• Overall structure and text revised, and order of register figures rearranged
		134	• **Table 12.1** and **Table 12.2** added
			• **Table 12.2** Values in WDC7 bit in WDC register changed
		135	• **Figure 12.1** partially modified
		136	• **Figure 12.2** NOTE partially modified
		-	• Section **Count Source Protection Mode** deleted
			DMAC
			• [Term changed] memory (forward direction) → incremented address
		138-150	• Text partially revised
		139	• **Table 13.1** Text partially modified, NOTE 1 deleted
		141	• **Table 13.2** Description added to NOTE 3

Rev.	Date	Description: Page	Description: Summary
			DMAC
		142-145	• **Figures 13.3** to **13.6** Order of register figures rearranged, NOTE partially modified
		146-147	• **Figures 13.7** and **13.8** Two flow charts added
		-	• Figure **Transfer Cycle Examples with the Source-Read Bus Cycle** deleted
		149	• **Table 13.3** Title and structure partially modified
		150	• **Figure 13.9** Title and figure modified
			DMACII
			• [Term changed] relocatable address → incremented address
		151-158	• Text partially modified
		151	• **Table 14.1** Structure and text partially modified
		152	• **Figure 14.1** NOTE partially modified
		157-158	• **Figures 14.4** and **14.5** partially modified
			Timers
			• [Modification throughout all modes] Specification tables: Structure, text, and NOTES partially modified Operation timing diagrams: added or partially modified Register figures: Register figures of TA0MR to TA4MR in each mode are moved to earlier in the **15.1 Timer A** section.
		159-195	• Text partially modified, order of register figures rearranged
		162-171	• **Figures 15.4 to 15.13** Register name, bit name, description in Function field, and NOTE partially modified
		172	• **Tables 15.1** and **15.2** Structure partially modified, NOTE 1 added, NOTE 2 moved from NOTE 1
		174-178	• **15.1.2 Event Counter Mode** Structure and text modified
		-	• Figure **Counter Reset Timing** deleted
		182-183	• **Figures 15.19** and **15.20** Titles and figures partially modified
		189	• **Figure 15.26** Register name and bit name partially modified
		194-195	• **Figures 15.29** and **15.30** Titles and figures partially modified
			Three-Phase Motor Control Timer Function (fully revised)
			• [Term changed] Positive and negative phases concurrent active → Upper and lower arm simultaneous turn-on
		196-213	• Structure, text, tables, and figures modified or added
		198-206	• Order of register figures rearranged, **Figures 16.2** to **16.10** Register name, bit name, description in Function field, and NOTE partially modified
		198	• Bits INV01 and INV00 in **Figure 16.2 INVC0 Register** Bit name, Function changed • **Figure 16.7** Two cases of “when n > 1” and “when n = 1” added under “When bits INV01 and INV00 are set to 11b...”.
			Serial Interfaces (revised overall)
			• Chapter is divided into two sections **17.1 UART 0 to 4** and **17.2 UART 5,6** • Continuous receive mode is deleted in special mode 2 • IEBus mode is deleted, “special mode 4 (IE mode)” is now “special mode 4 (SIM mode)”, “special mode 5 (SIM mode)” is now “special mode 5 (IrDA mode)”, and “special mode 6 (IrDA mode)” is now “special mode 6 (IE mode)”.

Rev.	Date	Description: Page	Description: Summary
			Serial Interfaces • [Term changed] Transmit buffer → UiTB register Transmit register → transmit shift register Transfer data length → data length Bit rate → baud rate Transfer clock → serial clock, internal transmit clock, internal receive clock Transfer format → bit order Actual bit rate, bit rate → actual baud rate, target baud rate • [Modification throughout all modes] Specification tables: Structure and text partially modified Block diagrams: Partially modified Pin setting tables: Partially modified and NOTE added Register setting tables: Changed to flow chart Operation timing diagrams: Partially modified
		214-292	• Text modified
		216-224	• Order of register figures rearranged **Figures 17.2 to 17.10** Register name, bit name, description in Function field, and NOTE partially modified
		233,241	• **CTS/RTS Function**, **Procedure When the Communication Error is Occurred** added
		239,291	• Formulas for baud rate added
		246-247	• **Tables 17.10** and **17.11** Structure and text partially modified
		270	• **Figure 17.38** Bit name and description in Function fields partially modified
		272-292	• Section **17.2 UART5, UART6** added as a section.
		273-277	• **Figures 17.41** to **17.45** Register name, bit name, description in Function field, and NOTE partially modified
		275	• **Figure 17.43** Bit 5 is changed to a reserved bit.
			A/D Converter
		293-313	• Text partially modified
		294	• **Table 18.1** Structure, text, and NOTE partially modified
		295	• **Figure 18.1** Figure partially modified
		296-300	• **Figures 18.2** to **18.6** Bit name, description in Function field, and NOTE partially modified
		301	• **Tables 18.2** and **18.3** added
		302-309	• **Tables 18.4** to **18.10** Structure and text partially modified
		307	• **Figure 18.7** added
		312	• Section **18.3 Read from the AD0i Register (i = 0 to 7)** added
		312-313	• Section **18.4** and **Figure 18.9** Values revised
			D/A Converter
		314	• Text partially modified • **Figure 19.1** moved before **Table 19.2,** NOTE 1 added • **Table 19.2 Pin Settings** Note 1 added
		316	• **Figure 19.3** Diagram and NOTE partially modified
			CRC Calculation
		317	• Text partially modified

<table>
<tr><th rowspan="2">Rev.</th><th rowspan="2">Date</th><th colspan="2">Description</th></tr>
<tr><th>Page</th><th>Summary</th></tr>
<tr><td></td><td></td><td></td><td>Intelligent I/O
• [Term changed]
Modulated span → Number of modulated pulses
Transmit buffer → GiTB register
Transmit register → Transmit shift register
Transfer format → Bit order
Transfer data format, character bit, transfer bit length → Data length
Bit rate → Baud rate
Transfer clock → Serial clock, baud rate, internal transmit clock, internal receive clock
During transmit data processing → When HDLC frame data is generated
During received data processing → When source data is generated
• [Modification throughout functions and modes]
Specification tables: Structure, text, NOTE, and formula: Partially modified
Block diagrams: Partially modified, figures for communication function is moved to communication function section.
Pin setting tables: Partially modified, column sequence rearranged
Register setting tables: Became flow chart
Operation timing diagrams: Partially modified</td></tr>
<tr><td></td><td></td><td>322-401</td><td>• Text partially modified, order of register figures rearranged</td></tr>
<tr><td></td><td></td><td>325-336</td><td>• Figures 22.3 to 22.14 Bit name, description in Function field, and NOTE partially modified</td></tr>
<tr><td></td><td></td><td>347-349</td><td>• 22.2.1 Prescaler function and 22.2.2 Gate function Flow charts for register settings added</td></tr>
<tr><td></td><td></td><td>352</td><td>• Table 22.6 Registers PSL5 and PSL7, and NOTE added</td></tr>
<tr><td></td><td></td><td>362</td><td>• Table 22.11 Structure and title partially modified</td></tr>
<tr><td></td><td></td><td>363, 365</td><td>• Tables 22.12 and 22.13 Inversed output functions deleted from Selectable function fields</td></tr>
<tr><td></td><td></td><td>368</td><td>• 22.3.7 GiPOj Register reload timing select function added</td></tr>
<tr><td></td><td></td><td>371-379</td><td>• Figures 22.38 to 22.46 Bit name, description in Function field, and NOTE partially modified</td></tr>
<tr><td></td><td></td><td>374</td><td>• Figure 22.41 Bits SMODE and BSINT deleted from the GiEMR register, bits SOF, ASTE, and TBSF0 deleted from the GiETC register</td></tr>
<tr><td></td><td></td><td>375</td><td>• Figure 22.42 RBSF0 bit deleted</td></tr>
<tr><td></td><td></td><td>376</td><td>• Figure 22.43 BSERR bit deleted</td></tr>
<tr><td></td><td></td><td>381</td><td>• Table 22.16 NOTE partially modified, Table 22.17 PSL5 register and NOTE added</td></tr>
<tr><td></td><td></td><td>387</td><td>• Table 22.19 PSL5 register and NOTE added,
Table Clock Settings in UART Mode (Group 1) deleted</td></tr>
<tr><td></td><td></td><td>394-398</td><td>• Figures 22.56 to 22.60 Bit name, description in Function field, and NOTE partially modified</td></tr>
<tr><td></td><td></td><td>401</td><td>• Table 22.27 PSL7 register and NOTE 3 added</td></tr>
<tr><td></td><td></td><td>–</td><td>• Section IEBus Mode deleted</td></tr>
<tr><td></td><td></td><td></td><td>CAN Module</td></tr>
<tr><td></td><td></td><td>402</td><td>• Table 23.1 Structure and text partially modified</td></tr>
<tr><td></td><td></td><td>405</td><td>• Table 23.2 NOTE 1 added</td></tr>
<tr><td></td><td></td><td>406</td><td>• Figure 23.3 NOTE partially modified</td></tr>
<tr><td></td><td></td><td>411</td><td>• Figure 23.6 Function description for bits TRMSUCC, RECSUCC and STATE_RESET partially modified</td></tr>
<tr><td></td><td></td><td>415</td><td>• Figure 23.8 NOTE 1 added</td></tr>
<tr><td></td><td></td><td>417</td><td>• Figure 23.9 NOTE 1 added</td></tr>
</table>

Rev.	Date	Description	
		Page	Summary
			CAN Module
		418	• **Figure 23.10** added
		431	• **Figure 23.22** Descriptions in function fields partially modified
		438	• **Figures 23.28** and **23.29** partially modified
		439	• **Figure 23.30** Descriptions in Function fields and NOTE partially modified
		440	• **Table 23.4** Structure partially modified
		440	• **23.1.20.2 TRMACTIVE/INVALDATA Bit** Text partially modified
		443	• Subsections **23.1.21.1** and **23.1.21.2** Text partially modified
		449	• **23.2 CAN Clock and CPU Clock** Title and structure partially modified, a table and a figure deleted
		450	• **23.3 Setting and Timing in CAN-Associated Registers** Title partially modified
		451-453	• **Figures 23.38** to **23.40** partially modified
		454-458	• **23.4 CAN Interrupts** Structure partially modified, **Figure 23.42** partially modified, **Figures 23.41**, **23.43, tables 23.5**, and **23.6** added
			Real-Time Port (RTP)
		459	• Text partially modified, Specification table deleted
		459,460	• **Figure 24.1** and **24.3** partially modified
		460	• **Figure 24.2** Register name and bit name partially revised
		461	• **Table 24.1** NOTE 1 added, **RTP-Associated Register Settings** Table deleted
			Programmable I/O Ports
		462-463	• Text partially modified
		464-466	• **Figures 25.1** to **25.3** partially modified
		467,468	• **Figure 25.5** and **25.6** Descriptions in Function field and NOTE partially modified
		469-481	• **Figures 25.7** to **25.19** Bit name and descriptions in Function fields partially modified
		476	• **Figure 25.14** PSL5 register added
		477	• **Figure 25.15** PSL7 register added
		486	• **Figure 25.24** Descriptions in Function fields partially modified
		488	• **Tables 25.1** and **25.2** AVCC and AVSS deleted, **Figure 25.26** Partially modified
		489-493	• **Tables 25.3** to **25.13** partially modified
			Flash Memory
		494-519	• Text partially modified
		494	• **Tables 26.1** and **26.2** Structure, text, and NOTE partially modified
		495	• **Figure 26.1** Title and NOTE partially modified
		497	• **Figure 26.2** Bit name, descriptions in Function fields, and NOTE partially modified
		497	• **Figure 26.3** partially modified
		498	• **Table 26.3** Structure, title, text, and NOTE partially modified
		501	• **Figure 26.5** Value after reset revised
		502-504	• **Figures 26.6** to **26.8** Flow chart and NOTE partially modified
		–	• Subsection **Precautions in CPU Rewrite Mode** moved to **28. Usage Notes**
		511	• **Table 26.5** D0 to D7 are changed to b7 to b0 respectively, **Table 26.6** Text partially modified
		514	• **Table 26.7** Structure and text partially modified
		–	• Subsection **ID Code Verify Function** deleted

Rev.	Date	Description: Page	Description: Summary
			Flash Memory
		518	• **Figures 26.17** and **26.18** partially modified
		–	• Subsection **ROM Code Protect Function** deleted
			Electrical Characteristics
			• [Term changed] Low Voltage Reset → Hardware Reset 2 Low Voltage Detection → Vdet3 and Vdet4 detection circuit
		520	• **Table 27.1** Description in Condition field of Pd (Power consumption) partially modified
		522-527	• **Tables 27.4** to **27.9** f(BCLK) is changed to f(CPU)
		522	• **Table 27.4** Description added in Parameter field of f(CPU), f(VCO) added
		524-526, 542-544	• **Tables 27.5** to **27.7** and **Tables 27.31** to **27.33** Description in XCOUT and Hysteresis in Parameter fields partially modified
		526,544	• **Table 27.7** and **27.33** Structure and standard values revised, items in Measurement Condition and NOTE added
		528	• **Table 27.10** Description in Parameter field and NOTE partially modified
		529	• **Tables 27.11** and **27.12** Description in Parameter field and standard value partially modified
		531,547	• **Tables 27.19** and **27.42** added
		532	**• Table 27.24** Values revised, **Table 27.25** and **27.26 added**
		533	• **Table 27.27** Titles modified, NOTE added
		534	• **Table 27.28** moved to the last table in Timing Requirements
		535-536	• **Table 27.29** NOTE 3 added, **Table 26.30** NOTE 5 added
		538-541	• **Figures 27.3** to **27.6** Order rearranged, measurement condition modified
		542-545	• **Table 27.31** to **27.35** f(BCLK) revised to f(CPU)
		548	**• Table 27.47** Values revised, **Table27.48** and **27.49** added
		549	• **Table 27.50** Titles modified, NOTE added
		550	• **Table 27.51** Table moved to the last table in Timing Requirements
		551-552	• **Table 27.52** NOTE 3 added**, Table 27.53** NOTE 5 added
		553-556	• **Figures 27.7** to **27.10** Order rearranged
			Usage Note (chapter title changed)
			• [New section]
		558	**28.1.2 Power Supply Ripple**
		560	**28.3 Processor Mode**
		573	**28.10 Three-Phase Motor Control Timer Function**
		578	**28.14 CAN**
		557-582	• Text partially modified
		557	• Section **Reset** changed to **28.1 Power Supply, 28.1.1 Power-on** added, **Figure 28.1** partially modified
		558	• **28.1.3 Noise** partially modified and moved earlier in the chapter
		559	• **Table 28.4** added
		-	• Subsection **External Bus** deleted
		562-564	• **28.5 Clock Generation Circuits** Structure modified
		567-568	• **Figures 28.3** and **28.4** added
		574	• **28.11 Serial Interfaces** Structure modified
		577	• **28.13 Intelligent I/O** Structure modified
		580-581	• **28.16 Flash Memory** Structure modified

Rev.	Date	Description	
		Page	Summary
1.51	Jul 31, 2008	–	**All in this manual** [pin and bit symbol notation modified] • P5_5($\overline{EPM}$) → $\overline{EPM}$(P5_5) • P5_0($\overline{CE}$) → $\overline{CE}$(P5_0) • PM04, PM05 → PM05 and PM04 [description modified] • Title of group tables "(current table number / total tables)" added
			Overview
		19	• **1.5 Pin Descriptions** Chapter and table title changed to **Pin Functions**
		21	• **Table 1.17** Supply voltage for AN0_0 to AN0_7, AN2_0 to AN2_7 modified
			Special Function Registers (SFRs)
		46	• **Table 4.20** A value of After Reset column in 03FFh modified
			Power Supply Voltage Detection Circuit
		57	• **Figure 6.6** NOTE 1 "internal VDC" changed to "the main voltage regulator"
			Clock Generation Function
		81	• **Figure 9.1** "Charge pump" changed to "Loop filter"
		91	• **Table 9.2** "(The clock keeps running in stop mode)" deleted
		93	• **9.1.4 PLL Clock** Text partly revised
		97	• **9.5.1.3 Low-Speed Mode** Text partly revised
		97	• **9.5.1.4 Low-Power Consumption Mode** Text partly revised
		98	• **Table 9.6** Values of "-" in bits CM21 and CM17 in the Sub Clock row and in CM17 bit in the On-chip oscillator mode row changed to "0"
		99, 102	• **Figure 9.14** and **9.15** Note 1 partly modified
		101	• **Table 9.8** Word "the clock input to the CLKi pin (i = 0 to 6)" changed to "the external clock"
		104	• **9.6 System Clock Protect Function** and **Figure 9.16** Text partly revised
			Interrupts
		110	• **11.5.1 Fixed Vector Table** Text partly revised
		110	• **Table 11.1** Reference in the Watchdog timer row, "Reset" changed to "Voltage detection"
		121	• **Figure 11.10** "Interrupt request priority level detection result" changed to "Interrupt request level determination"
			Watchdog Timer
		134	• **Table 12.2** Values in WDC7 bit in WDC register revised
			DMAC
		139	• **Table 13.1** "DMA stop condition" modified
			Timer
		187	• **Figure 15.24** NOTE 3 "a 0" deleted
		195	• **Figure 15.30** NOTE 3 "a 0" deleted
			Three-Phase Motor Control Timer Function
		201	• **Figure 16.5** Value in operating mode select bits column "01b" changed to "10b"
		203	• **Figure 16.7 Function column of ICTB2 register** In the sentence "When bits INV01 and INV00 are set to 11b**",** two cases of when n > 1 and when n = 1 are added subsequently
			Serial Interfaces
		242	• **Table 17.7** NOTE 3 Sentence of "The IR bit in the SiRIC register..." deleted
		243	• **Figure 17.23** "IICM2 = 1" changed to "IICM = 0 or IICM2 = 1"
		266	• **Figure 17.33** The sentence "m:setting value of the UiBRG register" added

Rev.	Date	Page	Summary
			A/D Converter
		294	• **Figure 18.1** Figure partly modified
			Intelligent I/O
		352	• **Figure 22.25** Procedure “G1BCR1 register: BTS bit = 1” moved
		353	• **Figure 22.26** Procedure “G1BCR1 register: BTS bit = 1” moved
		385-386	• **Table 22.18 and Figure 22.51** Baud rate “0001h to FFFDh” changed to “0006h to FFFDh“
		385	• **Table 22.18** In Receive start condition column, “(“L” level)” added
		387	• **Figure 22.52** In the final step, “(“L” level)” added
			Programmable I/O Ports
		461	• **25.2 Port Pi Register (Pi Register, i = 0 to 15)** Text partly modified
		463-465	• **Figures 25.1 to 25.3** Figures partly modified
			Flash Memory
		498	• **Figure 26.4** NOTE 1 “the FMR01 bit” changed to “bits FMR01 and FMR02”, The sentence of “Set it by the ...” deleted
		500	• **Figure 26.5** NOTE 1 “while the FMR01 bit is set to 1” added
		502	• **Figure 26.7** In EW1 mode enabled procedure, text partly modified
		505-508	• **26.3.2.4 Program Command - 26.3.2.7 Read Lock Bit Status Command** Text revised partially
		508	• **Figure 26.12** “to the highest-order even address of block” in the second box deleted
		511	• **Figure 26.13** “[During...operation]” changed to “[When...is executed]”
		515	• **Figure 26.15** A wiring line connecting between pin no.7 and pin no.97 added
		516	• **Figure 26.16** “VCC” modified to “VCC2”
			Usage Notes
		558	• **Table 28.4** “EXTZ” deleted from Arithmetic row
		559	• **28.3 Processor Mode** Text partly modified
		561	• **28.5.1 Main Clock** Text partly modified
		569	• **28.9 Timers** Text partly modified
		578	• **28.15 Programmable I/O Ports** Text partly modified
		579	• **28.16 Flash Memory** Text partly modified

M32C/87 Group (M32C/87, M32C/87A, M32C/87B) Hardware Manual

Publication Date : Rev.0.20 Dec.16, 2004
Rev.1.51 Jul 31, 2008

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

M32C/87 Group (M32C/87, M32C/87A, M32C/87B)
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0180-0151