To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

M16C/28 Group (T-ver./V-ver.)

Hardware Manual

RENESAS MCU
M16C FAMILY / M16C/Tiny SERIES

Renesas Electronics
www.renesas.com

Rev.1.10 2007.03

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 M16C/28 Group (T-ver./V-ver.). 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.
Hardware manual	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	M16C/28 Group (T-ver./V-ver.) Hardware Manual	This hardware manual
Software manual	Description of CPU instruction set	M16C/60, M16C/20, M16C/Tiny Series Software Manual	REJ09B-0137
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

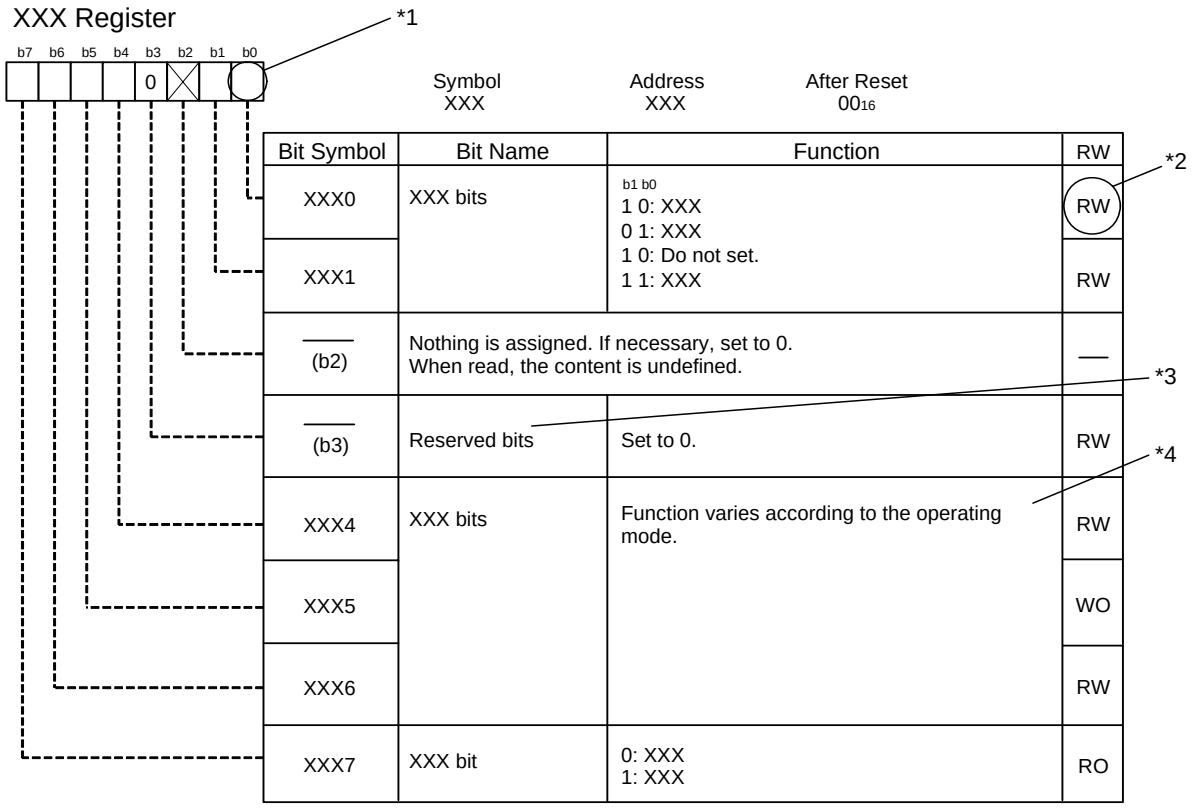
(2) Notation of Numbers

The indication "$_2$" is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication "$_{16}$" is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11_2
Hexadecimal: $EFA0_{16}$
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: Nothing is assigned.

*2

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

*3

- Reserved bit
  Reserved bit. Set to specified value.

*4

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

Quick Reference to Pages Classified by Address

Address	Register	Symbol	Page
0000₁₆			
0001₁₆			
0002₁₆			
0003₁₆			
0004₁₆	Processor mode register 0	PM0	34
0005₁₆	Processor mode register 1	PM1	34
0006₁₆	System clock control register 0	CM0	38
0007₁₆	System clock control register 1	CM1	39
0008₁₆			
0009₁₆	Address match interrupt enable register	AIER	77
000A₁₆	Protect register	PRCR	58
000B₁₆			
000C₁₆	Oscillation stop detection register	CM2	40
000D₁₆			
000E₁₆	Watchdog timer start register	WDTS	79
000F₁₆	Watchdog timer control register	WDC	79
0010₁₆ 0011₁₆ 0012₁₆	Address match interrupt register 0	RMAD0	77
0013₁₆			
0014₁₆ 0015₁₆ 0016₁₆	Address match interrupt register 1	RMAD1	77
0017₁₆			
0018₁₆			
0019₁₆			
001A₁₆			
001B₁₆			
001C₁₆	PLL control register 0	PLC0	42
001D₁₆			
001E₁₆	Processor mode register 2	PM2	35, 41
001F₁₆			
0020₁₆ 0021₁₆ 0022₁₆	DMA0 source pointer	SAR0	84
0023₁₆			
0024₁₆ 0025₁₆ 0026₁₆	DMA0 destination pointer	DAR0	84
0027₁₆			
0028₁₆ 0029₁₆	DMA0 transfer counter	TCR0	84
002A₁₆			
002B₁₆			
002C₁₆	DMA0 control register	DM0CON	83
002D₁₆			
002E₁₆			
002F₁₆			
0030₁₆ 0031₁₆ 0032₁₆	DMA1 source pointer	SAR1	84
0033₁₆			
0034₁₆ 0035₁₆ 0036₁₆	DMA1 destination pointer	DAR1	84
0037₁₆			
0038₁₆ 0039₁₆	DMA1 transfer counter	TCR1	84
003A₁₆			
003B₁₆			
003C₁₆	DMA1 control register	DM1CON	83
003D₁₆			
003E₁₆			
003F₁₆			

Address	Register	Symbol	Page
0040₁₆			
0041₁₆			
0042₁₆			
0043₁₆			
0044₁₆	INT3 interrupt control register	INT3IC	65
0045₁₆	IC/OC 0 interrupt control register	ICOC0IC	65
0046₁₆	IC/OC 1 interrupt control register, I^2C bus interface interrupt control register	ICOC1IC IICIC	65 65
0047₁₆	IC/OC base timer interrupt control register, SCLSDA interrupt control register	BTIC SCLDAIC	65 65
0048₁₆	SI/O4 interrupt control register, INT5 interrupt control register	S4IC, INT5IC	65 65
0049₁₆	SI/O3 interrupt control register, INT4 interrupt control register	S3IC, INT4IC	65 65
004A₁₆	UART2 Bus collision detection interrupt control register	BCNIC	65
004B₁₆	DMA0 interrupt control register	DM0IC	65
004C₁₆	DMA1 interrupt control register	DM1IC	65
004D₁₆			
004E₁₆	A/D conversion interrupt control register Key input interrupt control register	ADIC KUPIC	65 65
004F₁₆	UART2 transmit interrupt control register	S2TIC	65
0050₁₆	UART2 receive interrupt control register	S2RIC	65
0051₁₆	UART0 transmit interrupt control register	S0TIC	65
0052₁₆	UART0 receive interrupt control register	S0RIC	65
0053₁₆	UART1 transmit interrupt control register	S1TIC	65
0054₁₆	UART1 receive interrupt control register	S1RIC	65
0055₁₆	Timer A0 interrupt control register	TA0IC	65
0056₁₆	Timer A1 interrupt control register	TA1IC	65
0057₁₆	Timer A2 interrupt control register	TA2IC	65
0058₁₆	Timer A3 interrupt control register	TA3IC	65
0059₁₆	Timer A4 interrupt control register	TA4IC	65
005A₁₆	Timer B0 interrupt control register	TB0IC	65
005B₁₆	Timer B1 interrupt control register	TB1IC	65
005C₁₆	Timer B2 interrupt control register	TB2IC	65
005D₁₆	INT0 interrupt control register	INT0IC	65
005E₁₆	INT1 interrupt control register	INT1IC	65
005F₁₆	INT2 interrupt control register	INT2IC	65
0060₁₆			
0061₁₆			
0062₁₆			
0063₁₆			
0064₁₆			
0065₁₆			
0066₁₆			
0067₁₆			
0068₁₆			
0069₁₆			
006A₁₆			
006B₁₆			
006C₁₆			
006D₁₆			
006E₁₆			
006F₁₆			
0070₁₆			
0071₁₆			
0072₁₆			
0073₁₆			
0074₁₆			
0075₁₆			
0076₁₆			
0077₁₆			
0078₁₆ to 017F₁₆			

Note: The blank areas are reserved and cannot be accessed by users.

Quick Reference to Pages Classified by Address

Address	Register	Symbol	Page
0180₁₆			
0181₁₆			
0182₁₆			
0183₁₆			
0184₁₆			
0185₁₆			
0186₁₆			
0187₁₆ to 01AF₁₆			
01B0₁₆			
01B1₁₆			
01B2₁₆			
01B3₁₆	Flash memory control register 4 (Note 2)	FMR4	304
01B4₁₆			
01B5₁₆	Flash memory control register 1 (Note 2)	FMR1	303
01B6₁₆			
01B7₁₆	Flash memory control register 0 (Note 2)	FMR0	303
01B8₁₆			
01B9₁₆			
01BA₁₆			
01BB₁₆			
01BC₁₆			
01BD₁₆ to 019F₁₆			
0200₁₆			
0201₁₆			
0202₁₆			
0203₁₆			
0204₁₆			
0205₁₆			
0206₁₆			
0207₁₆			
0208₁₆			
0209₁₆			
020A₁₆			
020B₁₆			
020C₁₆			
020D₁₆			
020E₁₆			
020F₁₆			
0210₁₆	Low-power consumption control register 0	LPCC0	359
0211₁₆			
0212₁₆			
0213₁₆			
0214₁₆			
0215₁₆			
0216₁₆			
0217₁₆			
0218₁₆			
0219₁₆			
021A₁₆			
021B₁₆			
021C₁₆			
021D₁₆			
021E₁₆			
021F₁₆			
0210₁₆			
0211₁₆ to 02FD₁₆			
02FE₁₆			
02FF₁₆			

Address	Register	Symbol	Page
0240₁₆			
0241₁₆			
0242₁₆			
0243₁₆			
0244₁₆			
0245₁₆			
0246₁₆			
0247₁₆			
0248₁₆			
0249₁₆			
024A₁₆			
024C₁₆			
024D₁₆			
024E₁₆			
024F₁₆			
0250₁₆			
0251₁₆			
0252₁₆			
0253₁₆			
0254₁₆			
0255₁₆			
0256₁₆			
0257₁₆			
0258₁₆			
0259₁₆			
025A₁₆	Three-phase protect control register	TPRC	128
025B₁₆			
025C₁₆	0n-chip oscillator control register	ROCR	39
025D₁₆	Pin assignment control register	PACR	166, 289
025E₁₆	Peripheral clock select register	PCLKR	41
025F₁₆	Low-power consumption control register 1	LPCC1	359
0260₁₆			
0261₁₆			
0262₁₆			
0263₁₆			
0264₁₆			
0265₁₆			
0266₁₆			
0267₁₆			
0268₁₆			
0269₁₆			
026A₁₆			
026B₁₆			
026C₁₆			
026D₁₆ to 02DF₁₆			
02DF₁₆			
02E0₁₆	I^2C0 data shift register	S00	247
02E1₁₆			
02E2₁₆	I^2C0 address register	S0D0	246
02E3₁₆	I^2C0 control register 0	S1D0	248
02E4₁₆	I^2C0 clock control register	S20	247
02E5₁₆	I^2C0 start/stop condition control register	S2D0	252
02E6₁₆	I^2C0 control register 1	S3D0	250
02E7₁₆	I^2C0 control register 2	S4D0	251
02E8₁₆	I^2C0 status register	S10	249
02E9₁₆			
02EA₁₆			
02EB₁₆ to 02FE₁₆			
02FE₁₆			
02FF₁₆			

Note 1: The blank areas are reserved and cannot be accessed by users.
Note 2: This register is included in the flash memory version.

Quick Reference to Pages Classified by Address

Address	Register	Symbol	Page
0300_{16} 0301_{16}	TM, WG register 0	G1TM0, G1PO0	135, 136
0302_{16} 0303_{16}	TM, WG register 1	G1TM1, G1PO1	135, 136
0304_{16} 0305_{16}	TM, WG register 2	G1TM2, G1PO2	135, 136
0306_{16} 0307_{16}	TM, WG register 3	G1TM3, G1PO3	135, 136
0308_{16} 0309_{16}	TM, WG register 4	G1TM4, G1PO4	135, 136
$030A_{16}$ $030B_{16}$	TM, WG register 5	G1TM5, G1PO5	135, 136
$030C_{16}$ $030D_{16}$	TM, WG register 6	G1TM6, G1PO6	135, 136
$030E_{16}$ $030F_{16}$	TM, WG register 7	G1TM7, G1PO7	135, 136
0310_{16}	WG control register 0	G1POCR0	135
0311_{16}	WG control register 1	G1POCR1	135
0312_{16}	WG control register 2	G1POCR2	135
0313_{16}	WG control register 3	G1POCR3	135
0314_{16}	WG control register 4	G1POCR4	135
0315_{16}	WG control register 5	G1POCR5	135
0316_{16}	WG control register 6	G1POCR6	135
0317_{16}	WG control register 7	G1POCR7	135
0318_{16}	TM control register 0	G1TMCR0	134
0319_{16}	TM control register 1	G1TMCR1	134
$031A_{16}$	TM control register 2	G1TMCR2	134
$031B_{16}$	TM control register 3	G1TMCR3	134
$031C_{16}$	TM control register 4	G1TMCR4	134
$031D_{16}$	TM control register 5	G1TMCR5	134
$031E_{16}$	TM control register 6	G1TMCR6	134
$031F_{16}$	TM control register 7	G1TMCR7	134
0320_{16} 0321_{16}	Base timer register	G1BT	131
0322_{16}	Base timer control register 0	G1BCR0	131
0323_{16}	Base timer control register 1	G1BCR1	132
0324_{16}	TM prescale register 6	G1TPR6	134
0325_{16}	TM prescale register 7	G1TPR7	134
0326_{16}	Function enable register	G1FE	137
0327_{16}	Function select register	G1FS	137
0328_{16} 0329_{16}	Base timer reset register	G1BTRR	133
$032A_{16}$	Divider register	G1DV	132
$032B_{16}$			
$032C_{16}$			
$032D_{16}$			
$032E_{16}$			
$032F_{16}$			
0330_{16}	Interrupt request register	G1IR	138
0331_{16}	Interrupt enable register 0	G1IE0	139
0332_{16}	Interrupt enable register 1	G1IE1	139
0333_{16}			
0334_{16}			
0335_{16}			
0336_{16}			
0337_{16}			
0338_{16}			
0339_{16}			
$033A_{16}$			
$033B_{16}$			
$033C_{16}$			
$033D_{16}$			
$033E_{16}$	NMI digital debounce register	NDDR	290
$033F_{16}$	P17 digital debounce register	P17DDR	290

Address	Register	Symbol	Page
0340_{16}			
0341_{16}			
0342_{16} 0343_{16}	Timer A1-1 register	TA11	119
0344_{16} 0345_{16}	Timer A2-1 register	TA21	119
0346_{16} 0347_{16}	Timer A4-1 register	TA41	119
0348_{16}	Three-phase PWM control register 0	INVC0	116
0349_{16}	Three-phase PWM control register 1	INVC1	117
$034A_{16}$	Three-phase output buffer register 0	IDB0	118
$034B_{16}$	Three-phase output buffer register 1	IDB1	118
$034C_{16}$	Dead time timer	DTT	118
$034D_{16}$	Timer B2 interrupt occurrence frequency set counter	ICTB2	118
$034E_{16}$	Position-data-retain function contol register	PDRF	126
$034F_{16}$			
0350_{16}			
0351_{16}			
0352_{16}			
0353_{16}			
0354_{16}			
0355_{16}			
0356_{16}			
0357_{16}			
0358_{16}	Port Function control register	PFCR	128
0359_{16}			
$035A_{16}$			
$035B_{16}$			
$035C_{16}$			
$035D_{16}$			
$035E_{16}$	Interrupt request cause select register 2	IFSR2A	66
$035F_{16}$	Interrupt request cause select register	IFSR	66, 74
0360_{16}	SI/O3 transmit/receive register	S3TRR	207
0361_{16}			
0362_{16}	SI/O3 control register	S3C	207
0363_{16}	SI/O3 bit rate generator	S3BRG	207
0364_{16}	SI/O4 transmit/receive register	S4TRR	207
0365_{16}			
0366_{16}	SI/O4 control register	S4C	207
0367_{16}	SI/O4 bit rate generator	S4BRG	207
0368_{16}			
0369_{16}			
$036A_{16}$			
$036B_{16}$			
$036C_{16}$			
$036D_{16}$			
$036E_{16}$			
$036F_{16}$			
0370_{16}			
0371_{16}			
0372_{16}			
0373_{16}			
0374_{16}	UART2 special mode register 4	U2SMR4	168
0375_{16}	UART2 special mode register 3	U2SMR3	168
0376_{16}	UART2 special mode register 2	U2SMR2	167
0377_{16}	UART2 special mode register	U2SMR	167
0378_{16}	UART2 transmit/receive mode register	U2MR	164
0379_{16}	UART2 bit rate generator	U2BRG	163
$037A_{16}$ $037B_{16}$	UART2 transmit buffer register	U2TB	163
$037C_{16}$	UART2 transmit/receive control register 0	U2C0	165
$037D_{16}$	UART2 transmit/receive control register 1	U2C1	166
$037E_{16}$ $037F_{16}$	UART2 receive buffer register	U2RB	163

Note : The blank areas are reserved and cannot be accessed by users.

Quick Reference to Pages Classified by Address

Address	Register	Symbol	Page
0380₁₆	Count start flag	TABSR	93, 107, 121
0381₁₆	Clock prescaler reset flag	CPSRF	94, 107
0382₁₆	One-shot start flag	ONSF	94
0383₁₆	Trigger select register	TRGSR	94, 121
0384₁₆	Up-down flag	UDF	93
0385₁₆			
0386₁₆ 0387₁₆	Timer A0 register	TA0	93
0388₁₆ 0389₁₆	Timer A1 register	TA1	93
038A₁₆ 038B₁₆	Timer A2 register	TA2	93
038C₁₆ 038D₁₆	Timer A3 register	TA3	93
038E₁₆ 038F₁₆	Timer A4 register	TA4	93
0390₁₆ 0391₁₆	Timer B0 register	TB0	107
0392₁₆ 0393₁₆	Timer B1 register	TB1	107
0394₁₆ 0395₁₆	Timer B2 register	TB2	107
0396₁₆	Timer A0 mode register	TA0MR	92,122
0397₁₆	Timer A1 mode register	TA1MR	92,122
0398₁₆	Timer A2 mode register	TA2MR	92,122
0399₁₆	Timer A3 mode register	TA3MR	92,122
039A₁₆	Timer A4 mode register	TA4MR	92,122
039B₁₆	Timer B0 mode register	TB0MR	106,122
039C₁₆	Timer B1 mode register	TB1MR	106,122
039D₁₆	Timer B2 mode register	TB2MR	106,122
039E₁₆	Timer B2 special mode register	TB2SC	120,216
039F₁₆			
03A0₁₆	UART0 transmit/receive mode register	U0MR	164
03A1₁₆	UART0 bit rate generator	U0BRG	163
03A2₁₆ 03A3₁₆	UART0 transmit buffer register	U0TB	163
03A4₁₆	UART0 transmit/receive control register 0	U0C0	165
03A5₁₆	UART0 transmit/receive control register 1	U0C1	165
03A6₁₆ 03A7₁₆	UART0 receive buffer register	U0RB	163
03A8₁₆	UART1 transmit/receive mode register	U1MR	164
03A9₁₆	UART1 bit rate generator	U1BRG	163
03AA₁₆ 03AB₁₆	UART1 transmit buffer register	U1TB	163
03AC₁₆	UART1 transmit/receive control register 0	U1C0	165
03AD₁₆	UART1 transmit/receive control register 1	U1C1	166
03AE₁₆ 03AF₁₆	UART1 receive buffer register	U1RB	163
03B0₁₆	UART transmit/receive control register 2	UCON	165
03B1₁₆			
03B2₁₆			
03B3₁₆			
03B4₁₆ 03B5₁₆	CRC snoop address register	CRCSAR	277
03B6₁₆	CRC mode register	CRCMR	277
03B7₁₆			
03B8₁₆	DMA0 request cause select register	DM0SL	82
03B9₁₆			
03BA₁₆	DMA1 request cause select register	DM1SL	83
03BB₁₆			
03BC₁₆ 03BD₁₆	CRC data register	CRCD	277
03BE₁₆	CRC input register	CRCIN	277
03BF₁₆			

Address	Register	Symbol	Page
03C0₁₆ 03C1₁₆	A/D register 0	AD0	215
03C2₁₆ 03C3₁₆	A/D register 1	AD1	215
03C4₁₆ 03C5₁₆	A/D register 2	AD2	215
03C6₁₆ 03C7₁₆	A/D register 3	AD3	215
03C8₁₆ 03C9₁₆	A/D register 4	AD4	215
03CA₁₆ 03CB₁₆	A/D register 5	AD5	215
03CC₁₆ 03CD₁₆	A/D register 6	AD6	215
03CE₁₆ 03CF₁₆	A/D register 7	AD7	215
03D0₁₆			
03D1₁₆			
03D2₁₆	A/D trigger control register	ADTRGCON	214
03D3₁₆	A/D convert status register 0	ADSTAT0	215
03D4₁₆	A/D control register 2	ADCON2	213
03D5₁₆			
03D6₁₆	A/D control register 0	ADCON0	213
03D7₁₆	A/D control register 1	ADCON1	213
03D8₁₆			
03D9₁₆			
03DA₁₆			
03DB₁₆			
03DC₁₆			
03DD₁₆			
03DE₁₆			
03DF₁₆			
03E0₁₆	Port P0 register	P0	287
03E1₁₆	Port P1 register	P1	287
03E2₁₆	Port P0 direction register	PD0	286
03E3₁₆	Port P1 direction register	PD1	286
03E4₁₆	Port P2 register	P2	287
03E5₁₆	Port P3 register	P3	287
03E6₁₆	Port P2 direction register	PD2	286
03E7₁₆	Port P3 direction register	PD3	286
03E8₁₆			
03E9₁₆			
03EA₁₆			
03EB₁₆			
03EC₁₆	Port P6 register	P6	287
03ED₁₆	Port P7 register	P7	287
03EE₁₆	Port P6 direction register	PD6	286
03EF₁₆	Port P7 direction register	PD7	286
03F0₁₆	Port P8 register	P8	287
03F1₁₆	Port P9 register	P9	287
03F2₁₆	Port P8 direction register	PD8	286
03F3₁₆	Port P9 direction register	PD9	286
03F4₁₆	Port P10 register	P10	287
03F5₁₆			
03F6₁₆	Port P10 direction register	PD10	286
03F7₁₆			
03F8₁₆			
03F9₁₆			
03FA₁₆			
03FB₁₆			
03FC₁₆	Pull-up control register 0	PUR0	288
03FD₁₆	Pull-up control register 1	PUR1	288
03FE₁₆	Pull-up control register 2	PUR2	288
03FF₁₆	Port control register	PCR	289

Note : The blank areas are reserved and cannot be accessed by users.

1. Overview

1.1 Features

The M16C/28 Group (T-ver./V-ver.) of single-chip control MCU incorporates the M16C/60 Series CPU core, employing the high-performance silicon gate CMOS technology and sophisticated instructions for a high level of efficiency. The M16C/28 Group (T-ver./V-ver.) is housed in 64-pin and 80-pin plastic molded LQFP packages. This single-chip MCU operates using sophisticated instructions featuring a high level of instruction efficiency. This MCU is capable of executing instructions at high speed, makes it suitable for control of cars and LAN system of FA. In addition, the CPU core boasts a multiplier and DMAC for high-speed processing to make adequate for office automation, communication devices, and other high-speed processing applications.
, also making it suitable for control of various OA, communication, and industrial equipment which requires high-speed arithmetic/logic operations.

1.1.1 Applications

Automotive body, car audio, LAN system of FA, etc.

1.1.2 Specifications

Table 1.1 lists performance overview of M16C/28 Group (T-ver./V-ver.) 80-pin package.

Table 1.2 lists performance overview of M16C/28 Group (T-ver./V-ver.) 64-pin package.

Table 1.1 Performance Overview of M16C/28 Group (T-ver./V-ver.) (80-Pin Package)

Item		Performance
CPU	Number of basic instructions	91 instructions
	Shortest instruction excution time	50 ns (f(BCLK)= 20MHz, VCC= 3.0 to 5.5V) (T-ver.) 50 ns (f(BCLK)= 20MHz, VCC= 4.2 to 5.5V, -40 to 105°C) (V-ver.) 62.5 ns (f(BCLK)= 16MHz, VCC= 4.2 to 5.5V, -40 to 125°C) (V-ver.)
	Operation mode	Single chip mode
	Address space	1 Mbytes
	Memory capacity	ROM/RAM : See **Table 1.3** and **Table 1.4**
Peripheral Function	port	Input/Output : 71 lines
	Multifunction timer	TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels Three-phase Motor Control Timer TimerS (Input Capture/Output Compare) : 16bit base timer x 1 channel (Input/Output x 8 channels)
	Serial I/O	2 channels (UART, clock synchronous serial I/O) 1 channel (UART, clock synchronous serial I/O, I^2C bus, or IEbus(1)) 2 channels (Clock synchronous serial I/O) 1 channel (Multi-Master I^2C bus)
	A/D converter	10 bits x 27 channels
	DMAC	2 channels
	CRC calculation circuit	2 polynomial (CRC-CCITT and CRC-16) with MSB/LSB selectable
	Watchdog timer	15 bits x 1 channel (with prescaler)
	Interrupt	25 internal and 8 external sources, 4 software sources, 7 levels
	Clock generation circuit	4 circuits • Main clock, • Sub-clock (These circuits contain a built-in feedback resistor) • On-chip oscillator(main-clock oscillation stop detect function) • PLL frequency synthesizer
	Oscillation Stop Detect Function	Main clock oscillation stop, re-oscillation detect function
	Voltage detection circuit	Not available
Electrical Characteristics	Power supply voltage	VCC=3.0 to 5.5V (T-ver.)
		VCC=4.2 to 5.5V (V-ver.)
	Power consumption	18mA (VCC=5V, f(BCLK)=20MHz) 25 µA (VCC=5V, f(BCLK)=f(XCIN)=32kHz on RAM) 3 µA (VCC=5V, f(BCLK)=f(XCIN)=32kHz, in wait mode) 0.8 µA (VCC=5V, in stop mode)
Flash Memory	Program/erase voltage	3.0V to 5.5V (T-ver.) 4.2V to 5.5V (V-ver.)
	Number of program/erase	100 times (all space) or 1,000 times (blocks 0 to 4)/ 10,000 times (blocks A and B(2))
Operating Ambient Temperature		-40 to 85°C (T-ver.), -40 to 125°C (V-ver.)
Package		80-pin plastic mold LQFP

NOTES:
1. IEBus is a trademark of NEC Electronics Corporation.
2. Refer to **Table 1.5** and **Table 1.6** for number of program/erase endurance and ambient temperature.

Table 1.2 Performance Overview of M16C/28 Group (64-Pin Package)

Item		Performance
CPU	Number of basic instructions	91 instructions
	Shortest instruction excution time	50 ns (f(BCLK)= 20MHz, VCC= 3.0 to 5.5V) (T-ver.) 50 ns (f(BCLK)= 20MHz, VCC= 4.2 to 5.5V, -40 to 105°C) (V-ver.) 62.5 ns (f(BCLK)= 16MHz, VCC= 4.2 to 5.5V, -40 to 125°C) (V-ver.)
	Operation mode	Single chip mode
	Address space	1 Mbytes
	Memory capacity	ROM/RAM : See **Table 1.3** and **Table 1.4**
Peripheral Function	port	Input/Output : 55 lines
	Multifunction timer	TimerA:16 bits x 5 channels, TimerB:16 bits x 3 channels Three-phase Motor Control Timer TimerS (Input Capture/Output Compare) : 16bit base timer x 1 channel (Input/Output x 8 channels)
	Serial I/O	2 channels (UART, clock synchronous serial I/O) 1 channel (UART, clock synchronous serial I/O, I^2C bus, or IEbus(1)) 1 channel (Clock synchronous serial I/O) 1 channel (Multi-Master I^2C bus)
	A/D converter	10 bits x 16 channels
	DMAC	2 channels
	CRC calculation circuit	2 polynomial (CRC-CCITT and CRC-16) with MSB/LSB selectable
	Watchdog timer	15 bits x 1 channel (with prescaler)
	Interrupt	24 internal and 8 external sources, 4 software sources, 7 levels
	Clock generation circuit	4 circuits • Main clock, • Sub-clock (These circuits contain a built-in feedback resistor) • On-chip oscillator(main-clock oscillation stop detect function) • PLL frequency synthesizer
	Oscillation Stop Detect Function	Main clock oscillation stop, re-oscillation detect function
	Voltage detection circuit	Not available
Electrical Characteristics	Power supply voltage	VCC=3.0 to 5.5V (T-ver.) VCC=4.2 to 5.5V (V-ver.)
	Power consumption	18mA (VCC=5V, f(BCLK)=20MHz) 25 µA (VCC=5V, f(BCLK)=f(XCIN)=32kHz on RAM) 3 µA (VCC=5V, f(BCLK)=f(XCIN)=32kHz, in wait mode) 0.8 µA (VCC=5V, in stop mode)
Flash Memory	Program/erase voltage	3.0V to 5.5V (T-ver.) 4.2V to 5.5V (V-ver.)
	Number of program/erase	100 times (all space) or 1,000 times (blocks 0 to 4)/ 10,000 times (blocks A and B(2))
Operating Ambient Temperature		-40 to 85°C (T-ver.), -40 to 125°C (V-ver.)
Package		64-pin plastic mold LQFP

NOTES:
1. IEBus is a trademark of NEC Electronics Corporation.
2. Refer to **Table 1.5** and **Table 1.6** for number of program/erase endurance and ambient temperature.

1.2 Block Diagram

Figure 1.1 is a block diagram of the M16C/28 Group (T-ver./V-ver.), 80-pin package.

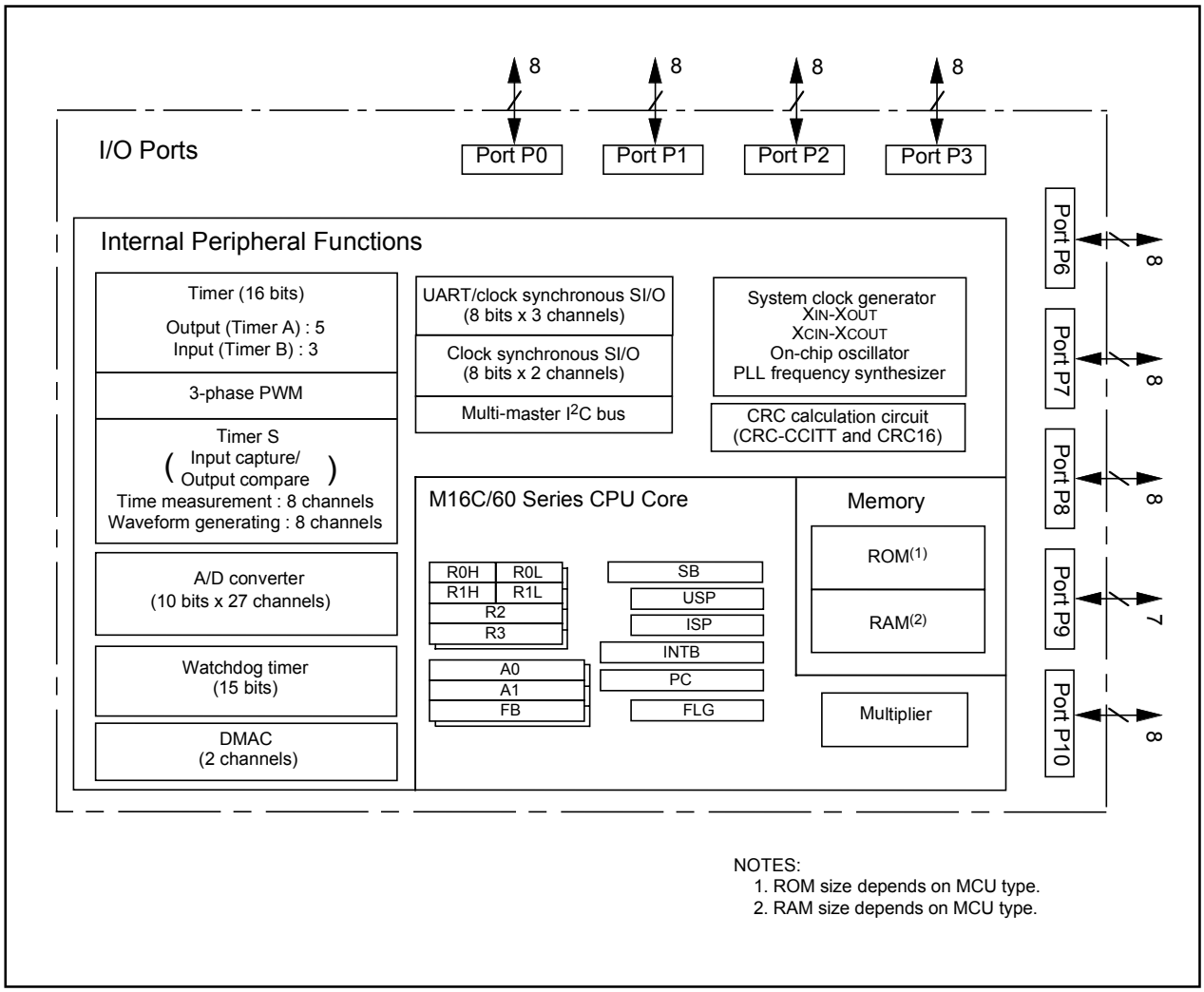


Figure 1.1 M16C/28 Group (T-ver./V-ver.), 80-Pin Package Block Diagram

Figure 1.2 is a block diagram of the M16C/28 Group (T-ver./V-ver.), 64-pin package.

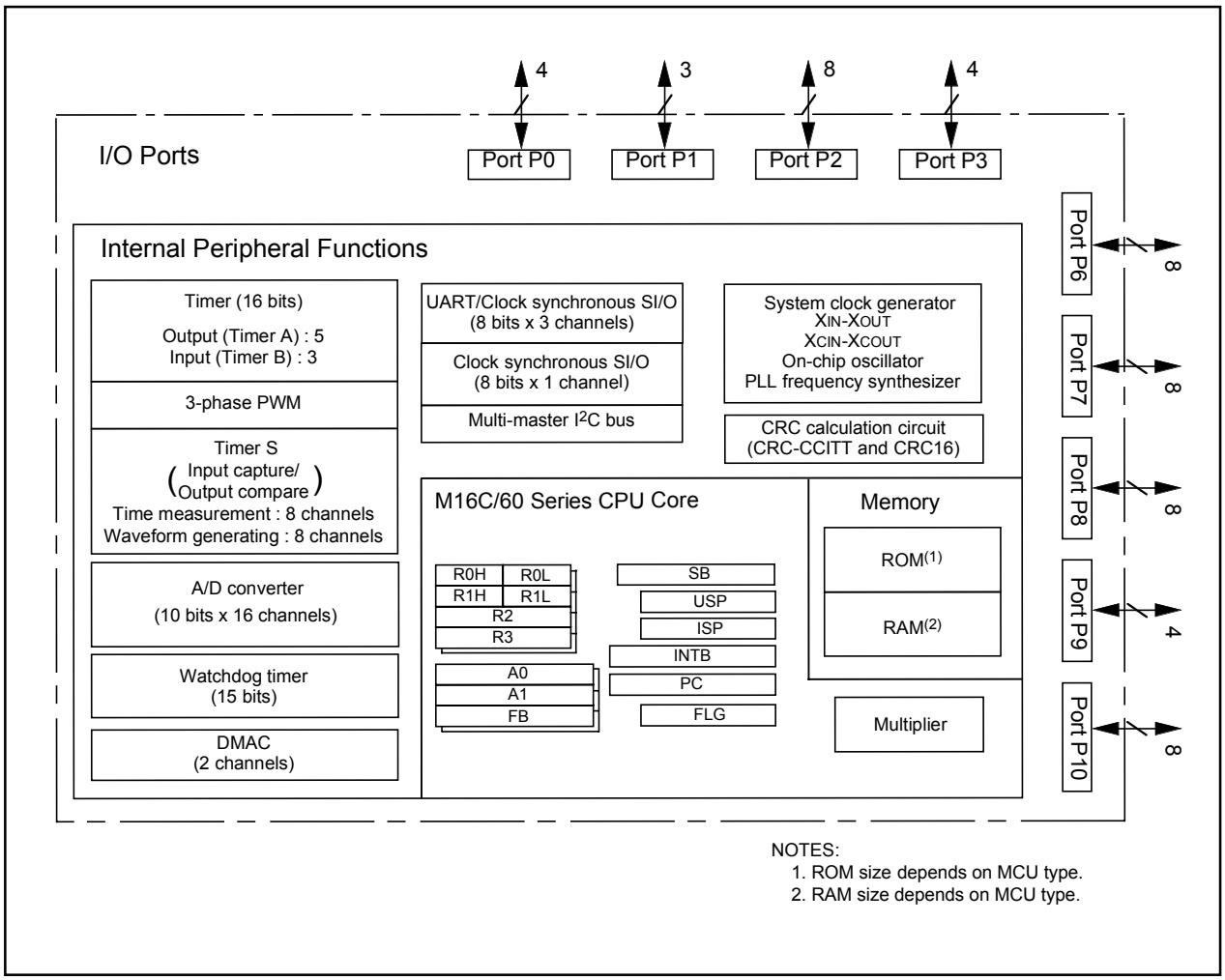


Figure 1.2 M16C/28 Group (T-ver./V-ver.), 64-Pin Package Block Diagram

1.3 Product List

Tables 1.3 and 1.4 list the M16C/28 Group (T-ver./V-ver.) products and **Figure 1.3** shows the type numbers, memory sizes and packages. **Tables 1.5 and 1.8** list the product code for M16C/28 Group.
Figure 1.4 shows the marking diagram of flash memory version for M16C/28 Group T version. **Figure 1.5** shows the marking diagram of flash memory version for M16C/28 Group V version.

Table 1.3 Product List (1) -T Version **As of March, 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30280FATHP	96 K + 4 K	8K	PLQP0080KB-A (80P6Q-A)	Flash Memory	U3, U7
M30281FATHP	96 K + 4 K	8K	PLQP0064KB-A (64P6Q-A)		
M30280M8T-XXXHP	64 K	4K	PLQP0080KB-A (80P6Q-A)	Mask ROM	U0
M30280MAT-XXXHP	96 K	8K			
M30281M8T-XXXHP	64 K	4K	PLQP0064KB-A (64P6Q-A)		
M30281MAT-XXXHP	96 K	8K			

Table 1.4 Product List (2) -V Version **As of March, 2007**

Type Number	ROM Capacity	RAM Capacity	Package Type	Remarks	Product Code
M30280FAVHP	96 K + 4 K	8K	PLQP0080KB-A (80P6Q-A)	Flash Memory	U3, U7
M30281FAVHP	96 K + 4 K	8K	PLQP0064KB-A (64P6Q-A)		
M30280M8V-XXXHP	64 K	4K	PLQP0080KB-A (80P6Q-A)	Mask ROM	U0
M30280MAV-XXXHP	96 K	8K			
M30281M8V-XXXHP	64 K	4K	PLQP0064KB-A (64P6Q-A)		
M30281MAV-XXXHP	96 K	8K			

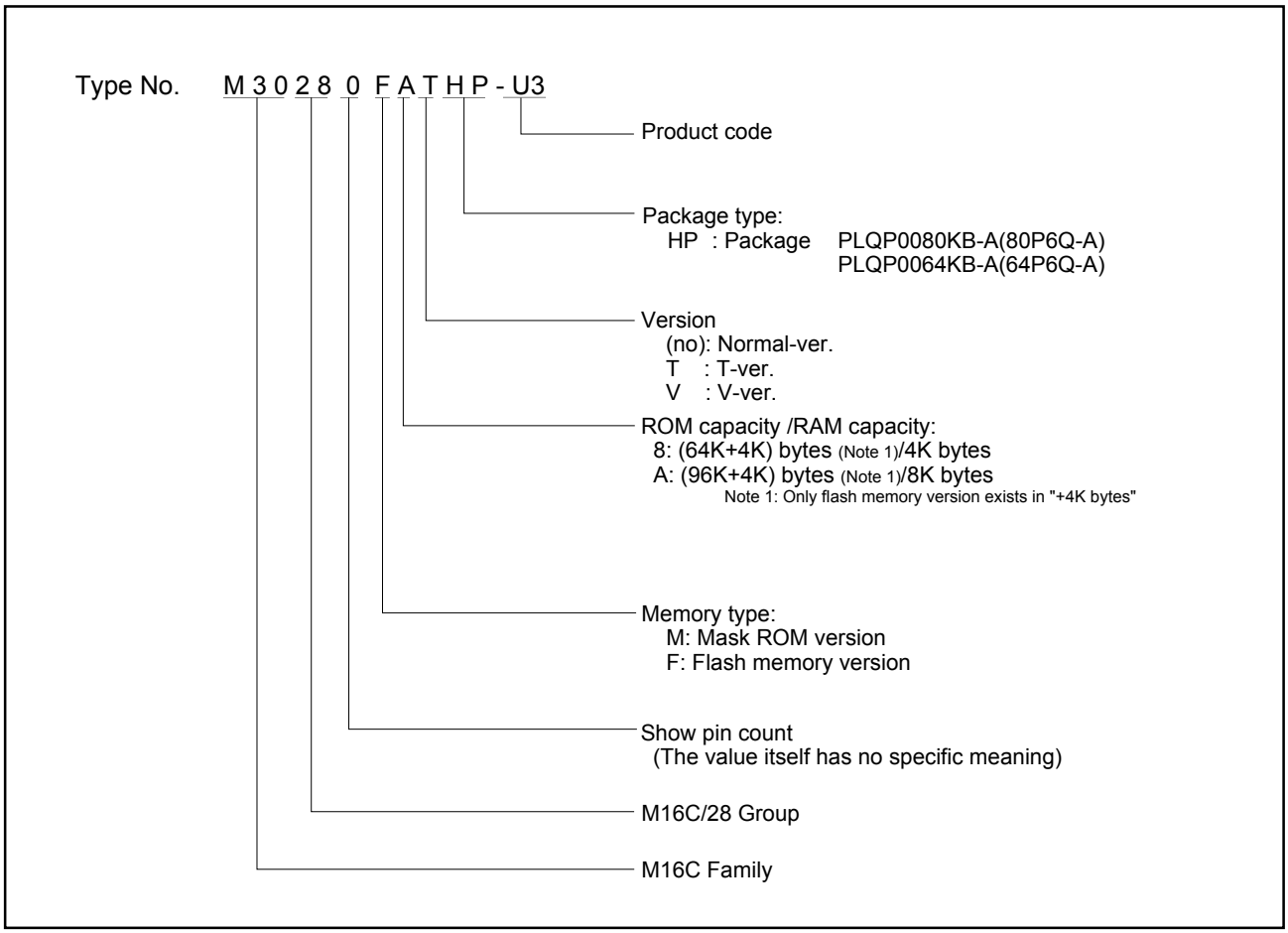


Figure 1.3 Product Numbering System

Table 1.5 Product Code of Flash Memory Version -M16C/28 Group T-ver.

Product Code	Package	Internal ROM (Program Space: Blocks 0 to 4)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temperature
		Program and Erase Endurance	Temperature Range	Program and Erase Endurance	Temperature Range	
U3	Lead-free	100	0 to 60°C	100	-40 to 85°C	-40 to 85°C
U7		1,000		10,000		

Table 1.6 Product Code of Flash Memory Version -M16C/28 Group V-ver.

Product Code	Package	Internal ROM (Program Space: Blocks 0 to 4)		Internal ROM (Data Space: Blocks A and B)		Operating Ambient Temperature
		Program and Erase Endurance	Temperature Range	Program and Erase Endurance	Temperature Range	
U3	Lead-free	100	0 to 60°C	100	-40 to 125°C	-40 to 125°C
U7		1,000		10,000		

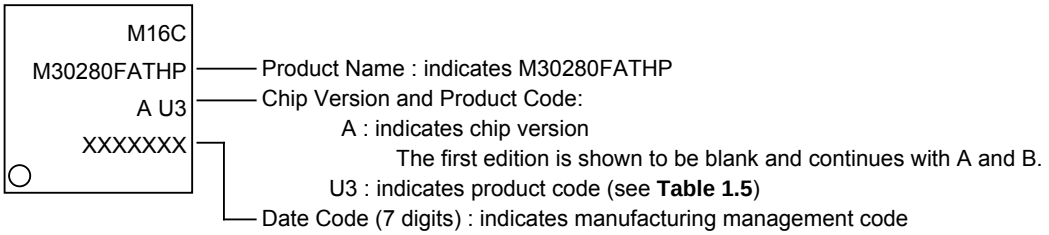
Table 1.7 Product Code of Mask ROM Version -M16C/28 Group T-ver.

Product Code	Package	Operating Ambient Temperature
U0	Lead-free	-40 to 85°C

Table 1.8 Product Code of Mask ROM Version -M16C/28 Group V-ver.

Product Code	Package	Operating Ambient Temperature
U0	Lead-free	-40 to 125°C

(1) Flash Memory Version, PLQP0080KB-A (80P6Q-A), T-ver.



(2) Flash Memory Version, PLQP0064KB-A (64P6Q-A), T-ver.

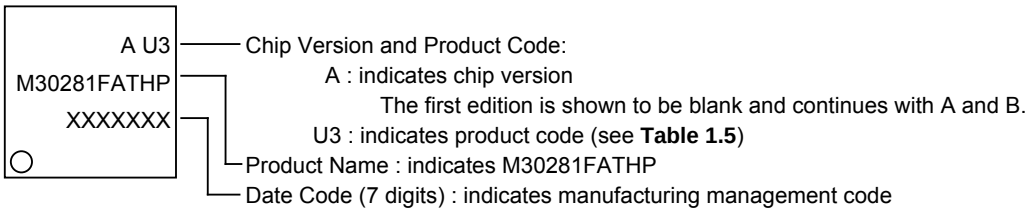
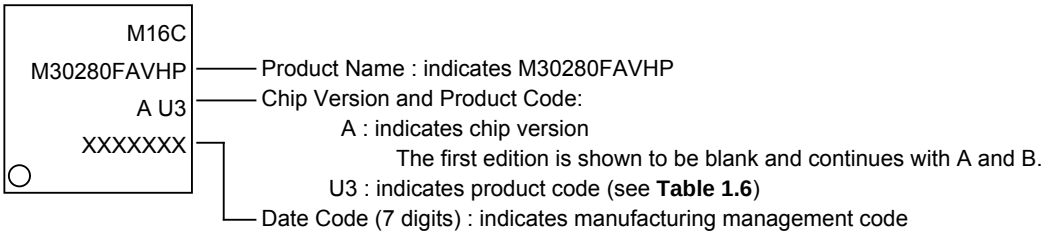


Figure 1.4 Marking Diagram of Flash Memory Version - M16C/28 Group T-ver. (Top View)

(1) Flash Memory Version, PLQP0080KB-A (80P6Q-A), V-ver.



(2) Flash Memory Version, PLQP0064KB-A (64P6Q-A), V-ver.

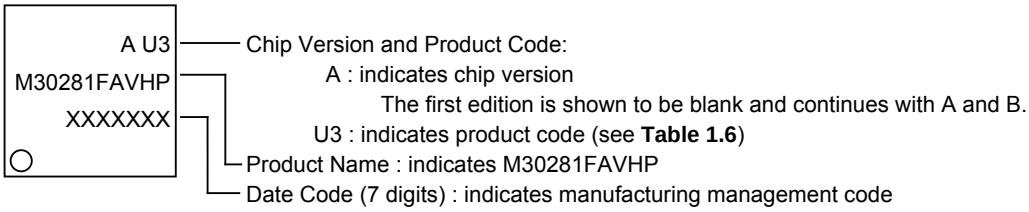
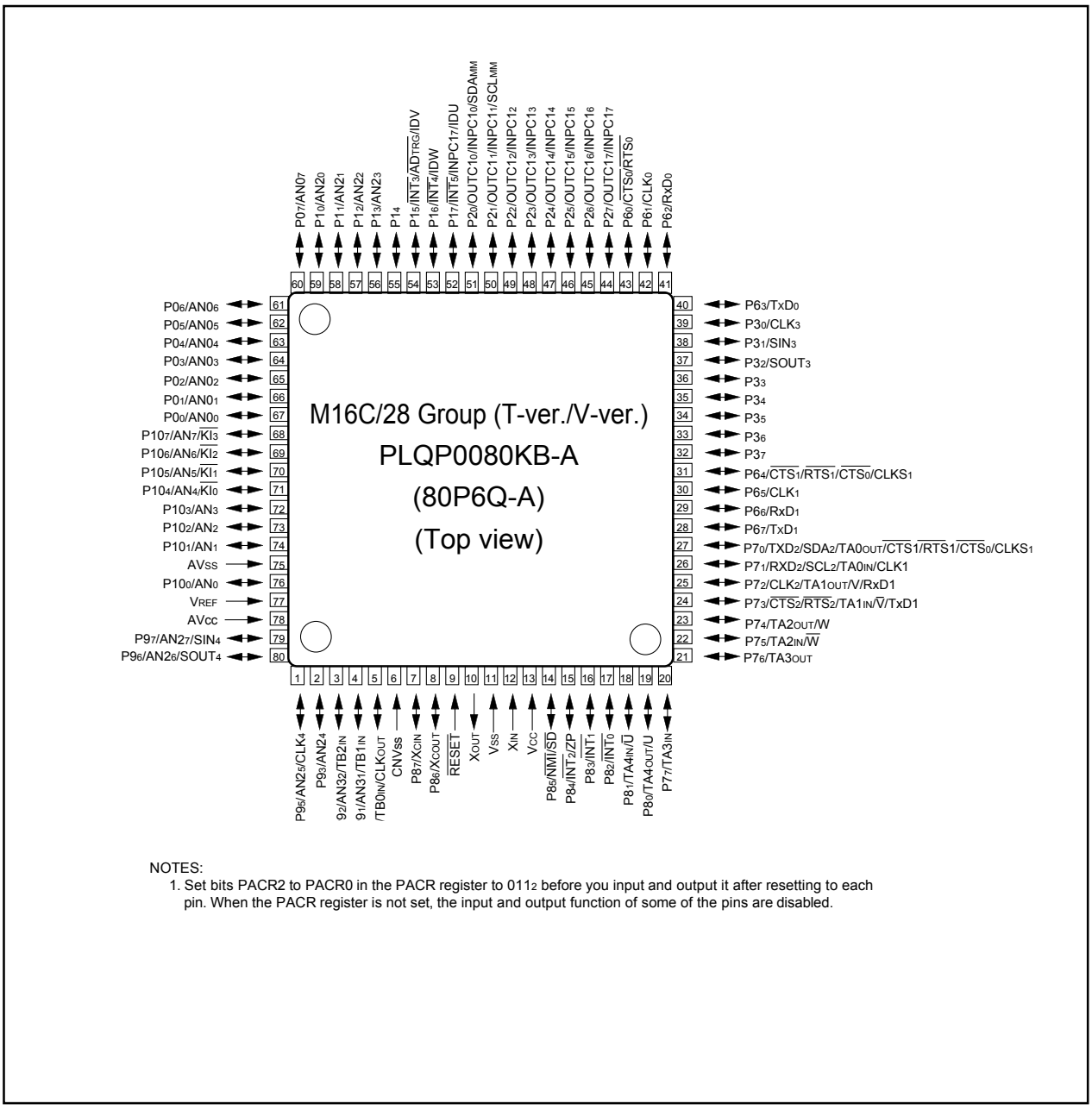


Figure 1.5 Marking Diagram of Flash Memory version - M16C/28 Group V-ver. (Top View)

1.4 Pin Configuration

Figures 1.6 and 1.7 show the pin configurations (top view).



NOTES:
1. Set bits PACR2 to PACR0 in the PACR register to 011_2 before you input and output it after resetting to each pin. When the PACR register is not set, the input and output function of some of the pins are disabled.

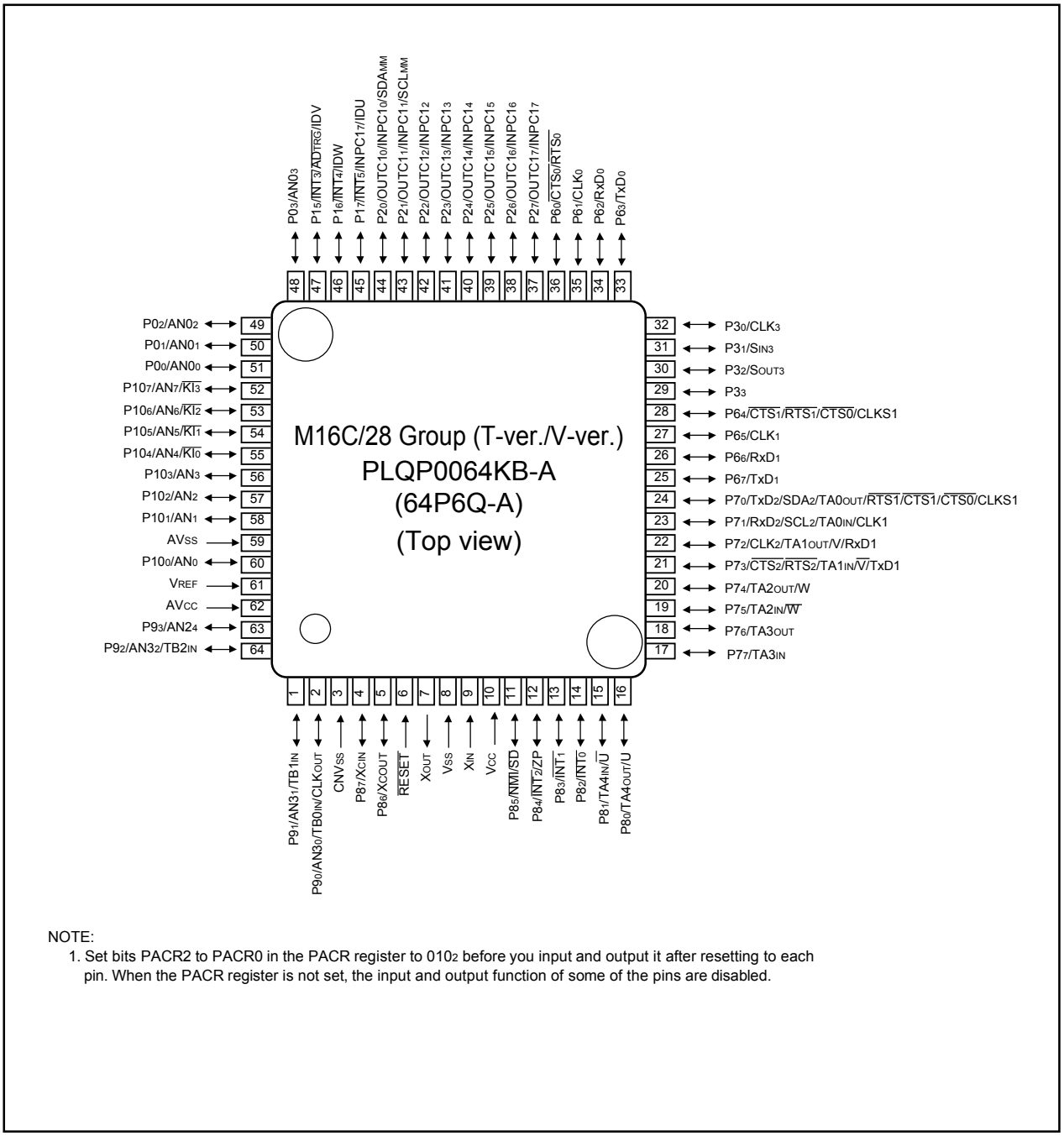
Figure 1.6 Pin Assignment (Top View) of 80-Pin Package

Table 1.9 Pin Characteristics for 80-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	Timer S Pin	UART Pin	Multi-master I^2C bus Pin	Analog Pin
1		$P9_5$				CLK_4		$AN2_5$
2		$P9_3$						$AN2_4$
3		$P9_2$		TB_{2IN}				$AN3_2$
4		$P9_1$		TB_{1IN}				$AN3_1$
5	CLK_{OUT}	$P9_0$		TB_{0IN}				$AN3_0$
6	CNVss							
7	X_{CIN}	$P8_7$						
8	X_{COUT}	$P8_6$						
9	$\overline{RESET}$							
10	X_{OUT}							
11	Vss							
12	X_{IN}							
13	Vcc							
14		$P8_5$	$\overline{NMI}$	$\overline{SD}$				
15		$P8_4$	$\overline{INT_2}$	ZP				
16		$P8_3$	$\overline{INT_1}$					
17		$P8_2$	$\overline{INT_0}$					
18		$P8_1$		TA_{4IN} / $\overline{U}$				
19		$P8_0$		TA_{4OUT} / U				
20		$P7_7$		TA_{3IN}				
21		$P7_6$		TA_{3OUT}				
22		$P7_5$		TA_{2IN} / $\overline{W}$				
23		$P7_4$		TA_{2OUT} / W				
24		$P7_3$		TA_{1IN} / $\overline{V}$		$\overline{CTS_2}$ / $\overline{RTS_2}$ / TxD_1		
25		$P7_2$		TA_{1OUT} / V		CLK_2 / RxD_1		
26		$P7_1$		TA_{0IN}		RxD_2 / SCL_2 / CLK_1		
27		$P7_0$		TA_{0OUT}		TxD_2 / SDA_2 / $\overline{RTS_1}$ / $\overline{CTS_1}$ / $\overline{CTS_0}$ / $CLKS_1$		
28		$P6_7$				TxD_1		
29		$P6_6$				RxD_1		
30		$P6_5$				CLK_1		
31		$P6_4$				$\overline{RTS_1}$ / $\overline{CTS_1}$/ $\overline{CTS_0}$ / $CLKS_1$		
32		$P3_7$						
33		$P3_6$						
34		$P3_5$						
35		$P3_4$						
36		$P3_3$						
37		$P3_2$				$SOUT_3$		
38		$P3_1$				SIN_3		
39		$P3_0$				CLK_3		
40		$P6_3$				TxD_0		

Table 1.9 Pin Characteristics for 80-Pin Package (continued)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	Timer S Pin	UART Pin	Multi-master I²C bus Pin	Analog Pin
41		P6$_2$				RxD$_0$		
42		P6$_1$				CLK$_0$		
43		P6$_0$				$\overline{RTS_0}$ / $\overline{CTS_0}$		
44		P2$_7$			OUTC1$_7$ / INPC1$_7$			
45		P2$_6$			OUTC1$_6$ / INPC1$_6$			
46		P2$_5$			OUTC1$_5$ / INPC1$_5$			
47		P2$_4$			OUTC1$_4$ / INPC1$_4$			
48		P2$_3$			OUTC1$_3$ / INPC1$_3$			
49		P2$_2$			OUTC1$_2$ / INPC1$_2$			
50		P2$_1$			OUTC1$_1$ / INPC1$_1$		SCLMM	
51		P2$_0$			OUTC1$_0$ / INPC1$_0$		SDAMM	
52		P1$_7$	$\overline{INT_5}$	IDU	INPC1$_7$			
53		P1$_6$	$\overline{INT_4}$	IDW				
54		P1$_5$	$\overline{INT_3}$	IDV				$\overline{ADTRG}$
55		P1$_4$						
56		P1$_3$						AN2$_3$
57		P1$_2$						AN2$_2$
58		P1$_1$						AN2$_1$
59		P1$_0$						AN2$_0$
60		P0$_7$						AN0$_7$
61		P0$_6$						AN0$_6$
62		P0$_5$						AN0$_5$
63		P0$_4$						AN0$_4$
64		P0$_3$						AN0$_3$
65		P0$_2$						AN0$_2$
66		P0$_1$						AN0$_1$
67		P0$_0$						AN0$_0$
68		P10$_7$	$\overline{KI_3}$					AN$_7$
69		P10$_6$	$\overline{KI_2}$					AN$_6$
70		P10$_5$	$\overline{KI_1}$					AN$_5$
71		P10$_4$	$\overline{KI_0}$					AN$_4$
72		P10$_3$						AN$_3$
73		P10$_2$						AN$_2$
74		P10$_1$						AN$_1$
75	AVss							
76		P10$_0$						AN$_0$
77	VREF							
78	AVcc							
79		P9$_7$				SIN$_4$		AN2$_7$
80		P9$_6$				SOUT$_4$		AN2$_6$



NOTE:
1. Set bits PACR2 to PACR0 in the PACR register to 010_2 before you input and output it after resetting to each pin. When the PACR register is not set, the input and output function of some of the pins are disabled.

Figure 1.7 Pin Assignment (Top View) of 64-Pin Package

Table 1.10 Pin Characteristics for 64-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	Timer S Pin	UART Pin	Mult-master I²C bus Pin	Analog Pin
1		P9$_1$		TB$_{1IN}$				AN3$_1$
2	CLK$_{OUT}$	P9$_0$		TB$_{0IN}$				AN3$_0$
3	CNVss							
4	X$_{CIN}$	P8$_7$						
5	X$_{COUT}$	P8$_6$						
6	$\overline{\text{RESET}}$							
7	X$_{OUT}$							
8	Vss							
9	X$_{IN}$							
10	Vcc							
11		P8$_5$	$\overline{\text{NMI}}$	$\overline{\text{SD}}$				
12		P8$_4$	$\overline{\text{INT}}_2$	ZP				
13		P8$_3$	$\overline{\text{INT}}_1$					
14		P8$_2$	$\overline{\text{INT}}_0$					
15		P8$_1$		TA$_{4IN}$ / $\overline{\text{U}}$				
16		P8$_0$		TA$_{4OUT}$ / U				
17		P7$_7$		TA$_{3IN}$				
18		P7$_6$		TA$_{3OUT}$				
19		P7$_5$		TA$_{2IN}$ / $\overline{\text{W}}$				
20		P7$_4$		TA$_{2OUT}$ / W				
21		P7$_3$		TA$_{1IN}$ / $\overline{\text{V}}$		$\overline{\text{CTS}}_2$ / $\overline{\text{RTS}}_2$ / TxD$_1$		
22		P7$_2$		TA$_{1OUT}$ / V		CLK$_2$ / RxD$_1$		
23		P7$_1$		TA$_{0IN}$		RxD$_2$ / SCL$_2$ / CLK$_1$		
24		P7$_0$		TA$_{0OUT}$		TxD$_2$ / SDA$_2$ / $\overline{\text{RTS}}_1$ / $\overline{\text{CTS}}_1$ / $\overline{\text{CTS}}_0$ / CLKS$_1$		
25		P6$_7$				TxD$_1$		
26		P6$_6$				RxD$_1$		
27		P6$_5$				CLK$_1$		
28		P6$_4$				$\overline{\text{RTS}}_1$ / $\overline{\text{CTS}}_1$/ $\overline{\text{CTS}}_0$ / CLKS$_1$		
29		P3$_3$						
30		P3$_2$				SOUT$_3$		
31		P3$_1$				SIN$_3$		
32		P3$_0$				CLK$_3$		
33		P6$_3$				TxD$_0$		
34		P6$_2$				RxD$_0$		
35		P6$_1$				CLK$_0$		
36		P6$_0$				$\overline{\text{RTS}}_0$ / $\overline{\text{CTS}}_0$		
37		P2$_7$			OUTC1$_7$ / INPC1$_7$			
38		P2$_6$			OUTC1$_6$ / INPC1$_6$			
39		P2$_5$			OUTC1$_5$ / INPC1$_5$			
40		P2$_4$			OUTC1$_4$ / INPC1$_4$			

Table 1.10 Pin Characteristics for 64-Pin Package (continued)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	Timer S Pin	UART Pin	Multi-master I²C bus Pin	Analog Pin
41		$P2_3$			$OUTC1_3$ / $INPC1_3$			
42		$P2_2$			$OUTC1_2$ / $INPC1_2$			
43		$P2_1$			$OUTC1_1$ / $INPC1_1$		SCL_{MM}	
44		$P2_0$			$OUTC1_0$ / $INPC1_0$		SDA_{MM}	
45		$P1_7$	$\overline{INT}_5$	IDU	$INPC1_7$			
46		$P1_6$	$\overline{INT}_4$	IDW				
47		$P1_5$	$\overline{INT}_3$	IDV				AD_{TRG}
48		$P0_3$						$AN0_3$
49		$P0_2$						$AN0_2$
50		$P0_1$						$AN0_1$
51		$P0_0$						$AN0_0$
52		$P10_7$	$\overline{KI}_3$					AN_7
53		$P10_6$	$\overline{KI}_2$					AN_6
54		$P10_5$	$\overline{KI}_1$					AN_5
55		$P10_4$	$\overline{KI}_0$					AN_4
56		$P10_3$						AN_3
57		$P10_2$						AN_2
58		$P10_1$						AN_1
59	AVss							
60		$P10_0$						AN_0
61	V_{REF}							
62	AVcc							
63		$P9_3$						$AN2_4$
64		$P9_2$		TB_{2IN}				$AN3_2$

1.5 Pin Description

Table 1.11 Pin Description (64-Pin and 80-Pin Packages)

Classification	Symbol	I/O Type	Function
Power supply	VCC, VSS	I	Apply 0V to the Vss pin. Apply following voltage to the Vcc pin. 2.7 to 5.5 V (Normal), 3.0 to 5.5 V (T-ver.), 4.2 to 5.5 V (V-ver.)
Analog power supply	AVCC AVSS	I	Supplies power to the A/D converter. Connect the AVCC pin to VCC and the AVSS pin to VSS
Reset input	$\overline{\text{RESET}}$	I	The microcomputer is in a reset state when "L" is applied to the $\overline{\text{RESET}}$ pin
CNVSS	CNVSS	I	Connect the CNVSS pin to VSS
Main clock input	XIN	I	I/O pins for the main clock oscillation circuit. Connect a ceramic resonator or crystal oscillator between XIN and XOUT. To apply external clock, apply it to XIN and leave XOUT open. If XIN is not used (for external oscillator or external clock) connect XIN pin to VCC and leave XOUT open
Main clock output	XOUT	O	
Sub clock input	XCIN	I	I/O pins for the sub clock oscillation circuit. Connect a crystal oscillator between XCIN and XCOUT
Sub clock output	XCOUT	O	
Clock output	CLKOUT	O	Outputs the clock having the same frequency as f1, f8, f32, or fC
$\overline{\text{INT}}$ interrupt input	$\overline{\text{INT0}}$ to $\overline{\text{INT5}}$	I	Input pins for the $\overline{\text{INT}}$ interrupt. $\overline{\text{INT2}}$ can be used for Timer A Z-phase function
$\overline{\text{NMI}}$ interrupt input	$\overline{\text{NMI}}$	I	Input pin for the $\overline{\text{NMI}}$ interrupt. $\overline{\text{NMI}}$ cannot be used as I/O port while the three-phase motor control is enabled. Apply a stable "H" to $\overline{\text{NMI}}$ after setting it's direction register to "0" when the three-phase motor control is enabled
Key input interrupt	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	Input pins for the key input interrupt
Timer A	TA0OUT to TA4OUT	I/O	I/O pins for the timer A0 to A4
	TA0IN to TA4IN	I	Input pins for the timer A0 to A4
	ZP	I	Input pin for Z-phase
Timer B	TB0IN to TB2IN	I	Input pins for the timer B0 to B2
Three-phase motor control timer output	U, $\overline{\text{U}}$, V, $\overline{\text{V}}$, W, $\overline{\text{W}}$	O	Output pins for the three-phase motor control timer
	IDU, IDW, IDV, $\overline{\text{SD}}$	I/O	Input and output pins for the three-phase motor control timer
Serial I/O	$\overline{\text{CTS0}}$ to $\overline{\text{CTS2}}$	I	Input pins for data transmission control
	$\overline{\text{RTS0}}$ to $\overline{\text{RTS2}}$	O	Output pins for data reception control
	CLK0 to CLK3	I/O	Inputs and outputs the transfer clock
	RxD0 to RxD2	I	Inputs serial data
	SIN3	I	Inputs serial data
	TxD0 to TxD2	O	Outputs serial data
	SOUT3	O	Outputs serial data
	CLKS1	O	Output pin for transfer clock
I^2C bus Mode	SDA2	I/O	Inputs and outputs serial data
	SCL2		Inputs and outputs the transfer clock
Multi-master I^2C bus	SDAMM	I/O	Inputs and outputs serial data
	SCLMM		Inputs and outputs the transfer clock
Reference voltage input	VREF	I	Applies reference voltage to the A/D converter
A/D converter	AN0 to AN7 AN00 to AN03 AN24 AN30 to AN32	I	Analog input pins for the A/D converter
	$\overline{\text{ADTRG}}$	I	Input pin for an external A/D trigger

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

Table 1.11 Pin Description (64-Pin and 80-Pin Packages) (Continued)

Classification	Symbol	I/O Type	Function
Timer S	$INPC1_0$ to $INPC1_7$	I	Input pins for the time measurement function
	$OUTC1_0$ to $OUTC1_7$	O	Output pins for the waveform generating function
I/O Ports	$P0_0$ to $P0_3$ $P1_5$ to $P1_7$ $P2_0$ to $P2_7$ $P3_0$ to $P3_3$ $P6_0$ to $P6_7$ $P7_0$ to $P7_7$ $P8_0$ to $P8_7$ $P9_0$ to $P9_3$ $P10_0$ to $P10_7$	I/O	CMOS I/O ports which have a direction register determines an individual pin is used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports.

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

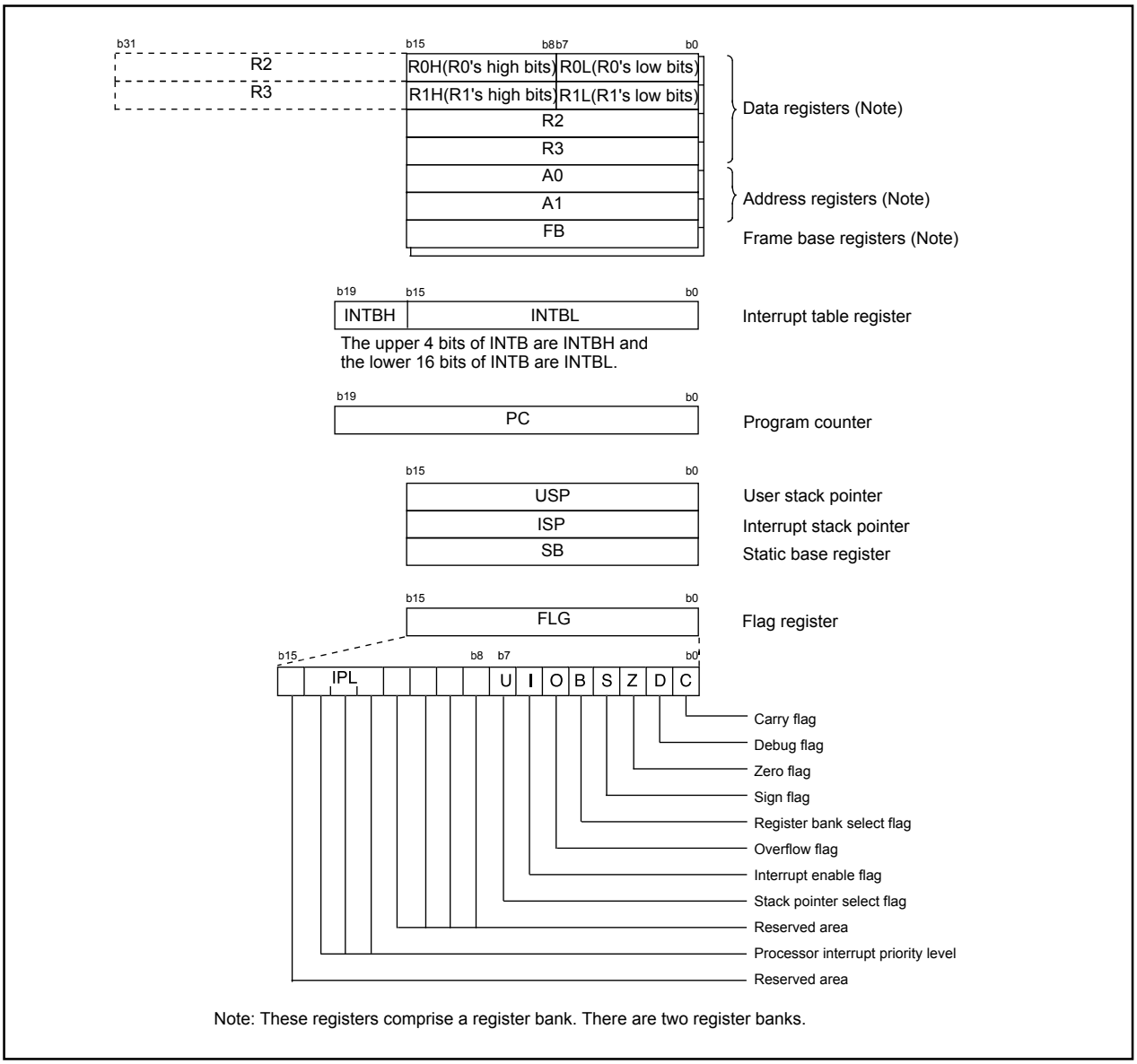
Table 1.12 Pin Description (80-Pin Packages Only) (Continued)

Classification	Symbol	I/O Type	Function
Serial I/O	CLK4	I/O	Inputs and outputs the transfer clock
	SIN4	I	Inputs serial data
	SOUT4	O	Outputs serial data
A/D Converter	AN04 to AN07 AN20 to AN23 AN25 to AN27	I	Analog input pins for the A/D converter
I/O Ports	P04 to P07 P10 to P14 P34 to P37 P95 to P97	I/O	CMOS I/O ports which have a direction register determines an individual pin is used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports.

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

2. Central Processing Unit (CPU)

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



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

Figure 2.1. Central Processing Unit Register

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

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

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

2.2 Address Registers (A0 and A1)

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

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

2.3 Frame Base Register (FB)

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

2.4 Interrupt Table Register (INTB)

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

2.5 Program Counter (PC)

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

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

Stack pointer (SP) comes in two types: USP and ISP, each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by the U flag of FLG.

2.7 Static Base Register (SB)

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

2.8 Flag Register (FLG)

FLG consists of 11 bits, indicating the CPU status.

2.8.1 Carry Flag (C Flag)

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

2.8.2 Debug Flag (D Flag)

The D flag is used exclusively for debugging purpose. During normal use, it must be set to 0.

2.8.3 Zero Flag (Z Flag)

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

2.8.4 Sign Flag (S Flag)

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

2.8.5 Register Bank Select Flag (B Flag)

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

2.8.6 Overflow Flag (O Flag)

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

2.8.7 Interrupt Enable Flag (I Flag)

This flag enables a maskable interrupt.

Maskable interrupts are disabled when the I flag is 0, and are enabled when the I flag is 1.

The I flag is cleared to 0 when the interrupt request is accepted.

2.8.8 Stack Pointer Select Flag (U Flag)

ISP is selected when the U flag is 0; USP is selected when the U flag is 1.

The U flag is cleared to 0 when a hardware interrupt request is accepted or an INT instruction for software interrupt Nos. 0 to 31 is executed.

2.8.9 Processor Interrupt Priority Level (IPL)

IPL is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has priority greater than IPL, the interrupt is enabled.

2.8.10 Reserved Area

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

3. Memory

Figure 3.1 is a memory map of the M16C/28 group. M16C/28 group provides 1-Mbyte address space from addresses 00000$_{16}$ to FFFFF$_{16}$. The internal ROM is allocated lower addresses beginning with address FFFFF$_{16}$. For example, 64 Kbytes internal ROM is allocated addresses F0000$_{16}$ to FFFFF$_{16}$.

Two 2-Kbyte internal ROM areas, block A and block B, are available in the flash memory version. The blocks are allocated addresses F000$_{16}$ to FFFF$_{16}$.

The fixed interrupt vector tables are allocated addresses FFFDC$_{16}$ to FFFFF$_{16}$. It stores the starting address of each interrupt routine. See the section on interrupts for details.

The internal RAM is allocated higher addresses beginning with address 00400$_{16}$. For example, 4-Kbytes internal RAM is allocated addresses 00400$_{16}$ to 013FF$_{16}$. Besides sotring data, it becomes stacks when the subroutines is called or an interrupt is acknowledged.

SFR, consisting of control registers for peripheral functions such as I/O port, A/D converter, serial I/O, timers is allocated addresses 00000$_{16}$ to 003FF$_{16}$. All blank spaces within SFR are reserved and cannot be accessed by users.

The special page vector table is allocated to the addresses FFE00$_{16}$ to FFFDB$_{16}$. This vector is used by the JMPS or JSRS instruction. For details, refer to the ***M16C/60 and M16C/20 Series Software Manual***.

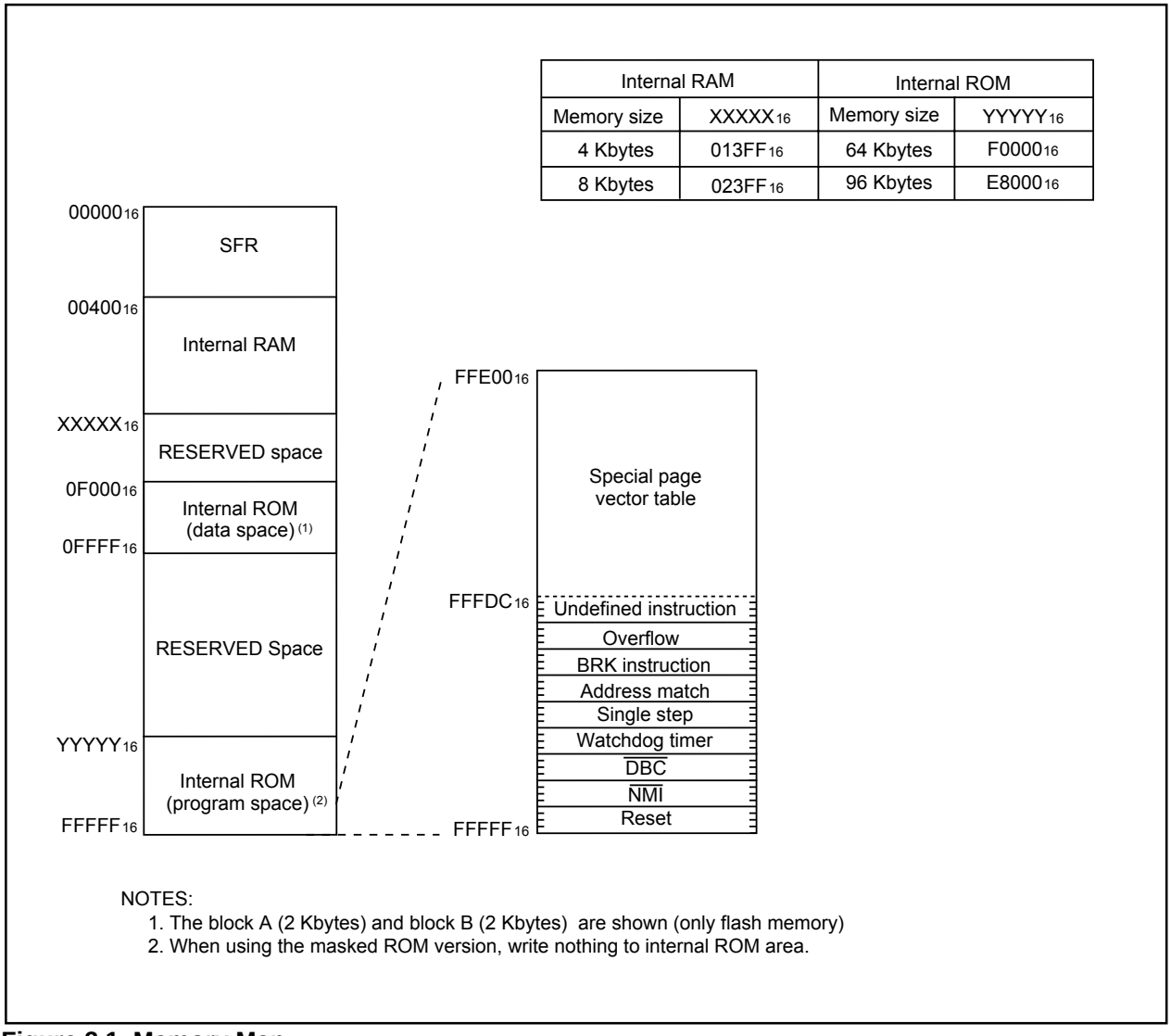


Figure 3.1 Memory Map

4. Special Function Register (SFRs)

SFRs (Special Function Registers) are the control registers of peripheral functions. Table 4.1 to 4.7 list the SFR address map.

Table 4.1 SFR Information(1)(1)

Address	Register	Symbol	After Reset
0000₁₆			
0001₁₆			
0002₁₆			
0003₁₆			
0004₁₆	Processor mode register 0	PM0	00₁₆
0005₁₆	Processor mode register 1	PM1	00001000₂
0006₁₆	System clock control register 0	CM0	01001000₂
0007₁₆	System clock control register 1	CM1	00100000₂
0008₁₆			
0009₁₆	Address match interrupt enable register	AIER	XXXXXX00₂
000A₁₆	Protect register	PRCR	XX000000₂
000B₁₆			
000C₁₆	Oscillation stop detection register (2)	CM2	0X000010₂
000D₁₆			
000E₁₆	Watchdog timer start register	WDTS	XX₁₆
000F₁₆	Watchdog timer control register	WDC	00XXXXXX₂
0010₁₆	Address match interrupt register 0	RMAD0	00₁₆
0011₁₆			00₁₆
0012₁₆			X0₁₆
0013₁₆			
0014₁₆	Address match interrupt register 1	RMAD1	00₁₆
0015₁₆			00₁₆
0016₁₆			X0₁₆
0017₁₆			
0018₁₆			
0019₁₆			
001A₁₆			
001B₁₆			
001C₁₆	PLL control register 0	PLC0	0001X010₂
001D₁₆			
001E₁₆	Processor mode register 2	PM2	XXX00000₂
001F₁₆			
0020₁₆	DMA0 source pointer	SAR0	XX₁₆
0021₁₆			XX₁₆
0022₁₆			XX₁₆
0023₁₆			
0024₁₆	DMA0 destination pointer	DAR0	XX₁₆
0025₁₆			XX₁₆
0026₁₆			X/₁₆
0027₁₆			
0028₁₆	DMA0 transfer counter	TCR0	XX₁₆
0029₁₆			XX₁₆
002A₁₆			
002B₁₆			
002C₁₆	DMA0 control register	DM0CON	00000X00₂
002D₁₆			
002E₁₆			
002F₁₆			
0030₁₆	DMA1 source pointer	SAR1	XX₁₆
0031₁₆			XX₁₆
0032₁₆			XX₁₆
0033₁₆			
0034₁₆	DMA1 destination pointer	DAR1	XX₁₆
0035₁₆			XX₁₆
0036₁₆			XX₁₆
0037₁₆			
0038₁₆	DMA1 transfer counter	TCR1	XX₁₆
0039₁₆			XX₁₆
003A₁₆			
003B₁₆			
003C₁₆	DMA1 control register	DM1CON	00000X00₂
003D₁₆			
003E₁₆			
003F₁₆			

NOTES:
1. The blank areas are reserved and cannot be used by users.
2. The CM20, CM21, and CM27 bits do not change at oscillation stop detection reset.

X : Undefined

Table 4.2 SFR Information(2)(1)

Address	Register	Symbol	After Reset
0040₁₆			
0041₁₆			
0042₁₆			
0043₁₆			
0044₁₆	INT3 interrupt control register	INT3IC	XX00X000₂
0045₁₆	IC/OC 0 interrupt control register	ICOC0IC	XXXXX000₂
0046₁₆	IC/OC 1 interrupt control register, I²C bus interface interrupt control register	ICOC1IC, IICIC	XXXXX000₂
0047₁₆	IC/OC base timer interrupt control register, SCLSDA interrupt control register	BTIC, SCLDAIC	XXXXX000₂
0048₁₆	SI/O4 interrupt control register, INT5 interrupt control register	S4IC, INT5IC	XX00X000₂
0049₁₆	SI/O3 interrupt control register, INT4 interrupt control register	S3IC, INT4IC	XX00X000₂
004A₁₆	UART2 Bus collision detection interrupt control register	BCNIC	XXXXX000₂
004B₁₆	DMA0 interrupt control register	DM0IC	XXXXX000₂
004C₁₆	DMA1 interrupt control register	DM1IC	XXXXX000₂
004D₁₆	Key input interrupt control register	KUPIC	XXXXX000₂
004E₁₆	A/D conversion interrupt control register	ADIC	XXXXX000₂
004F₁₆	UART2 transmit interrupt control register	S2TIC	XXXXX000₂
0050₁₆	UART2 receive interrupt control register	S2RIC	XXXXX000₂
0051₁₆	UART0 transmit interrupt control register	S0TIC	XXXXX000₂
0052₁₆	UART0 receive interrupt control register	S0RIC	XXXXX000₂
0053₁₆	UART1 transmit interrupt control register	S1TIC	XXXXX000₂
0054₁₆	UART1 receive interrupt control register	S1RIC	XXXXX000₂
0055₁₆	Timer A0 interrupt control register	TA0IC	XXXXX000₂
0056₁₆	Timer A1 interrupt control register	TA1IC	XXXXX000₂
0057₁₆	Timer A2 interrupt control register	TA2IC	XXXXX000₂
0058₁₆	Timer A3 interrupt control register	TA3IC	XXXXX000₂
0059₁₆	Timer A4 interrupt control register	TA4IC	XXXXX000₂
005A₁₆	Timer B0 interrupt control register	TB0IC	XXXXX000₂
005B₁₆	Timer B1 interrupt control register	TB1IC	XXXXX000₂
005C₁₆	Timer B2 interrupt control register	TB2IC	XXXXX000₂
005D₁₆	INT0 interrupt control register	INT0IC	XX00X000₂
005E₁₆	INT1 interrupt control register	INT1IC	XX00X000₂
005F₁₆	INT2 interrupt control register	INT2IC	XX00X000₂
0060₁₆			
0061₁₆			
0062₁₆			
0063₁₆			
0064₁₆			
0065₁₆			
0066₁₆			
0067₁₆			
0068₁₆			
0069₁₆			
006A₁₆			
006B₁₆			
006C₁₆			
006D₁₆			
006E₁₆			
006F₁₆			
0070₁₆			
0071₁₆			
0072₁₆			
0073₁₆			
0074₁₆			
0075₁₆			
0076₁₆			
0077₁₆			
0078₁₆			
0079₁₆			
007A₁₆			
007B₁₆			
007C₁₆			
007D₁₆			
007E₁₆			
007F₁₆			

Note 1: The blank spaces are reserved. No access is allowed.

X : Undefined

Table 4.3 SFR Information(3)(1)

Address	Register	Symbol	After Reset
≈			≈
01B016			
01B116			
01B216			
01B316	Flash memory control register 4 (2)	FMR4	010000002
01B416			
01B516	Flash memory control register 1 (2)	FMR1	000XXX0X2
01B616			
01B716	Flash memory control register 0 (2)	FMR0	000000012
01B816			
01B916			
≈			≈
021016	Low-power Consumption Control 0	LPCC0	X000000012
021116			
021216			
021316			
021416			
021516			
021616			
021716			
021816			
021916			
≈			≈
025016			
025116			
025216			
025316			
025416			
025516			
025616			
025716			
025816			
025916			
025A16	Three-phase protect control register	TPRC	0016
025B16			
025C16	On-chip oscillator control register	ROCR	X000010012
025D16	Pin assignment control register	PACR	0016
025E16	Peripheral clock select register	PCLKR	000000112
025F16	Low-power Consumption Control 1	LPCC1	0016
≈			≈
02E016	I^2C0 data shift register	S00	XX16
02E116			
02E216	I^2C0 address register	S0D0	0016
02E316	I^2C0 control register 0	S1D0	0016
02E416	I^2C0 clock control register	S20	0016
02E516	I^2C0 start/stop condition control register	S2D0	000110102
02E616	I^2C0 control register 1	S3D0	001100002
02E716	I^2C0 control register 2	S4D0	0016
02E816	I^2C0 status register	S10	0001000X2
02E916			
02EA16			
≈			≈
02FE16			
02FF16			

Note 1:The blank spaces are reserved. No access is allowed.
Note 2:This register is included in the flash memory version.

X : Undefined

Table 4.4 SFR Information(4)(1)

Address	Register	Symbol	After Reset
0300₁₆	Time measurement, Pulse generation register 0	G1TM0,G1PO0	XX₁₆
0301₁₆			XX₁₆
0302₁₆	Time measurement, Pulse generation register 1	G1TM1,G1PO1	XX₁₆
0303₁₆			XX₁₆
0304₁₆	Time measurement, Pulse generation register 2	G1TM2,G1PO2	XX₁₆
0305₁₆			XX₁₆
0306₁₆	Time measurement, Pulse generation register 3	G1TM3,G1PO3	XX₁₆
0307₁₆			XX₁₆
0308₁₆	Time measurement, Pulse generation register 4	G1TM4,G1PO4	XX₁₆
0309₁₆			XX₁₆
030A₁₆	Time measurement, Pulse generation register 5	G1TM5,G1PO5	XX₁₆
030B₁₆			XX₁₆
030C₁₆	Time measurement, Pulse generation register 6	G1TM6,G1PO6	XX₁₆
030D₁₆			XX₁₆
030E₁₆	Time measurement, Pulse generation register 7	G1TM7,G1PO7	XX₁₆
030F₁₆			XX₁₆
0310₁₆	Pulse generation control register 0	G1POCR0	0X00XX00₂
0311₁₆	Pulse generation control register 1	G1POCR1	0X00XX00₂
0312₁₆	Pulse generation control register 2	G1POCR2	0X00XX00₂
0313₁₆	Pulse generation control register 3	G1POCR3	0X00XX00₂
0314₁₆	Pulse generation control register 4	G1POCR4	0X00XX00₂
0315₁₆	Pulse generation control register 5	G1POCR5	0X00XX00₂
0316₁₆	Pulse generation control register 6	G1POCR6	0X00XX00₂
0317₁₆	Pulse generation control register 7	G1POCR7	0X00XX00₂
0318₁₆	Time measurement control register 0	G1TMCR0	00₁₆
0319₁₆	Time measurement control register 1	G1TMCR1	00₁₆
031A₁₆	Time measurement control register 2	G1TMCR2	00₁₆
031B₁₆	Time measurement control register 3	G1TMCR3	00₁₆
031C₁₆	Time measurement control register 4	G1TMCR4	00₁₆
031D₁₆	Time measurement control register 5	G1TMCR5	00₁₆
031E₁₆	Time measurement control register 6	G1TMCR6	00₁₆
031F₁₆	Time measurement control register 7	G1TMCR7	00₁₆
0320₁₆	Base timer register	G1BT	XX₁₆
0321₁₆			XX₁₆
0322₁₆	Base timer control register 0	G1BCR0	00₁₆
0323₁₆	Base timer control register 1	G1BCR1	00₁₆
0324₁₆	Time measurement prescale register 6	G1TPR6	00₁₆
0325₁₆	Time measurement prescale register 7	G1TPR7	00₁₆
0326₁₆	Function enable register	G1FE	00₁₆
0327₁₆	Function select register	G1FS	00₁₆
0328₁₆	Base timer reset register	G1BTRR	XX₁₆
0329₁₆			XX₁₆
032A₁₆	Count source division register	G1DV	00₁₆
032B₁₆			
032C₁₆			
032D₁₆			
032E₁₆			
032F₁₆			
0330₁₆	Interrupt request register	G1IR	XX₁₆
0331₁₆	Interrupt enable register 0	G1IE0	00₁₆
0332₁₆	Interrupt enable register 1	G1IE1	00₁₆
0333₁₆			
0334₁₆			
0335₁₆			
0336₁₆			
0337₁₆			
0338₁₆			
0339₁₆			
033A₁₆			
033B₁₆			
033C₁₆			
033D₁₆			
033E₁₆	NMI digital debounce register	NDDR	FF₁₆
033F₁₆	Port P17 digital debounce register	P17DDR	FF₁₆

NOTE:
1. The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.5 SFR Information(5)(1)

Address	Register	Symbol	After Reset
0340₁₆			
0341₁₆			
0342₁₆	Timer A1-1 register	TA11	XX₁₆
0343₁₆			XX₁₆
0344₁₆	Timer A2-1 register	TA21	XX₁₆
0345₁₆			XX₁₆
0346₁₆	Timer A4-1 register	TA41	XX₁₆
0347₁₆			XX₁₆
0348₁₆	Three phase PWM control register 0	INVC0	00₁₆
0349₁₆	Three phase PWM control register 1	INVC1	00₁₆
034A₁₆	Three phase output buffer register 0	IDB0	00111111₂
034B₁₆	Three phase output buffer register 1	IDB1	00111111₂
034C₁₆	Dead time timer	DTT	XX₁₆
034D₁₆	Timer B2 Interrupt occurrence frequency set counter	ICTB2	XX₁₆
034E₁₆	Position - data - retain function control register	PDRF	XXXX0000₂
034F₁₆			
0350₁₆			
0351₁₆			
0352₁₆			
0353₁₆			
0354₁₆			
0355₁₆			
0356₁₆			
0357₁₆			
0358₁₆	Port function control register	PFCR	00111111₂
0359₁₆			
035A₁₆			
035B₁₆			
035C₁₆			
035D₁₆			
035E₁₆	Interrupt cause select register 2	IFSR2A	00XXXXX0₂(2)
035F₁₆	Interrupt cause select register	IFSR	00₁₆
0360₁₆	SI/O3 transmit/receive register	S3TRR	XX₁₆
0361₁₆			
0362₁₆	SI/O3 control register	S3C	01000000₂
0363₁₆	SI/O3 bit rate register	S3BRG	XX₁₆
0364₁₆	SI/O4 transmit/receive register	S4TRR	XX₁₆
0365₁₆			
0366₁₆	SI/O4 control register	S4C	01000000₂
0367₁₆	SI/O4 bit rate register	S4BRG	XX₁₆
0368₁₆			
0369₁₆			
036A₁₆			
036B₁₆			
036C₁₆			
036D₁₆			
036E₁₆			
036F₁₆			
0370₁₆			
0371₁₆			
0372₁₆			
0373₁₆			
0374₁₆	UART2 special mode register 4	U2SMR4	00₁₆
0375₁₆	UART2 special mode register 3	U2SMR3	000X0X0X₂
0376₁₆	UART2 special mode register 2	U2SMR2	X0000000₂
0377₁₆	UART2 special mode register	U2SMR	X0000000₂
0378₁₆	UART2 transmit/receive mode register	U2MR	00₁₆
0379₁₆	UART2 bit rate register	U2BRG	XX₁₆
037A₁₆	UART2 transmit buffer register	U2TB	XX₁₆
037B₁₆			XX₁₆
037C₁₆	UART2 transmit/receive control register 0	U2C0	00001000₂
037D₁₆	UART2 transmit/receive control register 1	U2C1	00000010₂
037E₁₆	UART2 receive buffer register	U2RB	XX₁₆
037F₁₆			XX₁₆

NOTES:
1. The blank areas are reserved and cannot be used by users.
2. Write "1" to bit 0 after reset.

X : Undefined

Table 4.6 SFR Information(6)(1)

Address	Register	Symbol	After Reset
0380₁₆	Count start flag	TABSR	00₁₆
0381₁₆	Clock prescaler reset flag	CPSRF	0XXXXXXX₂
0382₁₆	One-shot start flag	ONSF	00₁₆
0383₁₆	Trigger select register	TRGSR	00₁₆
0384₁₆	Up-dowm flag	UDF	00₁₆
0385₁₆			
0386₁₆ 0387₁₆	Timer A0 register	TA0	XX₁₆ XX₁₆
0388₁₆ 0389₁₆	Timer A1 register	TA1	XX₁₆ XX₁₆
038A₁₆ 038B₁₆	Timer A2 register	TA2	XX₁₆ XX₁₆
038C₁₆ 038D₁₆	Timer A3 register	TA3	XX₁₆ XX₁₆
038E₁₆ 038F₁₆	Timer A4 register	TA4	XX₁₆ XX₁₆
0390₁₆ 0391₁₆	Timer B0 register	TB0	XX₁₆ XX₁₆
0392₁₆ 0393₁₆	Timer B1 register	TB1	XX₁₆ XX₁₆
0394₁₆ 0395₁₆	Timer B2 register	TB2	XX₁₆ XX₁₆
0396₁₆	Timer A0 mode register	TA0MR	00₁₆
0397₁₆	Timer A1 mode register	TA1MR	00₁₆
0398₁₆	Timer A2 mode register	TA2MR	00₁₆
0399₁₆	Timer A3 mode register	TA3MR	00₁₆
039A₁₆	Timer A4 mode register	TA4MR	00₁₆
039B₁₆	Timer B0 mode register	TB0MR	00XX0000₂
039C₁₆	Timer B1 mode register	TB1MR	00XX0000₂
039D₁₆	Timer B2 mode register	TB2MR	00XX0000₂
039E₁₆	Timer B2 special mode register	TB2SC	X0000000₂
039F₁₆			
03A0₁₆	UART0 transmit/receive mode register	U0MR	00₁₆
03A1₁₆	UART0 bit rate register	U0BRG	XX₁₆
03A2₁₆ 03A3₁₆	UART0 transmit buffer register	U0TB	XX₁₆ XX₁₆
03A4₁₆	UART0 transmit/receive control register 0	U0C0	00001000₂
03A5₁₆	UART0 transmit/receive control register 1	U0C1	00000010₂
03A6₁₆ 03A7₁₆	UART0 receive buffer register	U0RB	XX₁₆ XX₁₆
03A8₁₆	UART1 transmit/receive mode register	U1MR	00₁₆
03A9₁₆	UART1 bit rate register	U1BRG	XX₁₆
03AA₁₆ 03AB₁₆	UART1 transmit buffer register	U1TB	XX₁₆ XX₁₆
03AC₁₆	UART1 transmit/receive control register 0	U1C0	00001000₂
03AD₁₆	UART1 transmit/receive control register 1	U1C1	00000010₂
03AE₁₆ 03AF₁₆	UART1 receive buffer register	U1RB	XX₁₆ XX₁₆
03B0₁₆	UART transmit/receive control register 2	UCON	X0000000₂
03B1₁₆			
03B2₁₆			
03B3₁₆			
03B4₁₆ 03B5₁₆	SFR snoop address register	CRCSAR	XX₁₆ 00XXXXXX₂
03B6₁₆	CRC mode register	CRCMR	0XXXXXX0₂
03B7₁₆			
03B8₁₆	DMA0 request cause select register	DM0SL	00₁₆
03B9₁₆			
03BA₁₆	DMA1 request cause select register	DM1SL	00₁₆
03BB₁₆			
03BC₁₆ 03BD₁₆	CRC data register	CRCD	XX₁₆ XX₁₆
03BE₁₆	CRC input register	CRCIN	XX₁₆
03BF₁₆			

NOTE:
1. The blank areas are reserved and cannot be used by users.

X : Undefined

Table 4.7 SFR Information(7)(1)

Address	Register	Symbol	After Reset
03C0₁₆	A/D register 0	AD0	XX₁₆
03C1₁₆			XX₁₆
03C2₁₆	A/D register 1	AD1	XX₁₆
03C3₁₆			XX₁₆
03C4₁₆	A/D register 2	AD2	XX₁₆
03C5₁₆			XX₁₆
03C6₁₆	A/D register 3	AD3	XX₁₆
03C7₁₆			XX₁₆
03C8₁₆	A/D register 4	AD4	XX₁₆
03C9₁₆			XX₁₆
03CA₁₆	A/D register 5	AD5	XX₁₆
03CB₁₆			XX₁₆
03CC₁₆	A/D register 6	AD6	XX₁₆
03CD₁₆			XX₁₆
03CE₁₆	A/D register 7	AD7	XX₁₆
03CF₁₆			XX₁₆
03D0₁₆			
03D1₁₆			
03D2₁₆	A/D trigger control register	ADTRGCON	00₁₆
03D3₁₆	A/D status register 0	ADSTAT0	00000X00₂
03D4₁₆	A/D control register 2	ADCON2	00₁₆
03D5₁₆			
03D6₁₆	A/D control register 0	ADCON0	00000XXX₂
03D7₁₆	A/D control register 1	ADCON1	00₁₆
03D8₁₆			
03D9₁₆			
03DA₁₆			
03DB₁₆			
03DC₁₆			
03DD₁₆			
03DE₁₆			
03DF₁₆			
03E0₁₆	Port P0 register	P0	XX₁₆
03E1₁₆	Port P1 register	P1	XX₁₆
03E2₁₆	Port P0 direction register	PD0	00₁₆
03E3₁₆	Port P1 direction register	PD1	00₁₆
03E4₁₆	Port P2 register	P2	XX₁₆
03E5₁₆	Port P3 register	P3	XX₁₆
03E6₁₆	Port P2 direction register	PD2	00₁₆
03E7₁₆	Port P3 direction register	PD3	00₁₆
03E8₁₆			
03E9₁₆			
03EA₁₆			
03EB₁₆			
03EC₁₆	Port P6 register	P6	XX₁₆
03ED₁₆	Port P7 register	P7	XX₁₆
03EE₁₆	Port P6 direction register	PD6	00₁₆
03EF₁₆	Port P7 direction register	PD7	00₁₆
03F0₁₆	Port P8 register	P8	XX₁₆
03F1₁₆	Port P9 register	P9	XX₁₆
03F2₁₆	Port P8 direction register	PD8	00₁₆
03F3₁₆	Port P9 direction register	PD9	000X0000₂
03F4₁₆	Port P10 register	P10	XX₁₆
03F5₁₆			
03F6₁₆	Port P10 direction register	PD10	00₁₆
03F7₁₆			
03F8₁₆			
03F9₁₆			
03FA₁₆			
03FB₁₆			
03FC₁₆	Pull-up control register 0	PUR0	00₁₆
03FD₁₆	Pull-up control register 1	PUR1	00₁₆
03FE₁₆	Pull-up control register 2	PUR2	00₁₆
03FF₁₆	Port control register	PCR	00₁₆

NOTE:
1. The blank areas are reserved and cannot be used by users.

X : Undefined

5. Reset

Hardware reset 1, software reset, watchdog timer reset, and oscillation stop detection reset are implemented to reset the MCU.

5.1 Hardware Reset

5.1.1 Hardware Reset 1

Pins, CPU, and SFRs are reset by using the $\overline{\text{RESET}}$ pin. When a low-level (“L”) signal is applied to the $\overline{\text{RESET}}$ pin while the supply voltage meets the recommended operating condition, pins, CPU, and SFRs are reset (see **Table 5.1** Pin Status When $\overline{\text{RESET}}$ Pin Level is “L”). The oscillation circuit is also reset and the on-chip oscillator starts oscillating as the CPU clock. CPU and SFRs re reset when the signal applied to the $\overline{\text{RESET}}$ pin changes from “L” to high (“H”). The MCU executes a program beginning with the address indicated by the reset vector. The internal RAM is not reset. When an “L” signal is applied to the $\overline{\text{RESET}}$ pin while writing data to the internal RAM, the content of internal RAM is undefined.
Figure 5.1 shows an example of the reset circuit. **Figure 5.2** shows a reset sequence. **Table 5.1** shows status of the other pins while the $\overline{\text{RESET}}$ pin is held “L”. **Figure 5.3** shows CPU register states after reset. Refer to **4. Special Function Register (SFR)** about SFR states after reset.

1. Reset on a stable supply voltage
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin
(2) Wait *td(ROC)* or more
(3) Apply an “H” signal to the $\overline{\text{RESET}}$ pin

2. Power-on reset
(1) Apply an “L” signal to the $\overline{\text{RESET}}$ pin
(2) Increase the supply voltage until it meets the the recommended performance condition
(3) Wait for *td(P-R)* or more to allow the internal power supply to stabilize
(4) Wait *td(ROC)* or more
(5) Apply an “H” signal to the $\overline{\text{RESET}}$ pin

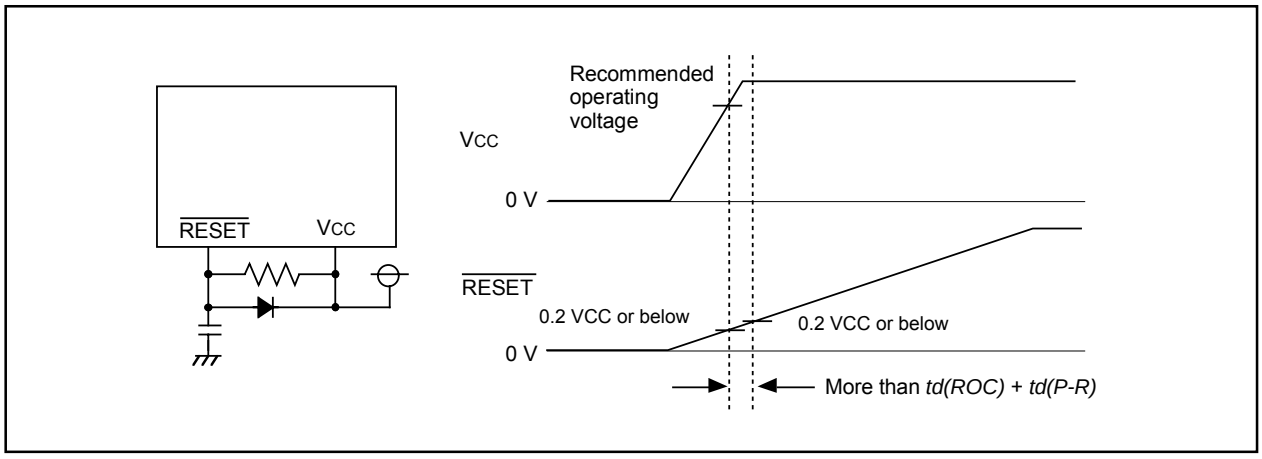


Figure 5.1 Example Reset Circuit

5.2 Software Reset

The MCU resets its pins, CPU, and SFRs when the PM03 bit in the PM0 register is set to 1 (reset) and the MCU executes a program in an address indicated by the reset vector. Then the on-chip oscillator is selected as the CPU clock.

The software reset does not reset some portions of the SFRs. Refer to **4. Special Function Registers (SFRs)** for details.

5.3 Watchdog Timer Reset

The MCU resets its pins, CPU, and SFRs when the PM12 bit in the PM1 register is set to 1 (watchdog timer reset) and the watchdog timer underflows. The MCU executes a program in an address indicated by the reset vector. Then the on-chip oscillator is selected as the CPU clock.

The watchdog timer reset does not reset some portions of the SFRs. Refer to **4. Special Function Registers (SFRs)** for details.

5.4 Oscillation Stop Detection Reset

The MCU resets its pins, CPU, and SFRs and stops if the main clock stop is detected when the CM20 bit in the CM2 register is set to 1 (oscillation stop, re-oscillation detection function enabled) and the CM27 bit in the CM2 register is 0 (reset at oscillation stop detection). Refer to the section **7.8 oscillation stop, re-oscillation detection function** for details.

The oscillation stop detection reset does not reset some portions of the SFRs. Refer to **4. Special Function Registers (SFRs)**.

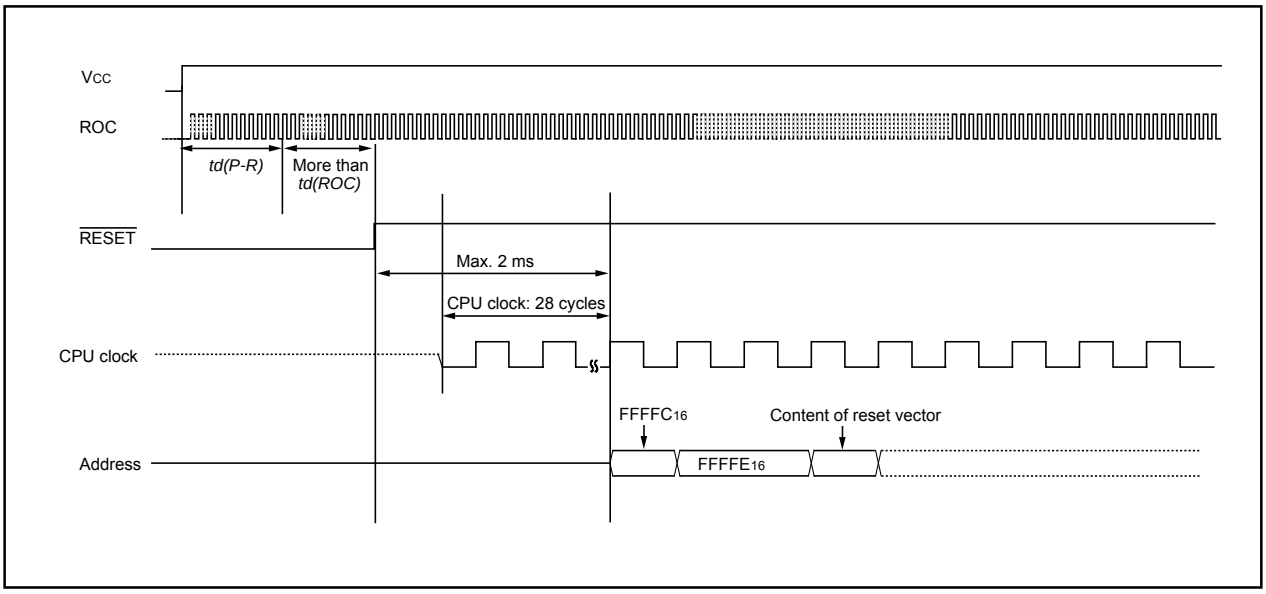


Figure 5.2 Reset Sequence

Table 5.1 Pin Status When $\overline{\text{RESET}}$ Pin Level is "L"

Pin Name	Status
P0 to P3, P6 to P10	Input port (high impedance)

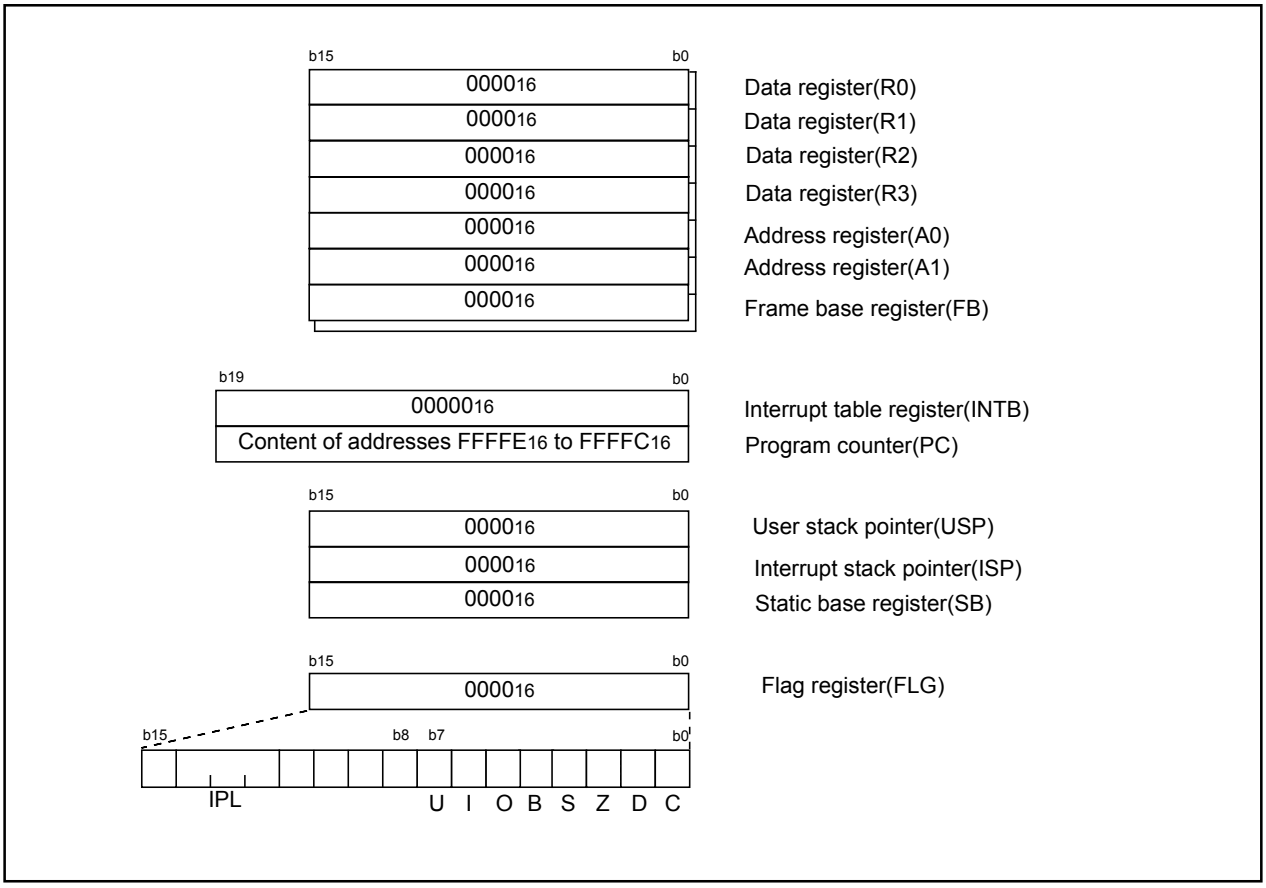


Figure 5.3 CPU Register Status After Reset

6. Processor Mode

The MCU supports single-chip mode only. **Figures 6.1** and **6.2** show the associated registers.

Processor Mode Register 0 (1)

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

Symbol	Address	After Reset
PM0	0004$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
$\overline{(b2\text{-}b0)}$	Reserved bit	Set to 0	RW
PM03	Software reset bit	The MCU is reset when this bit is set to 1. When read, its content is 0.	RW
$\overline{(b7\text{-}b4)}$	Reserved bit	Set to 0	RW

NOTE:
1. Set the PM0 register after the PRC1 bit in the PRCR register is set to 1 (write enable).

Processor Mode Register 1 (1)

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

Symbol	Address	After Reset
PM1	0005$_{16}$	00001000$_2$

Bit Symbol	Bit Name	Function	RW
PM10	Flash data block access bit (2)	0: Disabled 1: Enabled (3)	RW
$\overline{(b1)}$	Reserved bit	Set to 0	RW
PM12	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Watchdog timer reset (4)	RW
$\overline{(b3)}$	Reserved bit	Set to 1	RW
$\overline{(b6\text{-}b4)}$	Reserved bit	Set to 0	RW
PM17	Wait bit (5)	0 : No wait state 1 : Wait state (1 wait)	RW

NOTES:
1. Rewrite the PM1 register after the PRC1 bit in the PRCR register is set to 1 (write enable).
2. To access the two 2K-byte data spaces in data block A and data block B, set the PM10 bit to 1. The PM10 bit is not available in mask version.
3. When the FMR01 bit in the FMR0 register is set to 1 (enables CPU rewrite mode), the PM10 bit is automatically set to 1.
4. Set the PM12 bit to 1 by program. (Writing 0 by program has no effect)
5. When the PM17 bit is set to 1 (wait state), one wait is inserted when accessing the internal RAM or the internal ROM.

Figure 6.1 PM0 Register, PM1 Register

Processeor Mode Register 2 (1)

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

Symbol	Address	After Reset
PM2	001E16	XXX000002

Bit Symbol	Bit Name	Function	RW
PM20	Specifying wait when accessing SFR during PLL operation(2)	0: 2 waits 1: 1 wait	RW
PM21	System clock protective bit(3,4)	0: Clock is protected by PRCR register 1: Clock modification disabled	RW
PM22	WDT count source protective bit (3,5)	0: CPU clock is used for the watchdog timer count source 1: On-chip oscillator clock is used for the watchdog timer count source	RW
(b3)	Reserved bit	Set to “0”	RW
PM24	P85/$\overline{\text{NMI}}$ configuration bit(6,7)	0: P85 function ($\overline{\text{NMI}}$ disable) 1: $\overline{\text{NMI}}$ function	RW
(b7-b5)	Nothing is assigned. When write, set to“0”. When read,its content is indeterminate		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
2. The PM20 bit become effective when PLC07 bit in the PLC0 register is set to "1" (PLL on). Change the PM20 bit when the PLC07 bit is set to "0" (PLL off). Set the PM20 bit to "0" (2 waits) when PLL clock > 16 MHz.
3. Once this bit is set to “1”, it cannot be set to “0” by program.
4. Writing to the following bits has no effect when the PM21 bit is set to “1”:
   CM02 bit in the CM0 register
   CM05 bit in the CM0 register (main clock is not halted)
   CM07 bit in the CM0 register (CPU clock source does not change)
   CM10 bit in the CM1 register (stop mode is not entered)
   CM11 bit in the CM1 register (CPU clock source does not change)
   CM20 bit in the CM2 register (oscillation stop, re-oscillation detection function settings do not change)
   All bits in the PLC0 register (PLL frequency synthesizer setting do not change)
   When the PM21 bit is set to "1", do not execute the WAIT instruction.
5. Setting the PM22 bit to “1” results in the following conditions:
   - The on-chip oscillator continues oscillating even if the CM21 bit in the CM2 register is set to "0" (main clock or PLL clock) (system clock of count source selected by the CM21 bit is valid)
   - The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.
   - The CM10 bit in the CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered)
   - The watchdog timer does not stop in wait mode.
6. For $\overline{\text{NMI}}$ function, the PM24 bit must be set to “1”($\overline{\text{NMI}}$ function). Once this bit is set to “1”, it cannot be cleared to “0” by program.
7. $\overline{\text{SD}}$ input is valid regardless of the PM24 setting.

Figure 6.2 PM2 Register

The internal bus consists of CPU bus, memory bus, and peripheral bus. Bus Interface Unit (BIU) is used to interfere with CPU, ROM/RAM, and perpheral functions by controling CPU bus, memory bus, and peripheral bus. **Figure 6.3** shows the block diagram of the internal bus.

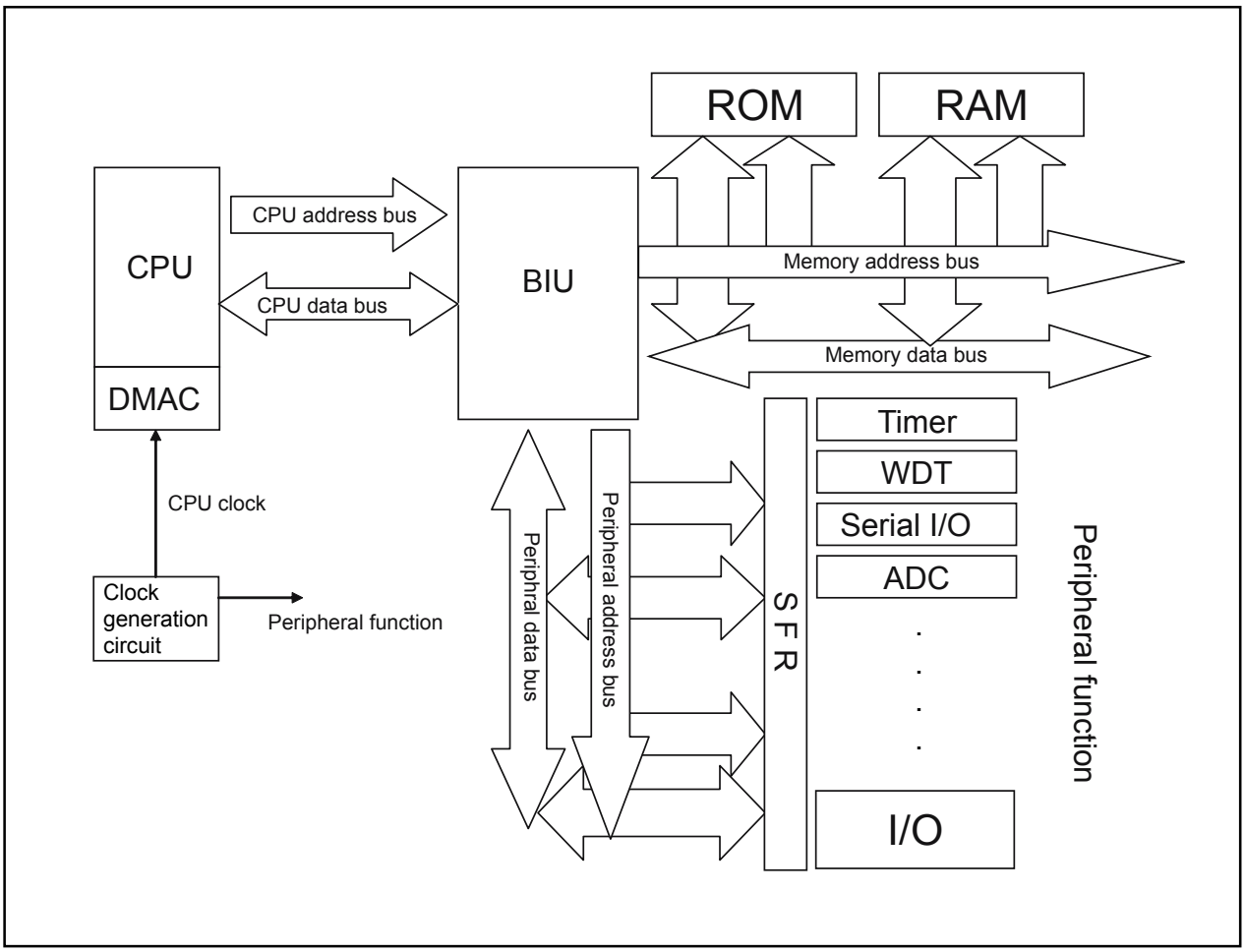


Figure 6.3 Bus Block Diagram

The number of bus cycle varies by the internal bus. **Table 6.1** lists the accessible area and bus cycle.

Table 6.1 Accessible Area and Bus Cycle

	Accessible Area	Bus Cycle
SFR	PM20 bit = 0 (2 waits)	3 CPU clock cycles
	PM20 bit = 1 (1 wait)	2 CPU clock cycles
ROM/RAM	PM17 bit = 0 (no wait)	1 CPU clock cycle
	PM17 bit = 1 (1 wait)	2 CPU clock cycles

7. Clock Generation Circuits

The MCU has four clock generation circuits as follows:

(1) Main clock oscillation circuit

(2) Sub clock oscillation circuit

(3) On-chip oscillator

(4) PLL frequency synthesizer

Table 7.1 lists the specifications of the clock generation circuit. **Figure 7.1** shows the clock generation circuits. **Figures 7.2** to **7.7** show the clock-associated registers.

Table 7.1 Clock Generation Circuit Specifications

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit	Variable On-chip Oscillator	PLL Frequency Synthesizer
Use of clock	- CPU clock source - Peripheral function clock source	- CPU clock source - Timer A, B's clock source	- CPU clock source - Peripheral function clock source - CPU and peripheral function clock sources when the main clock stops oscillating	- CPU clock source - Peripheral function clock source
Clock frequency	0 to 20 MHz	32.768 kHz	Selectable source frequency: $f_{1(ROC)}$, $f_{2(ROC)}$, $f_{3(ROC)}$ Selectable divider: by 2, by 4, by 8	10 to 20 MHz
Usable oscillator	- Ceramic oscillator - Crystal oscillator	- Crystal oscillator	———	———
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	———	———
Oscillation stop, restart function	Available	Available	Available	Available
Oscillator status after reset	Oscillating	Stopped	Oscillating (CPU clock source)	Stopped
Other	Externally derived clock can be input		———	———

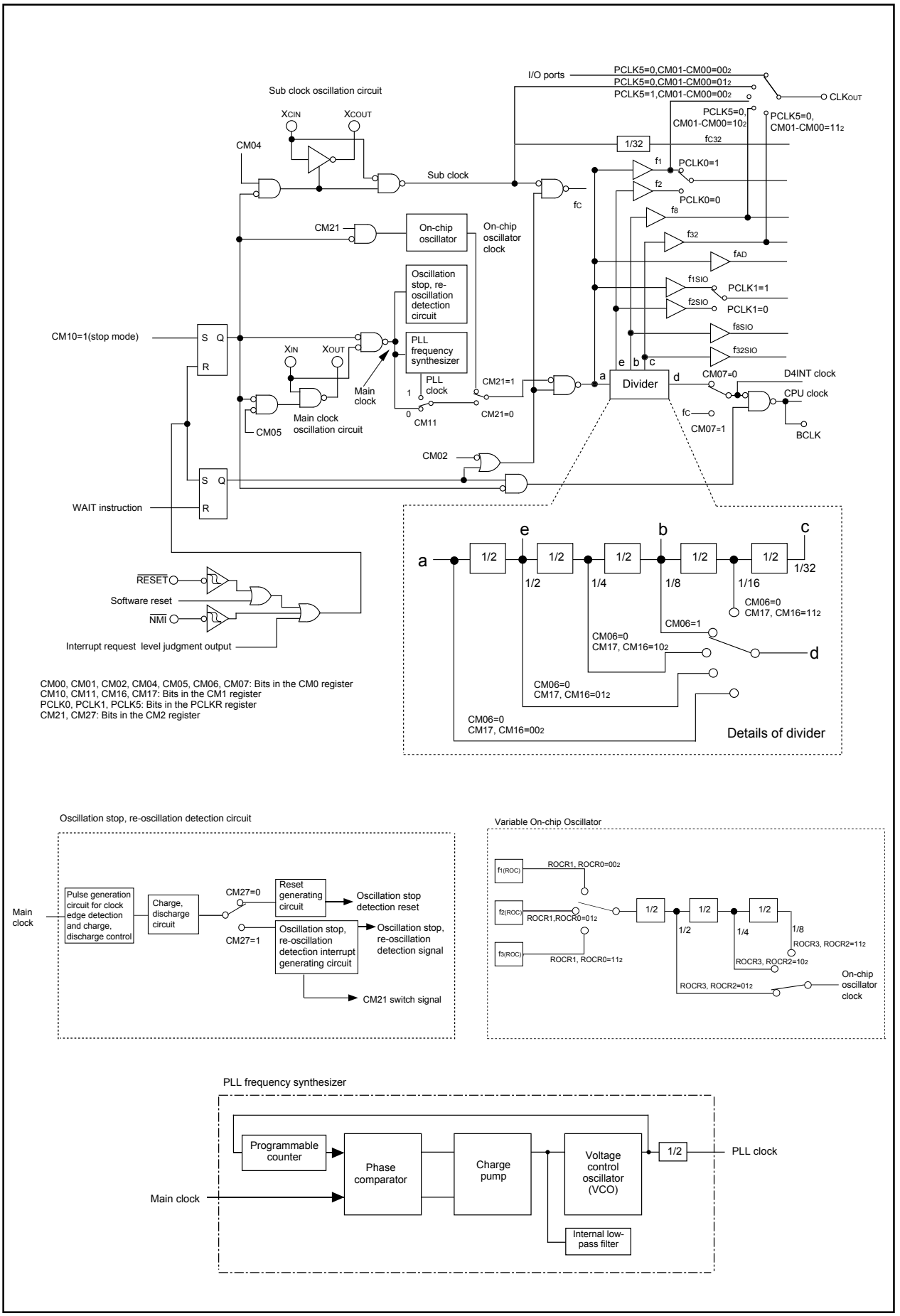


Figure 7.1 Clock Generation Circuits

System Clock Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
CM0	000616	010010002

Bit Symbol	Bit Name	Function	RW
CM00	Clock output function select bit	See **Table 7.3**	RW
CM01			RW
CM02	Wait Mode peripheral function clock stop bit (10)	0: Do not stop peripheral function clock in wait mode 1: Stop peripheral function clock in wait mode (8)	RW
CM03	XCIN-XCOUT drive capacity select bit (2)	0: LOW 1: HIGH	RW
CM04	Port XC select bit (2)	0: I/O port P86, P87 1: XCIN-XCOUT generation function(9)	RW
CM05	Main clock stop bit (3, 10, 12, 13)	0: On (4) 1: Off (5)	RW
CM06	Main clock division select bit 0 (7, 13, 14)	0: CM16 and CM17 valid 1: Division by 8 mode	RW
CM07	System clock select bit (6, 10, 11, 12)	0: Main clock, PLL clock, or on-chip oscillator clock 1: Sub-clock	RW

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to 1 (write enable).
2. The CM03 bit is set to 1 (high) when the CM04 bit is set to 0 (I/O port) or the MCU goes to a stop mode.
3. This bit is provided to stop the main clock when the low power dissipation mode or on-chip oscillator low power dissipation mode is selected. This bit cannot be used for detection as to whether the main clock stopped or not. To stop the main clock, the following setting is required:
   (1) Set the CM07 bit to 1 (Sub-clock select) or the CM21 bit in the CM2 register to 1 (on-chip oscillator select) with the sub-clock stably oscillating.
   (2) Set the CM20 bit in the CM2 register to 0 (Oscillation stop, re-oscillation detection function disabled).
   (3) Set the CM05 bit to 1 (Stop).
4. During external clock input, set the CM05 bit to 0 (On).
5. When CM05 bit is set to 1, the XOUT pin goes "H". Futhermore, because the internal feedback resistor remains connected, the XIN pin is pulled "H" to the same level as XOUT via the feedback resistor.
6. After setting the CM04 bit to 1 (XCIN-XCOUT oscillator function), wait until the sub-clock oscillates stably before switching the CM07 bit from 0 to 1 (sub-clock).
7. When entering stop mode from high or middle speed mode, on-chip oscillator mode or on-chip oscillator low power mode, the CM06 bit is set to 1 (divided-by-8 mode).
8. The fC32 clock does not stop. During low speed or low power dissipation mode, do not set this bit to 1(peripheral clock turned off in wait mode).
9. To use a sub-clock, set this bit to 1. Also, make sure ports P86 and P87 are directed for input, with no pull-ups.
10. When the PM21 bit in the PM2 register is set to 1 (clock modification disable), writing to bits CM02, CM05, and CM07 has no effect.
11. If the PM21 bit needs to be set to 1, set the CM07 bit to 0 (main clock) before setting it.
12. To use the main clock a the clock source for the CPU clock, follow the procedure below.
    (1) Set the CM05 bit to 0 (oscillate).
    (2) Wait the main clock oscillation stabilized.
    (3) Set all bits CM11, CM21, and CM07 to 0.
13. When the CM21 bit is set to 0 (on-chip oscillaor turned off) and the CM05 bit is set to 1 (main clock turned off), the CM06 bit is fixed to 1 (divide-by-8 mode) and the CM15 bit is fixed to 1 (drive capability High).
14. To return from on-chip oscillator mode to high-speed or middle-speed mode set both bits CM06 and CM15 to 1.

Figure 7.2 CM0 Register

System Clock Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0		

Symbol	Address	After Reset
CM1	0007₁₆	00100000₂

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

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to 1 (write enable).
2. When entering stop mode from high or middle speed mode, or when the CM05 bit is set to 1 (main clock turned off) in low speed mode, the CM15 bit is set to 1 (drive capability high).
3. Effective when the CM06 bit is 0 (bits CM16 and CM17 enable).
4. If the CM10 bit is 1 (stop mode), XOUT goes "H" and the internal feedback resistor is disconnected. The XCIN and XCOUT pins are placed in the high-impedance state. When the CM11 bit is set to 1 (PLL clock), or the CM20 bit in the CM2 register is set to 1 (oscillation stop, re-oscillation detection function enabled), do not set the CM10 bit to 1.
5. After setting the PLC07 bit in the PLC0 register to 1 (PLL operation), wait until tsu (PLL) elapses before setting the CM11 bit to 1 (PLL clock).
6. When the PM21 bit in the PM2 register is set to 1 (clock modification disable), writing to bits CM10, CM11 has no effect. When the PM22 bit in the PM2 register is set to 1 (watchdog timer count source is on-chip oscillator clock), writing to the CM10 bit has no effect.
7. Effective when CM07 bit is 0 and CM21 bit is 0 .

Figure 7.3 CM1 Register

On-chip Oscillator Control Register(1)

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

Symbol	Address	After Reset
ROCR	025C₁₆	X00001012

Bit Symbol	Bit Name	Function	RW
ROCR0	Frequency select bits	b1 b0 0 0: f1 (ROC) 0 1: f2 (ROC) 1 0: Do not set to this value 1 1: f3 (ROC)	RW
ROCR1			RW
ROCR2	Divider select bits	b3 b2 0 0: Do not set to this value 0 1: divide by 2 1 0: divide by 4 1 1: divide by 8	RW
ROCR3			RW
(b6-b4)	Reserved bit	Set to 0	RW
(b7)	Nothing is assigned. When write, set to 0. When read, the content is undefined		—

NOTE:
1. Write to this register after setting the PRC0 bit in the PRCR register to 1 (write enable).

Figure 7.4 ROCR Register

Oscillation Stop Detection Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
	X	0	0				

Symbol	Address	After Reset
CM2	000C16	0X000010 2(11)

Bit Symbol	Bit Name	Function	RW
CM20	Oscillation stop, re-oscillation detection bit (7, 9, 10, 11)	0: Oscillation stop, re-oscillation detection function disabled 1: Oscillation stop, re-oscillation detection function enabled	RW
CM21	System clock select bit 2 (2, 3, 6, 8, 11, 12)	0: Main clock or PLL clock 1: On-chip oscillator clock (On-chip oscillator oscillating)	RW
CM22	Oscillation stop, re-oscillation detection flag (4)	0: Main clock stop,or re-oscillation not detected 1: Main clock stop,or re-oscillation detected	RW
CM23	XIN monitor flag (5)	0: Main clock oscillating 1: Main clock not oscillating	RO
— (b5-b4)	Reserved bit	Set to 0	RW
— (b6)	Nothing is assigned. When write, set to 0. When read, its content is undefined		—
CM27	Operation select bit (when an oscillation stop, re-oscillation is detected) (11)	0: Oscillation stop detection reset 1: Oscillation stop, re-oscillation detection interrupt	RW

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to 1 (write enable).
2. When the CM20 bit is 1 (oscillation stop, re-oscillation detection function enabled), the CM27 bit is set to 1 (oscillation stop, re-oscillation detection interrupt), and the CPU clock source is the main clock, the CM21 bit is automatically set to 1 (on-chip oscillator clock) if the main clock stop is detected.
3. If the CM20 bit is set to 1 and the CM23 bit is set to 1 (main clock not oscillating), do not set the CM21 bit to 0.
4. This flag is set to 1 when the main clock is detected to have stopped or when the main clock is detected to have restarted oscillating. When this flag changes state from 0 to 1, an oscillation stop, reoscillation restart detection interrupt is generated. Use this flag in an interrupt routine to discriminate the causes of interrupts between the oscillation stop, reoscillation detection interrupts and the watchdog timer interrupt. The flag is cleared to 0 by writing 0 by program. (Writing 1 has no effect. Nor is it cleared to 0 by an oscillation stop or an oscillation restart detection interrupt request acknowledged.) If when the CM22 bit is set to 1 an oscillation stoppage or an oscillation restart is detected, no oscillation stop, reoscillation restart detection interrupts are generated.
5. Read the CM23 bit in an oscillation stop, re-oscillation detection interrupt handling routine to determine the main clock status.
6. Effective when the CM07 bit in the CM0 register is set to 0.
7. When the PM21 bit in the PM2 register is 1 (clock modification disabled), writing to the CM20 bit has no effect.
8. When the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled), the CM27 bit is set 1 (oscillation stop, re-oscillation detection interrupt), and the CM11 bit is 1 (the CPU clock source is PLL clock), the CM21 bit remains unchanged even when main clock stop is detected. If the CM22 bit is set to 0 under these conditions, oscillation stop, re-oscillation detection interrupt occur at main clock stop detection; it is, therefore, necessary to set the CM21 bit to 1 (on-chip oscillator clock) inside the interrupt routine.
9. Set the CM20 bit to 0 (disable) before entering stop mode. After exiting stop mode, set the CM20 bit back to 1 (enable).
10. Set the CM20 bit to 0 (disable) before setting the CM05 bit in the CM0 register.
11. Bits CM20, CM21 and CM27 do not change at oscillation stop detection reset.
12. When the CM21 bit is set to 0 (on-chip oscillator turned off) and the CM05 bit is set to 1 (main clock turned off), the CM06 bit is fixed to 1 (divide-by-8 mode) and the CM15 bit is fixed to 1 (drive capability High).

Figure 7.5 CM2 Register

Peripheral Clock Select Register (1)

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

Symbol	Address	After Reset
PCLKR	$025E_{16}$	00000011_2

Bit Symbol	Bit Name	Function	RW
PCLK0	Timers A, B clock select bit (Clock source for the timers A, B, the timer S, the dead timer, SI/O3, SI/O4 and multi-master I^2C bus)	0: f_2 1: f_1	RW
PCLK1	SI/O clock select bit (Clock source for UART0 to UART2)	0: f_2SIO 1: f_1SIO	RW
$\overline{(b4\text{-}b2)}$	Reserved bit	Set to 0	RW
PCLK5	Clock output function expansion select bit	Refer to **Table 7.3**	RW
$\overline{(b7\text{-}b6)}$	Reserved bit	Set to 0	RW

NOTE:
1. Write to this register after setting the PRC0 bit in PRCR register to 1 (write enable).

Processeor Mode Register 2 (1)

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

Symbol	Address	After Reset
PM2	$001E_{16}$	$XXX00000_2$

Bit Symbol	Bit Name	Function	RW
PM20	Specifying wait when accessing SFR during PLL operation(2)	0: 2 waits 1: 1 wait	RW
PM21	System clock protective bit(3,4)	0: Clock is protected by PRCR register 1: Clock modification disabled	RW
PM22	WDT count source protective bit (3,5)	0: CPU clock is used for the watchdog timer count source 1: On-chip oscillator clock is used for the watchdog timer count source	RW
$\overline{(b3)}$	Reserved bit	Set to “0”	RW
PM24	P85/$\overline{NMI}$ configuration bit(6,7)	0: $P8_5$ function ($\overline{NMI}$ disable) 1: $\overline{NMI}$ function	RW
$\overline{(b7\text{-}b5)}$	Nothing is assigned. When write, set to“0”. When read,its content is indeterminate		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to “1” (write enable).
2. The PM20 bit become effective when PLC07 bit in the PLC0 register is set to "1" (PLL on). Change the PM20 bit when the PLC07 bit is set to "0" (PLL off). Set the PM20 bit to "0" (2 waits) when PLL clock > 16 MHz.
3. Once this bit is set to “1”, it cannot be set to “0” by program.
4. Writing to the following bits has no effect when the PM21 bit is set to “1”:
   CM02 bit in the CM0 register
   CM05 bit in the CM0 register (main clock is not halted)
   CM07 bit in the CM0 register (CPU clock source does not change)
   CM10 bit in the CM1 register (stop mode is not entered)
   CM11 bit in the CM1 register (CPU clock source does not change)
   CM20 bit in the CM2 register (oscillation stop, re-oscillation detection function settings do not change)
   All bits in the PLC0 register (PLL frequency synthesizer setting do not change)
   When the PM21 bit is set to "1", do not execute the WAIT instruction.
5. Setting the PM22 bit to “1” results in the following conditions:
   - The on-chip oscillator continues oscillating even if the CM21 bit in the CM2 register is set to "0" (main clock or PLL clock) (system clock of count source selected by the CM21 bit is valid)
   - The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.
   - The CM10 bit in the CM1 register is disabled against write. (Writing a “1” has no effect, nor is stop mode entered)
   - The watchdog timer does not stop in wait mode.
6. For $\overline{NMI}$ function, the PM24 bit must be set to “1”($\overline{NMI}$ function). Once this bit is set to “1”, it cannot be cleared to “0” by program.
7. $\overline{SD}$ input is valid regardless of the PM24 setting.

Figure 7.6 PCLKR Register and PM2 Register

PLL Control Register 0 (1,2)

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

Symbol	Address	After Reset
PLC0	$001C_{16}$	$0001X010_2$

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

NOTES:
1. Write to this register after setting the PRC0 bit in the PRCR register to 1 (write enable).
2. When the PM21 bit in the PM2 register is 1 (clock modification disable), writing to this register has no effect.
3. These three bits can only be modified when the PLC07 bit is set to 0 (PLL turned off). The value once written to this bit cannot be modified.
4. Before setting this bit to 1 , set the CM07 bit to 0 (main clock), set bits CM17 to CM16 bits to 00_2 (main clock undivided mode), and set the CM06 bit to 0 (CM16 and CM17 bits enable).

Figure 7.7 PLC0 Register

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

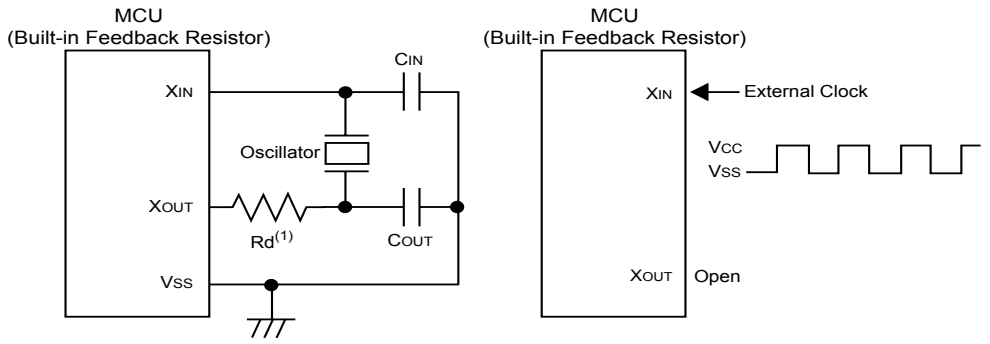
7.1 Main Clock

The main clock is generated by the main clock oscillation circuit. This clock is used as the clock source for the CPU and peripheral function clocks. The main clock oscillator circuit is configured by connecting a resonator between the XIN and XOUT pins. The main clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The main clock oscillator circuit may also be configured by feeding an exter nally generated clock to the XIN pin. **Figure 7.8** shows the examples of main clock connection circuit.

The power consumption in the chip can be reduced by setting the CM05 bit in the CM0 register to 1 (main clock oscillator circuit turned off) after switching the clock source for the CPU clock to a sub clock or on-chip oscillator clock. In this case, XOUT goes “H”. Furthermore, because the internal feedback resistor remains on, XIN is pulled “H” to XOUT via the feedback resistor.

During stop mode, all clocks including the main clock are turned off. Refer to “power control”.

If the main clock is not used, it is recommended to connect the XIN pin to VCC to reduce power consumption during reset.



NOTE:
1. Insert a damping resistor if required. Resistance value varies depending on the oscillator setting. Use resistance value 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.
2. The external clock should not be stopped when it is connected to the XIN pin and the main clock is selected as the CPU clock.

Figure 7.8 Examples of Main Clock Connection Circuit

7.2 Sub Clock

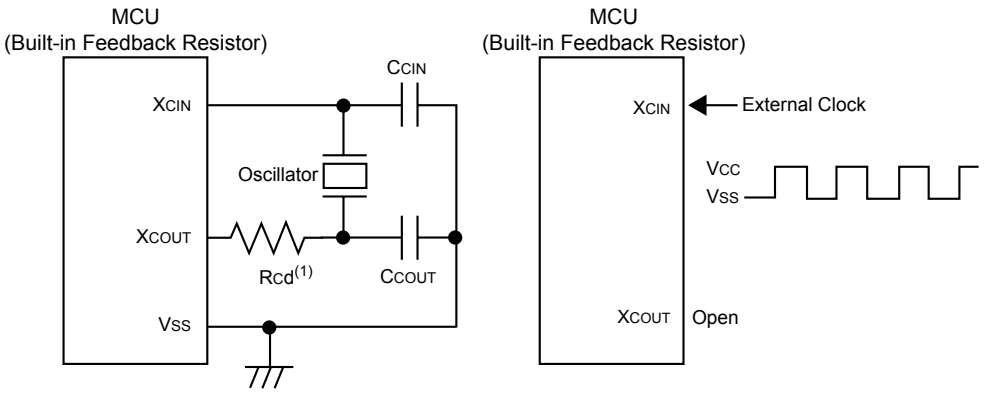
The sub clock is generated by the sub clock oscillation circuit. This clock is used as the clock source for the CPU clock, as well as the timer A and timer B count sources.

The sub clock oscillator circuit is configured by connecting a crystal resonator between the XCIN and XCOUT pins. The sub clock oscillator circuit contains a feedback resistor, which is disconnected from the oscillator circuit during stop mode in order to reduce the amount of power consumed in the chip. The sub clock oscillator circuit may also be configured by feeding an externally generated clock to the XCIN pin. **Figure 7.9** shows the examples of sub clock connection circuit.

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

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

During stop mode, all clocks including the sub clock are turned off. Refer to “power control”.



NOTE:
1. Place a damping resistor if required. Resistance values vary depending on the oscillator setting. Use values recommended by each oscillator manufacturer. Place a feedback resistor between XCIN and XCOUT if the oscillator manufacturer recommends placing the resistor externally.

Figure 7.9 Examples of Sub Clock Connection Circuit

7.3 On-chip Oscillator Clock

This clock is supplied by a variable on-chip oscillator. This clock is used as the clock source for the CPU and peripheral function clocks. In addition, if the PM22 bit in the PM2 register is 1 (on-chip oscillator clock for the watchdog timer count source), this clock is used as the count source for the watchdog timer (Refer to 10. Watchdog Timer • Count source protective mode").

After reset, the on-chip oscillator clock divided by 16 is used for the CPU clock. It can also be turned on by setting the CM21 bit in the CM2 register to 1 (on-chip oscillator clock), and is used as the clock source for the CPU and peripheral function clocks. If the main clock stops oscillating when the CM20 bit in the CM2 register is 1 (oscillation stop, re-oscillation detection function enabled) and the CM27 bit is 1 (oscillation stop, re-oscillation detection interrupt), the on-chip oscillator automatically starts operating, supplying the necessary clock for the MCU.

7.4 PLL Clock

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

Before entering wait mode or stop mode, be sure to set the CM11 bit to 0 (CPU clock source is the main clock). Furthermore, before entering stop mode, be sure to set the PLC07 bit in the PLC0 register to 0 (PLL stops). **Figure 7.10** shows the procedure for using the PLL clock as the clock source for the CPU.

The PLL clock frequency is determined by the equation below.

PLL clock frequency=f(X_{IN}) × (multiplying factor set by bits PLC02 to PLC00 in the PLC0 register

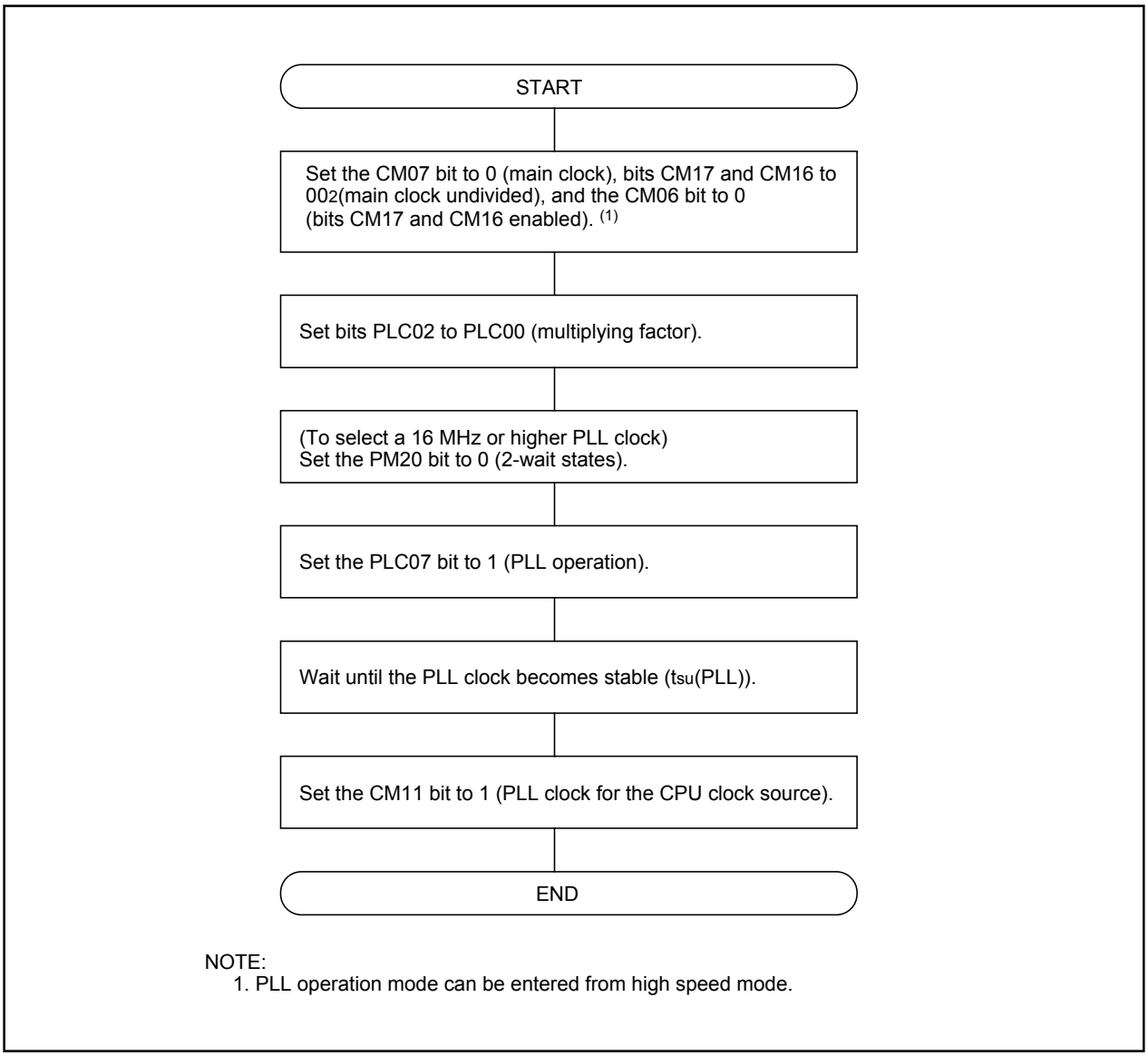
(However, 10 MHz ≤ PLL clock frequency ≤ 20 MHz)

Bits PLC02 to PLC00 can be set only once after reset. **Table 7.2** shows the example for setting PLL clock frequencies.

Table 7.2 Example for Setting PLL Clock Frequencies

X_{IN} (MHz)	PLC02	PLC01	PLC00	Multiplying factor	PLL clock (MHz)(1)
10	0	0	1	2	20
5	0	1	0	4	

NOTE:
1. 10MHz ≤ PLL clock frequency ≤ 20MHz.



NOTE:
1. PLL operation mode can be entered from high speed mode.

Figure 7.10 Procedure to Use PLL Clock as CPU Clock Source

7.5 CPU Clock and Peripheral Function Clock

The CPU clock is used to operate the CPU and peripheral function clocks are used to operate the peripheral functions.

7.5.1 CPU Clock

This is the operating clock for the CPU and watchdog timer.

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

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

When the PLL clock is selected as the clock source for the CPU clock, the CM06 bit should be set to 0 and bits CM17 and CM16 to 00_2 (undivided).

After reset, the on-chip oscillator clock divided by 16 provides the CPU clock.

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

7.5.2 Peripheral Function Clock(f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}, f_{AD}, f_{C32})

These are operating clocks for the peripheral functions.

Of these, fi (i = 1, 2, 8, 32) and f_{iSIO} are derived from the main clock, PLL clock, or on-chip oscillator clock divided by 1, 2, 8, or 32. The clock fi is used for Timer A, Timer B, SI/O3 and SI/O4 while fiSIO is used for UART0 to UART2. Additionally, the f1 and f2 are also used for dead time timer, Timer S, and multi-master I^2C bus. The f_{AD} is produced from the main clock, PLL clock, or on-chip oscillator clock, and is used for the A/D converter.

When the WAIT instruction is executed after setting the CM02 bit in the CM0 register to 1 (peripheral function clock turned off during wait mode), or when the MCU is in low power dissipation mode, the fi, f_{iSIO}, and f_{AD} are turned off.

The f_{C32} clock is produced from the sub clock, and is used for timers A and B. This clock can only be used when the sub clock is on.

7.5.3 ClockOutput Function

The f_1, f_8, f_{32} or f_C clock can be output from the CLK_{OUT} pin. Use the PCLK5 bit in the PCLKR register and bits CM01 to CM00 in the CM0 register to select. **Table 7.3** shows the function of the CLK_{OUT} pin.

Table 7.3 The function of the CLK_{OUT} pin

PCLK5	CM01	CM00	The function of the CLK_{OUT} pin
0	0	0	I/O port $P9_0$
0	0	1	f_C
0	1	0	f_8
0	1	1	f_{32}
1	0	0	f_1
1	0	1	Do not set
1	1	0	Do not set
1	1	1	Do not set

7.6 Power Control

There are three power control modes. In this chapter, all modes other than wait and stop modes are referred to as normal operation mode.

7.6.1 Normal Operation Mode

Normal operation mode is further classified into seven modes.

In normal operation mode, because the CPU clock and the peripheral function clocks both are on, the CPU and the peripheral functions are operating. Power control is exercised by controlling the CPU clock frequency. The higher the CPU clock frequency, the greater the processing capability. The lower the CPU clock frequency, the smaller the power consumption in the chip. If the unnecessary oscillator circuits are turned off, the power consumption is further reduced.

Before the clock sources for the CPU clock can be switched over, the new clock source must be in stable oscillation. If the new clock source is the main clock, sub clock or PLL clock, allow a sufficient wait time in a program until it becomes oscillating stably.

Note that operation modes cannot be changed directly from low power dissipation mode to on-chip oscillator mode or on-chip oscillator dissipation mode. Nor can operation modes be changed directly from on-chip oscillator mode or on-chip oscillator dissipation mode to low power dissipation mode.

When the CPU clock source is changed from the on-chip oscillator to the main clock, change the operation mode to the medium speed mode (divided by 8 mode) after the clock was divided by 8 (the CM06 bit in the CM0 register was set to 1) in the on-chip oscillator mode.

7.6.1.1 High-speed Mode

The main clock divided by 1 provides the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

7.6.1.2 PLL Operation Mode

The main clock multiplied by 2 or 4 provides the PLL clock, and this PLL clock serves as the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B. PLL operation mode can be entered from high speed mode. If PLL operation mode is to be changed to wait or stop mode, first go to high speed mode before changing.

7.6.1.3 Medium-speed Mode

The main clock divided by 2, 4, 8 or 16 provides the CPU clock. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

7.6.1.4 Low-speed Mode

The sub clock provides the CPU clock. The main clock is used as the clock source for the peripheral function clock when the CM21 bit is set to 0 (on-chip oscillator turned off), and the on-chip oscillator clock is used when the CM21 bit is set to 1 (on-chip oscillator oscillating).

The f_{C32} clock can be used as the count source for timers A and B.

7.6.1.5 Low Power Dissipation Mode

In this mode, the main clock is turned off after being placed in low speed mode. The sub clock provides the CPU clock. The f_{C32} clock can be used as the count source for timers A and B. Peripheral function clock can use only f_{C32}.

Simultaneously when this mode is selected, the CM06 bit in the CM0 register becomes 1 (divided by 8 mode). In the low power dissipation mode, do not change the CM06 bit. Consequently, the medium speed (divided by 8) mode is to be selected when the main clock is operated next.

7.6.1.6 On-chip Oscillator Mode

The selected on-chip oscillator clock divided by 1 (undivided), 2, 4, 8 or 16 provides the CPU clock. The on-chip oscillator clock is also the clock source for the peripheral function clocks. If the sub clock is on, f_{C32} can be used as the count source for timers A and B. The on-chip oscillator frequency can be selected by bits ROCR3 to ROCR0 in the ROCR register. When the operation mode is returned to the high and medium speed modes, set the CM06 bit to 1 (divided by 8 mode).

7.6.1.7 On-chip Oscillator Low Power Dissipation Mode

The main clock is turned off after being placed in on-chip oscillator mode. The CPU clock can be selected as in the on-chip oscillator mode. The on-chip oscillator clock is the clock source for the peripheral function clocks. If the sub clock is on, f_{C32} can be used as the count source for timers A and B.

Table 7.4 Setting Clock Related Bit and Modes

Modes		CM2 Register	CM1 Register		CM0 Register			
		CM21	CM11	CM17, CM16	CM07	CM06	CM05	CM04
PLL operation mode		0	1	00_2	0	0	0	—
High-speed mode		0	0	00_2	0	0	0	—
Medium-speed mode	divided by 2	0	0	01_2	0	0	0	—
	divided by 4	0	0	10_2	0	0	0	—
	divided by 8	0	0	—	0	1	0	—
	divided by 16	0	0	11_2	0	0	0	—
Low-speed mode		—	—	—	1	—	0	1
Low power dissipation mode		—	—	—	1	1(1)	1(1)	1
On-chip oscillator mode(3)	divided by 1	1	—	00_2	0	0	0	—
	divided by 2	1	—	01_2	0	0	0	—
	divided by 4	1	—	10_2	0	0	0	—
	divided by 8	1	—	—	0	1	0	—
	divided by 16	1	—	11_2	0	0	0	—
On-chip oscillator low power dissipation mode		1	—	(2)	0	(2)	1	—

NOTES:
1. When the CM05 bit is set to 1 (main clock turned off) in low-speed mode, the mode goes to low power dissipation mode and CM06 bit is set to 1(divided by 8 mode) simultaneously.
2. The divide-by-n value can be selected the same way as in on-chip oscillator mode.
3. On-chip oscillator frequency can be any of those described in the section **7.6.1.6 On-chip Oscillator Mode**.

7.6.2 Wait Mode

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

7.6.2.1 Peripheral Function Clock Stop Function

When the CM02 bit is 1 (peripheral function clocks turned off during wait mode), f_1, f_2, f_8, f_{32}, f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}, and f_{AD} stop running in wait mode to reduce power consumption. However, f_{C32} remains active.

7.6.2.2 Entering Wait Mode

The MCU enters wait mode by executing the WAIT instruction.

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

7.6.2.3 Pin Status During Wait Mode

Table 7.5 lists pin status during wait mode.

Table 7.5 Pin Status in Wait Mode

Pin		Status
I/O ports		Maintains status immediately before entering wait mode
CLKOUT	When fC selected	Outputs clock
	When f1, f8, f32 selected	Outputs the clock when the CM02 bit in the CM0 register is set to 0 (peripheral clock does not stop in wait mode
		Maintains state immediately before entering stop mode when the CM02 bit is set to 1 (peripheral clock stops in wait mode)

7.6.2.4 Exiting Wait Mode

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

If wait mode is exited by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority bits ILVL2 to ILVL0 to 000_2 (interrupts disabled) before executing the WAIT instruction.

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

Table 7.6 lists the interrupts to exit wait mode.

Table 7.6 Interrupts to Exit Wait Mode

Interrupt	CM02 = 0	CM02 = 1
$\overline{\text{NMI}}$ interrupt	Available	Available
Serial I/O interrupt	Available when internal and external clocks are used	Available when external clock is used
Multi-master I^2C interrupt	Available	Do not used
Key input interrupt	Available	Available
A/D conversion interrupt	Available in one-shot or single sweep mode	Do not use
Timer A interrupt Timer B interrupt	Available in all modes	Available in event counter mode or when count source is fC32
Timer S interrupt	Available in all modes	Do not use
$\overline{\text{INT}}$ interrupt	Available	Available

To use peripheral function interrupts to exit wait mode, set the followings before executing the WAIT instruction.

1. Set the interrupt priority level to the bits ILVL2 to ILVL0 in the interrupt control register of the peripheral function interrupts that are used to exit wait mode. Also, set bits ILVL2 to ILVL0 of all peripheral function interrupts that are not used to exit wait mode to 000_2 (interrupt disabled).
2. Set the I flag to 1.
3. Operate the peripheral functions that are used to exit wait mode.

When the peripheral function interrupts are used to exit wait mode, an interrupt routine is executed after an interrupt request is generated and the CPU is clocked.

The CPU clock used when exiting wait mode by a peripheral function interrupt is the same CPU clock that is used when executing the WAIT instruction.

7.6.3 Stop Mode

In stop mode, all oscillator circuits are turned off, so are the CPU clock and the peripheral function clocks. Therefore, the CPU and the peripheral functions clocked by these clocks stop operating. The least amount of power is consumed in this mode. If the voltage applied to Vcc pin is VRAM or more, the internal RAM is retained. When applying 2.7 or less voltage to Vcc pin, make sure Vcc≥VRAM.

However, the peripheral functions clocked by external signals keep operating. The following interrupts can be used to exit stop mode.

- $\overline{\text{NMI}}$ interrupt
- Key interrupt
- $\overline{\text{INT}}$ interrupt
- Timer A, Timer B interrupt (when counting external pulses in event counter mode)
- Serial I/O interrupt (when external clock is selected)

7.6.3.1 Entering Stop Mode

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

Before entering stop mode, set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disable).

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

7.6.3.2 Pin Status during Stop Mode

The I/O pins retain their status held just prior to entering stop mode.

7.6.3.3 Exiting Stop Mode

The MCU is moved out of stop mode by a hardware reset, $\overline{\text{NMI}}$ interrupt or peripheral function interrupt.

If the MCU is to be moved out of stop mode by a hardware reset or $\overline{\text{NMI}}$ interrupt, set the peripheral function interrupt priority bits ILVL2 to ILVL0 to 000_2 (interrupts disable) before setting the CM10 bit to 1.

If the MCU is to be moved out of stop mode by a peripheral function interrupt, set up the following before setting the CM10 bit to 1.

1. In bits ILVL2 to ILVL0 of the interrupt control register, set the interrupt priority level of the peripheral function interrupt to be used to exit stop mode.
   Also, for all of the peripheral function interrupts not used to exit stop mode, set bits ILVL2 to ILVL0 to 000_2.
2. Set the I flag to 1.
3. Enable the peripheral function whose interrupt is to be used to exit stop mode.
   In this case, when an interrupt request is generated and the CPU clock is thereby turned on, an interrupt service routine is executed.

Which CPU clock will be used after exiting stop mode by a peripheral function or $\overline{\text{NMI}}$ interrupt is determined by the CPU clock that was on when the MCU was placed into stop mode as follows:

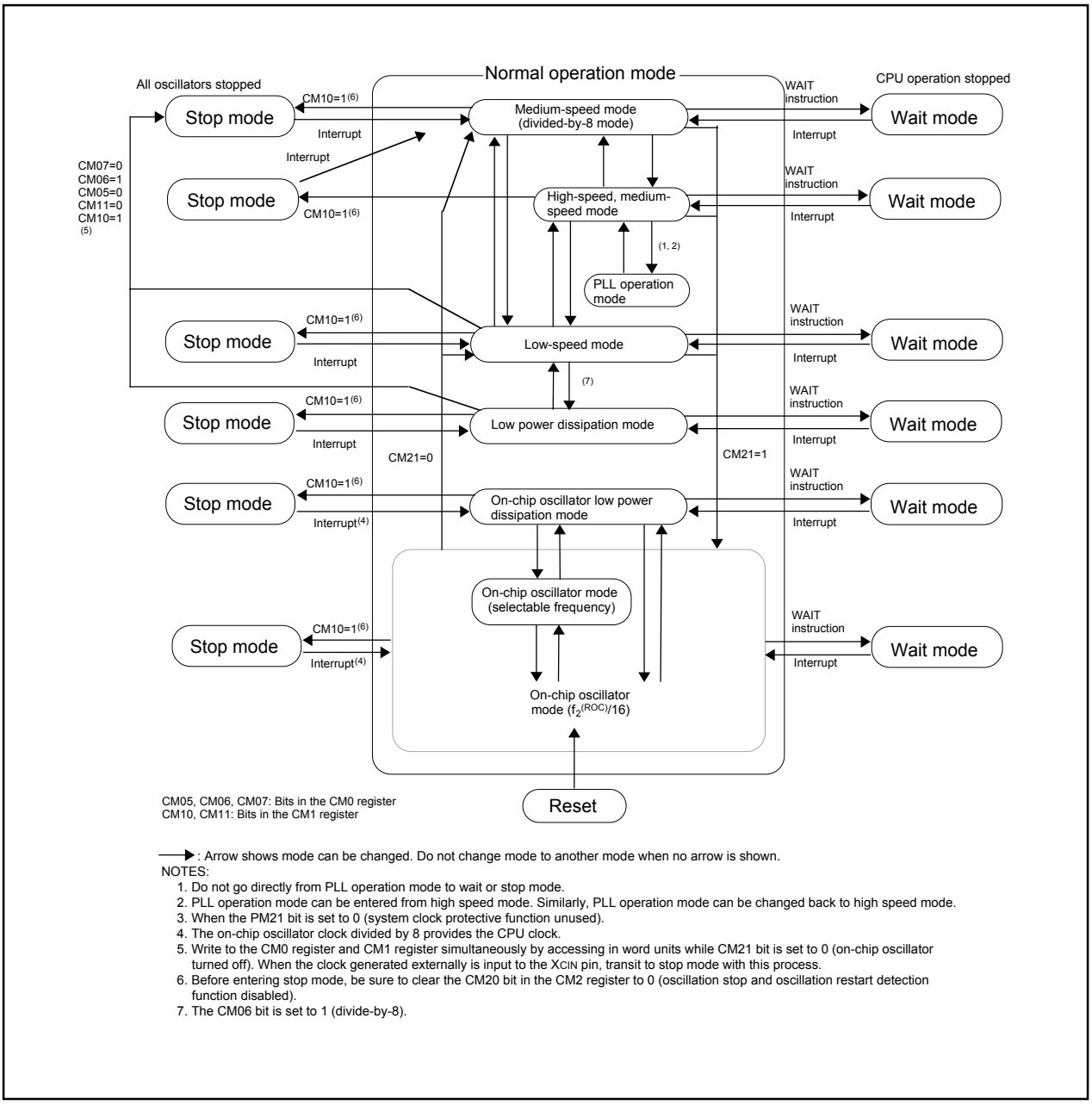
If the CPU clock before entering stop mode was derived from the sub clock: sub clock

If the CPU clock before entering stop mode was derived from the main clock: main clock divide-by-8

If the CPU clock before entering stop mode was derived from the on-chip oscillator clock: on-chip oscillator clock divide-by-8

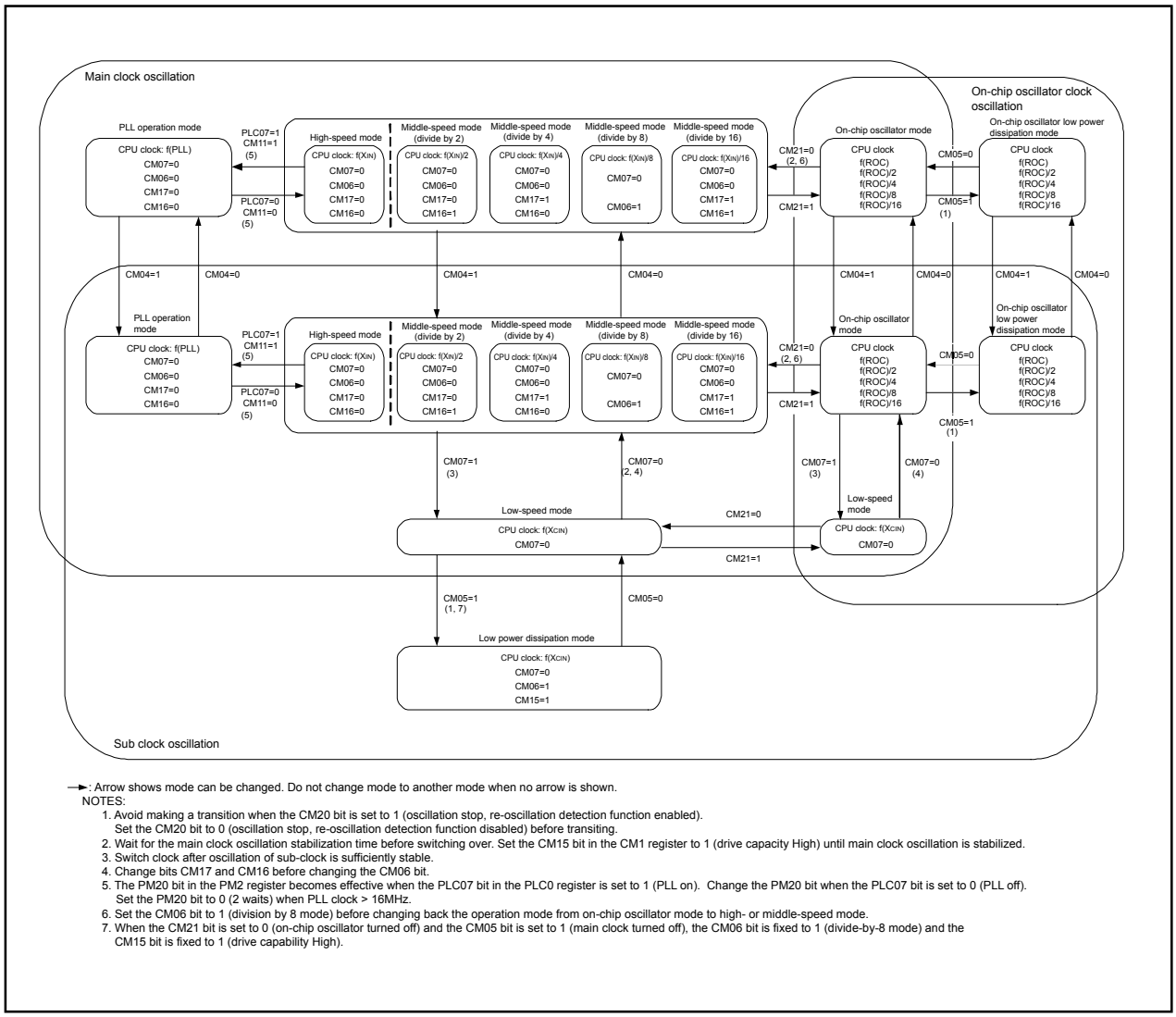
Figure 7.11 shows the state transition from normal operation mode to stop mode and wait mode. **Figure 7.12** shows the state transition in normal operation mode.

Table 7.7 shows a state transition matrix describing allowed transition and setting. The vertical line shows current state and horizontal line shows state after transition.



NOTES:
1. Do not go directly from PLL operation mode to wait or stop mode.
2. PLL operation mode can be entered from high speed mode. Similarly, PLL operation mode can be changed back to high speed mode.
3. When the PM21 bit is set to 0 (system clock protective function unused).
4. The on-chip oscillator clock divided by 8 provides the CPU clock.
5. Write to the CM0 register and CM1 register simultaneously by accessing in word units while CM21 bit is set to 0 (on-chip oscillator turned off). When the clock generated externally is input to the XCIN pin, transit to stop mode with this process.
6. Before entering stop mode, be sure to clear the CM20 bit in the CM2 register to 0 (oscillation stop and oscillation restart detection function disabled).
7. The CM06 bit is set to 1 (divide-by-8).

Figure 7.11 State Transition to Stop Mode and Wait Mode



→: Arrow shows mode can be changed. Do not change mode to another mode when no arrow is shown.

NOTES:

1. Avoid making a transition when the CM20 bit is set to 1 (oscillation stop, re-oscillation detection function enabled). Set the CM20 bit to 0 (oscillation stop, re-oscillation detection function disabled) before transiting.
2. Wait for the main clock oscillation stabilization time before switching over. Set the CM15 bit in the CM1 register to 1 (drive capacity High) until main clock oscillation is stabilized.
3. Switch clock after oscillation of sub-clock is sufficiently stable.
4. Change bits CM17 and CM16 before changing the CM06 bit.
5. The PM20 bit in the PM2 register becomes effective when the PLC07 bit in the PLC0 register is set to 1 (PLL on). Change the PM20 bit when the PLC07 bit is set to 0 (PLL off). Set the PM20 bit to 0 (2 waits) when PLL clock > 16MHz.
6. Set the CM06 bit to 1 (division by 8 mode) before changing back the operation mode from on-chip oscillator mode to high- or middle-speed mode.
7. When the CM21 bit is set to 0 (on-chip oscillator turned off) and the CM05 bit is set to 1 (main clock turned off), the CM06 bit is fixed to 1 (divide-by-8 mode) and the CM15 bit is fixed to 1 (drive capability High).

Figure 7.12 State Transition in Normal Mode

Table 7.7 Allowed Transition and Setting

Current state	State after transition: High-speed mode, middle-speed mode	Low-speed mode[2]	Low power dissipation mode	PLL operation mode[2]	On-chip oscillator mode	On-chip oscillator low power dissipation mode	Stop mode	Wait mode
High-speed mode, middle-speed mode	8	(9)[7]	--	(13)[3]	(15)	--	(16)[1]	(17)
Low-speed mode[2]	(8)		(11)[1, 6]	--	(8)	--	(16)[1]	(17)
Low power dissipation mode	--	(10)		--	--	--	(16)[1]	(17)
PLL operation mode[2]	(12)[3]	--	--		--	--	--	--
On-chip oscillator mode	(14)[4]	(9)[7]	--	--	8	(11)[1]	(16)[1]	(17)
On-chip oscillator low power dissipation mode	--	--	--	--	(10)	8	(16)[1]	(17)
Stop mode	(18)[5]	(18)	(18)	--	(18)[5]	(18)[5]		--
Wait mode	(18)	(18)	(18)	--	(18)	(18)	--	

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

--: Cannot transit

		Sub clock oscillating: No division	Divided by 2	Divided by 4	Divided by 8	Divided by 16	Sub clock turned off: No division	Divided by 2	Divided by 4	Divided by 8	Divided by 16
Sub clock oscillating	No division		(4)	(5)	(7)	(6)	(1)	--	--	--	--
	Divided by 2	(3)		(5)	(7)	(6)	--	(1)	--	--	--
	Divided by 4	(3)	(4)		(7)	(6)	--	--	(1)	--	--
	Divided by 8	(3)	(4)	(5)		(6)	--	--	--	(1)	--
	Divided by 16	(3)	(4)	(5)	(7)		--	--	--	--	(1)
Sub clock turned off	No division	(2)	--	--	--	--		(4)	(5)	(7)	(6)
	Divided by 2	--	(2)	--	--	--	(3)		(5)	(7)	(6)
	Divided by 4	--	--	(2)	--	--	(3)	(4)		(7)	(6)
	Divided by 8	--	--	--	(2)	--	(3)	(4)	(5)		(6)
	Divided by 16	--	--	--	--	(2)	(3)	(4)	(5)	(7)	

9. ( ) : setting method. Refer to following table.

--: Cannot transit

	Setting	Operation
(1)	CM04 = 0	Sub clock turned off
(2)	CM04 = 1	Sub clock oscillating
(3)	CM06 = 0, CM17 = 0 , CM16 = 0	CPU clock no division mode
(4)	CM06 = 0, CM17 = 0 , CM16 = 1	CPU clock division by 2 mode
(5)	CM06 = 0, CM17 = 1 , CM16 = 0	CPU clock division by 4 mode
(6)	CM06 = 0, CM17 = 1 , CM16 = 1	CPU clock division by 16 mode
(7)	CM06 = 1	CPU clock division by 8 mode
(8)	CM07 = 0	Main clock, PLL clock, or on-chip oscillator clock selected
(9)	CM07 = 1	Sub clock selected
(10)	CM05 = 0	Main clock oscillating
(11)	CM05 = 1	Main clock turned off
(12)	PLC07 = 0, CM11 = 0	Main clock selected
(13)	PLC07 = 1, CM11 = 1	PLL clock selected
(14)	CM21 = 0	Main clock or PLL clock selected
(15)	CM21 = 1	On-chip oscillator clock selected
(16)	CM10 = 1	Transition to stop mode
(17)	wait instruction	Transition to wait mode
(18)	Hardware interrupt	Exit stop mode or wait mode

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

7.7 System Clock Protective Function

When the main clock is selected for the CPU clock source, this function protects the clock from modifications in order to prevent the CPU clock from becoming halted by run-away.

If the PM21 bit in the PM2 register is set to 1 (clock modification disabled), the following bits are protected against writes:

- Bits CM02, CM05, and CM07 in CM0 register
- Bits CM10 and CM11 in CM1 register
- CM20 bit in CM2 register
- All bits in the PLC0 register

Before the system clock protective function can be used, the following register settings must be made while the CM05 bit in the CM0 register is 0 (main clock oscillating) and CM07 bit is 0 (main clock selected for the CPU clock source):

(1) Set the PRC1 bit in the PRCR register to 1 (enable writes to PM2 register).

(2) Set the PM21 bit in the PM2 register to 1 (disable clock modification).

(3) Set the PRC1 bit in the PRCR register to 0 (disable writes to PM2 register).

Do not execute the WAIT instruction when the PM21 bit is 1.

7.8 Oscillation Stop and Re-oscillation Detect Function

The oscillation stop and re-oscillation detect function detects the re-oscillation after stop of main clock oscillation circuit. When the oscillation stop and re-oscillation detection occurs, the oscillation stop detect function is reset or oscillation stop and re-oscillation detection interrupt is generated, depending on the CM27 bit set in the CM2 register. The oscillation stop detect function is enabled or disabled by the CM20 bit in the CM2 register. **Table 7.8** lists a specification overview of the oscillation stop and re-oscillation detect function.

Table 7.8 Specification Overview of Oscillation Stop and Re-oscillation Detect Function

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	$f(X_{IN}) \geq 2$ MHz
Enabling condition for oscillation stop, re-oscillation detection function	Set CM20 bit to 1(enable)
Operation at oscillation stop, re-oscillation detection	•Reset occurs (when CM27 bit =0) •Oscillation stop, re-oscillation detection interrupt occurs(when CM27 bit =1)

7.8.1 Operation When CM27 bit = 0 (Oscillation Stop Detection Reset)

When main clock stop is detected when the CM20 bit is 1 (oscillation stop, re-oscillation detection function enabled), the MCU is initialized, coming to a halt (oscillation stop reset; refer to “SFR”, “Reset”). This status is reset with hardware reset 1. Also, even when re-oscillation is detected, the MCU can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to 1 and the CM27 bit to 0.)

7.8.2 Operation When CM27 bit = 1 (Oscillation Stop and Re-oscillation Detect Interrupt)

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

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

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

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

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

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source by program. **Figure 7.13** shows the procedure for switching the clock source from the on-chip oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes 1. When the CM22 bit is set at 1, oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to 0 by program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is 1, an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the on-chip oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the on-chip oscillator clock.
- To enter wait mode while using the oscillation stop, re-oscillation detection function, set the CM02 bit to 0 (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop, re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to 0 (Oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated by program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to 0.

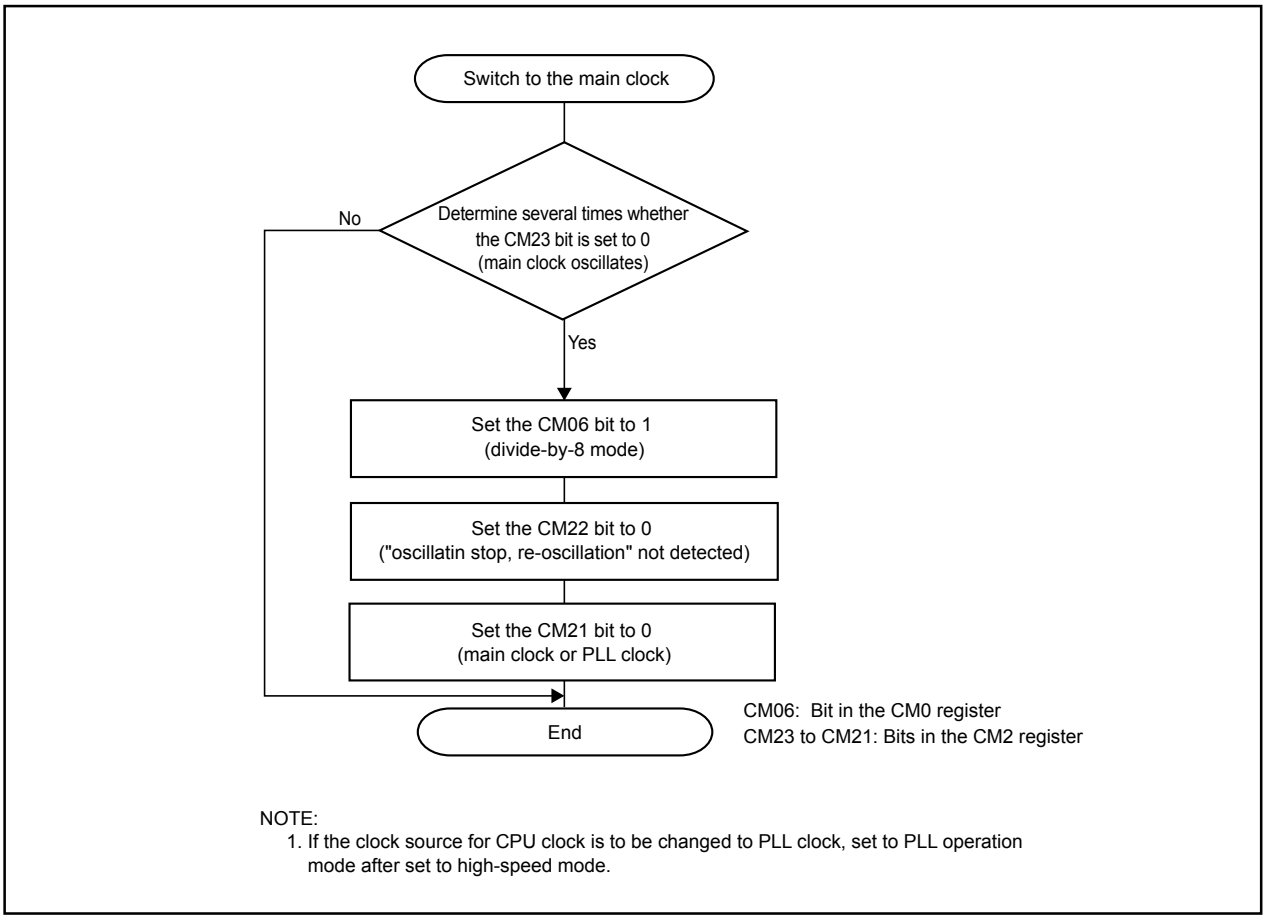


Figure 7.13 Procedure to Switch Clock Source From On-chip Oscillator to Main Clock

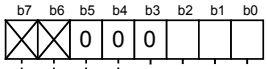
8. Protection

In the event that a program runs out of control, this function protects the important registers so that they will not be rewritten easily. **Figure 8.1** shows the PRCR register. The following lists the registers protected by the PRCR register.

- Registers protected by the PRC0 bit: CM0, CM1, CM2, LPCC1, PLC0, ROCR, and PCLKR
- Registers protected by the PRC1 bit: PM0, PM1, PM2, TB2SC, INVC0, and INVC1
- Registers protected by the PRC2 bit: PD9 , PACR, S4C, and NDDR

The PRC2 bit is set to 0 (write enabled) when data is written to the SFR area after setting the PRC2 bit to 1 (write enable). Set registers PD9, PACR, S4C and NDDR immediately after setting the PRC2 bit in the PRCR register to 1 (write enable). Do not generate an interrupt or a DMA transfer between the instruction to set the PRC2 bit to 1 and the next instruction. Bits PRC3, PRC1, and PRC0 are not set to 0 even if data is written to the SFR area. Set bits PRC3, PRC1, and PRC0 to 0 by program.

Protect Register



Symbol	Address	After Reset
PRCR	000A16	XX0000002

Bit Symbol	Bit Name	Function	RW
PRC0	Protect bit 0	Enable write to registers CM0, CM1, CM2, LPCC1, ROCR, PLC0, and PCLKR 0 : Write protected 1 : Write enabled	RW
PRC1	Protect bit 1	Enable write to PM0, PM1, PM2, TB2SC, INVC0 and INVC1 registers 0 : Write protected 1 : Write enabled	RW
PRC2	Protect bit 2	Enable write to PD9, PACR, S4C, and NDDR registers 0 : Write protected 1 : Write enabled(1)	RW
(b5-b3)	Reserved bit	Set to 0	RW
(b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—

NOTE:
1. The PRC2 bit is set to 0 when writing into the SFR area after the PRC2 bit is set to 1. Bits PRC0 and PRC1 are not automatically set to 0. Set them to 0 by program.

Figure 8.1 PRCR Register

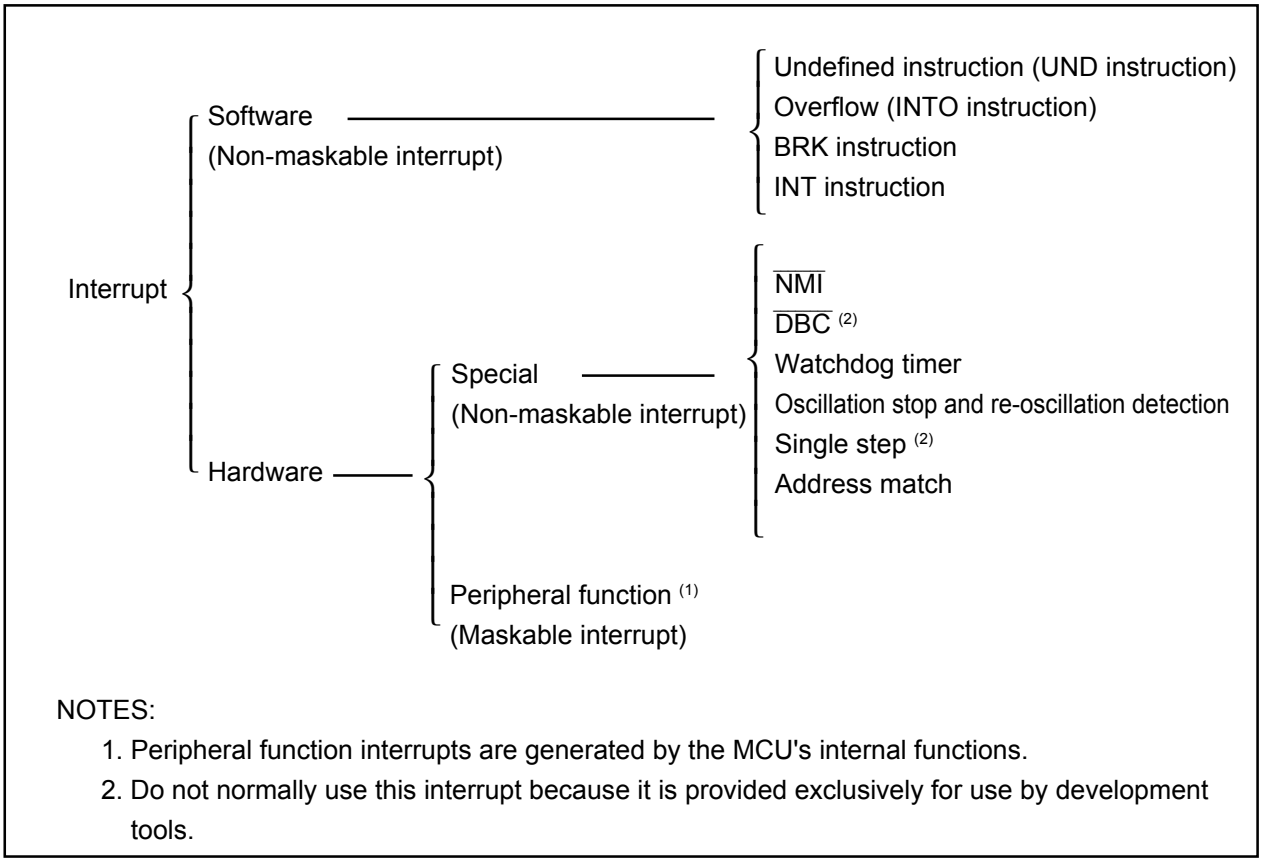
9. Interrupts

Note

The SI/O4 interrupt of peripheral function interrupts is not available in the 64-pin package.

9.1 Type of Interrupts

Figure 9.1 shows types of interrupts.



NOTES:
1. Peripheral function interrupts are generated by the MCU's internal functions.
2. Do not normally use this interrupt because it is provided exclusively for use by development tools.

Figure 9.1 Interrupts

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

9.1.1 Software Interrupts

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

9.1.1.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

9.1.1.2 Overflow Interrupt

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

9.1.1.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

9.1.1.4 INT Instruction Interrupt

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

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

9.1.2 Hardware Interrupts

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

9.1.2.1 Special Interrupts

Special interrupts are non-maskable interrupts.

9.1.2.1.1 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low. For details about the $\overline{\text{NMI}}$ interrupt, refer to the section "NMI interrupt".

9.1.2.1.2 $\overline{\text{DBC}}$ Interrupt

This interrupt is exclusively for debugger, do not use in any other circumstances.

9.1.2.1.3 Watchdog Timer Interrupt

Generated by the watchdog timer. Once a watchdog timer interrupt is generated, be sure to initialize the watchdog timer. For details about the watchdog timer, refer to the section "watchdog timer".

9.1.2.1.4 Oscillation Stop and Re-oscillation Detection Interrupt

Generated by the oscillation stop and re-oscillation detection function. For details about the oscillation stop and re-oscillation detection function, refer to the section "clock generating circuit".

9.1.2.1.5 Single-step Interrupt

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

9.1.2.1.6 Address Match Interrupt

An address match interrupt is generated immediately before executing the instruction at the address indicated by the RMAD0 or RMAD1 register, if the corresponding enable bit (AIER0 or AIER1 bit in the AIER register) is set to 1. For details about the address match interrupt, refer to the section “address match interrupt”.

9.1.2.2 Peripheral Function Interrupts

Peripheral function interrupts are maskable interrupts and generated by the MCU's internal functions. The interrupt sources for peripheral function interrupts are listed in **Table 9.2** Relocatable Vector Tables. For details about the peripheral functions, refer to the description of each peripheral function in this manual.

9.2 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. **Figure 9.2** shows the interrupt vector.

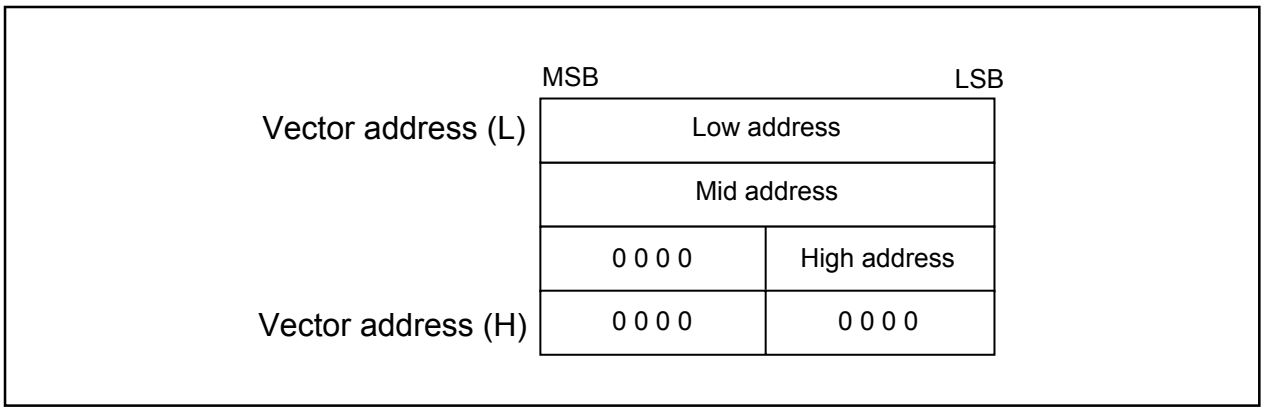


Figure 9.2 Interrupt Vector

9.2.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC16 to FFFFF16. **Table 9.1** lists the fixed vector tables. In the flash memory version of MCU, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section "flash memory rewrite disabling function".

Table 9.1 Fixed Vector Tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction	M16C/60, M16C/20 serise software maual
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction	
BRK instruction	FFFE416 to FFFE716	If the contents of address FFFE716 is FF16, program execution starts from the address shown by the vector in the relocatable vector table	
Address match	FFFE816 to FFFEB16		Address match interrupt
Single step (1)	FFFEC16 to FFFEF16		
Watchdog timer Oscillation stop and re-oscillation detection	FFFF016 to FFFF316		Watchdog timer, clock generation circuits
$\overline{\text{DBC}}$(1)	FFFF416 to FFFF716		
$\overline{\text{NMI}}$	FFFF816 to FFFFB16		$\overline{\text{NMI}}$ interrupt
Reset(2)	FFFFC16 to FFFFF16		Reset

NOTE:

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

9.2.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a reloacatable vector table area. **Table 9.2** lists the relocatable vector tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than in the case of odd addresses.

Table 9.2 Relocatable Vector Tables

Interrupt source	Vector address (1) Address (L) to address (H)	Software interrupt number	Reference
BRK instruction (5)	+0 to +3 (0000 16 to 0003 16)	0	M16C/60, M16C/20 series software manual
——— (Reserved)		1 to 3	
$\overline{\text{INT3}}$	+16 to +19 (0010 16 to 0013 16)	4	$\overline{\text{INT}}$ interrupt
IC/OC interrupt 0	+20 to +23 (0014 16 to 0017 16)	5	Timer S
IC/OC interrupt 1, I²C bus interface (4)	+24 to +27 (0018 16 to 001B 16)	6	Timer S Multi-Master I²C bus interface
IC/OC base timer, SCL/SDA (4)	+28 to +31 (001C 16 to 001F 16)	7	
SI/O4, $\overline{\text{INT5}}$ (2)	+32 to +35 (0020 16 to 0023 16)	8	$\overline{\text{INT}}$ interrupt Serial I/O
SI/O3, $\overline{\text{INT4}}$ (2)	+36 to +39 (0024 16 to 0027 16)	9	
UART 2 bus collision detection (6)	+40 to +43 (0028 16 to 002B 16)	10	Serial I/O
DMA0	+44 to +47 (002C 16 to 002F 16)	11	DMAC
DMA1	+48 to +51 (0030 16 to 0033 16)	12	
Key input interrupt	+52 to +55 (0034 16 to 0037 16)	13	Key input interrupt
A/D	+56 to +59 (0038 16 to 003B 16)	14	A/D convertor
UART2 transmit, NACK2 (3)	+60 to +63 (003C 16 to 003F 16)	15	Serial I/O
UART2 receive, ACK2 (3)	+64 to +67 (0040 16 to 0043 16)	16	
UART0 transmit	+68 to +71 (0044 16 to 0047 16)	17	
UART0 receive	+72 to +75 (0048 16 to 004B 16)	18	
UART1 transmit	+76 to +79 (004C 16 to 004F 16)	19	
UART1 receive	+80 to +83 (0050 16 to 0053 16)	20	
Timer A0	+84 to +87 (0054 16 to 0057 16)	21	Timer
Timer A1	+88 to +91 (0058 16 to 005B 16)	22	
Timer A2	+92 to +95 (005C 16 to 005F 16)	23	
Timer A3	+96 to +99 (0060 16 to 0063 16)	24	
Timer A4	+100 to +103 (0064 16 to 0067 16)	25	
Timer B0	+104 to +107 (0068 16 to 006B 16)	26	
Timer B1	+108 to +111 (006C 16 to 006F 16)	27	
Timer B2	+112 to +115 (0070 16 to 0073 16)	28	
$\overline{\text{INT0}}$	+116 to +119 (0074 16 to 0077 16)	29	$\overline{\text{INT}}$ interrupt
$\overline{\text{INT1}}$	+120 to +123 (0078 16 to 007B 16)	30	
$\overline{\text{INT2}}$	+124 to +127 (007C 16 to 007F 16)	31	
Software interrupt (5)	+128 to +131 (0080 16 to 0083 16) to +252 to +255 (00FC 16 to 00FF 16)	32 to 63	M16C/60, M16C/20 series software manual

NOTES:
1. Address relative to address in INTB.
2. Use the IFSR6 and IFSR7 bits in the IFSR register to select.
3. During I²C bus mode, NACK and ACK interrupts comprise the interrupt source.
4. Use the IFSR26 and IFSR27 bits in the IFSR2A register to select.
5. These interrupts cannot be disabled using the I flag.
6. Bus collision detection:
   During IEBus mode, this bus collision detection constitutes the cause of an interrupt.
   During I²C Bus mode, however, a start condition or a stop condition detection constitutes the cause of an interrupt.

9.3 Interrupt Control

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

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

Figure 9.3 shows the interrupt control registers.

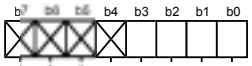
Also, the following interrupts share a vector and an interrupt control register.

- $\overline{\text{INT4}}$ and SIO3
- $\overline{\text{INT5}}$ and SIO4
- IC/OC base timer and SCL/SDA
- IC/OC interrupt 1 and I^2C bus interface

An interrupt request is set by bits IFSR7 and IFSR6 in the IFSR register and bits IFSR27 and IFSR26 in the IFSR2A register. **Figure 9.4** shows registers IFSR and IFSR2A.

Interrupt Control Register(2)

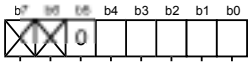
Symbol	Address	After Reset
ICOC0IC	0045_{16}	$XXXXX000_2$
ICOC1IC, IICIC(3)	0046_{16}	$XXXXX000_2$
BTIC, SCLDAIC(3)	0047_{16}	$XXXXX000_2$
BCNIC	$004A_{16}$	$XXXXX000_2$
DM0IC, DM1IC	$004B_{16}$, $004C_{16}$	$XXXXX000_2$
ADIC, KUPIC(3)	$004E_{16}$	$XXXXX000_2$
S0TIC to S2TIC	0051_{16}, 0053_{16}, $004F_{16}$	$XXXXX000_2$
S0RIC to S2RIC	0052_{16}, 0054_{16}, 0050_{16}	$XXXXX000_2$
TA0IC to TA4IC	0055_{16} to 0059_{16}	$XXXXX000_2$
TB0IC to TB2IC	$005A_{16}$ to $005C_{16}$	$XXXXX000_2$



Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	RW(1)
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the contents are undefined		—

NOTES:
1. This bit can only be reset by writing 0 (Do not write 1).
2. To rewrite the interrupt control registers, do so at a point that does not generate the interrupt request for that register. For details, refer to **21. 4 Interrupts**.
3. Use the IFSR2A register to select.

Symbol	Address	After Reset
INT3IC	0044_{16}	$XX00X000_2$
S4IC, INT5IC	0048_{16}	$XX00X000_2$
S3IC, INT4IC	0049_{16}	$XX00X000_2$
INT0IC to INT2IC	$005D_{16}$ to $005F_{16}$	$XX00X000_2$

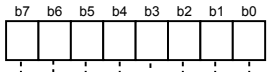


Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0: Interrupt not requested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

NOTES:
1. This bit can only be reset by writing 0 (Do not write 1).
2. To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. For details, refer to **21.4 Interrupts**.
3. If the IFSRi bit in the IFSR register (i = 0 to 5) is 1 (both edges), set the POL bit in the INTiIC register to 0 (falling edge).
4. Set the POL bit in register S3IC or S4IC to 0 (falling edge) when the IFSR6 bit in the IFSR register is set to 0 (SI/O3 selected) or IFSR7 bit in the IFSR register to 0 (SI/O4 selected), respectively.

Figure 9.3 Interrupt Control Registers

Interrupt Request Cause Select Register

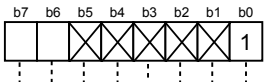


Symbol	Address	After Reset
IFSR	$035F_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR2	INT2 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR3	INT3 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR4	INT4 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR5	INT5 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR6	Interrupt request cause select bit	0 : SI/O 3 (2) 1 : $\overline{INT4}$	RW
IFSR7	Interrupt request cause select bit	0 : SI/O 4 (2) 1 : $\overline{INT5}$	RW

NOTES:
1. When setting this bit to 1 (both edges), make sure the POL bit in registers INT0IC to INT5IC is set to 0 (falling edge).
2. When setting this bit to 0 (SI/O3, SI/O4), make sure the POL bit in registers S3IC and S4IC is set to 0 (falling edge).

Interrupt Request Cause Select Register 2



Symbol	Address	After Reset
IFSR2A	$035E_{16}$	$00XXXXX0_2$

Bit Symbol	Bit Name	Function	RW
IFSR20	Reserved bit (1)	Set to 1	RW
$\overline{\text{(b5-b1)}}$	Nothing is assigned. If necessary, set to 0 When read, their contents are undefined		—
IFSR26	Interrupt request cause select bit	0: IC/OC base timer 1: SCL/SDA	RW
IFSR27	Interrupt request cause select bit	0: IC/OC interrupt 1 1: I^2C bus interface	RW

NOTE:
1. After reset, set this bit to 1 to enable interrupt.

Figure 9.4 IFSR Register and IFSR2A Register

9.3.1 I Flag

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

9.3.2 IR Bit

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

9.3.3 ILVL2 to ILVL0 Bits and IPL

Interrupt priority levels can be set using bits ILVL2 to ILVL0.
Table 9.3 shows the settings of interrupt priority levels and **Table 9.4** shows the interrupt priority levels enabled by the IPL.

The following are conditions under which an interrupt is accepted:
· I flag = 1
· IR bit = 1
· interrupt priority level > IPL

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

Table 9.3 Settings of Interrupt Priority Levels

ILVL2 to ILVL0 bits	Interrupt priority level	Priority order
000_2	Level 0 (interrupt disabled)	———
001_2	Level 1	Low ↓ High
010_2	Level 2	
011_2	Level 3	
100_2	Level 4	
101_2	Level 5	
110_2	Level 6	
111_2	Level 7	

Table 9.4 Interrupt Priority Levels Enabled by IPL

IPL	Enabled interrupt priority levels
000_2	Interrupt levels 1 and above are enabled
001_2	Interrupt levels 2 and above are enabled
010_2	Interrupt levels 3 and above are enabled
011_2	Interrupt levels 4 and above are enabled
100_2	Interrupt levels 5 and above are enabled
101_2	Interrupt levels 6 and above are enabled
110_2	Interrupt levels 7 and above are enabled
111_2	All maskable interrupts are disabled

9.4 Interrupt Sequence

An interrupt sequence (the device behavior from the instant an interrupt is accepted to the instant the interrupt routine is executed) is described here.

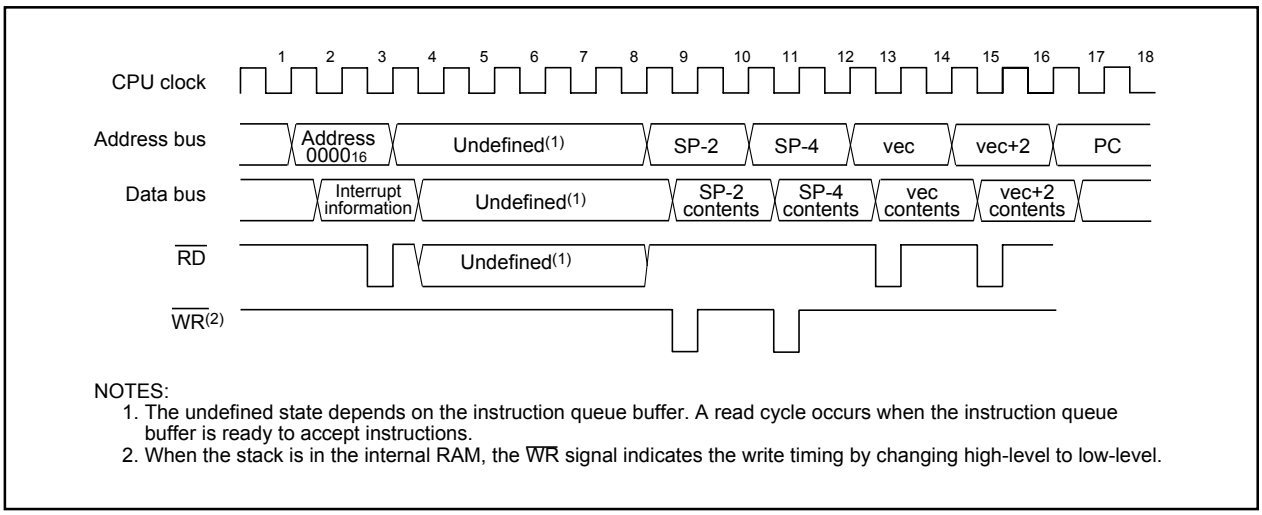
If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

The CPU behavior during the interrupt sequence is described below. **Figure 9.5** shows time required for executing the interrupt sequence.

(1) The CPU gets interrupt information (interrupt number and interrupt request priority level) by reading the address 0000_{16}. Then it clears the IR bit for the corresponding interrupt to 0 (interrupt not requested).
(2) The FLG register immediately before entering the interrupt sequence is saved to the CPU's internal temporary register(Note).
(3) The I, D and U flags in the FLG register become as follows:
The I flag is cleared to 0 (interrupts disabled).
The D flag is cleared to 0 (single-step interrupt disabled).
The U flag is cleared to 0 (ISP selected).
However, the U flag does not change state if an INT instruction for software interrupt Nos. 32 to 63 is executed.
(4) The CPU's internal temporary register(1) is saved to the stack.
(5) The PC is saved to the stack.
(6) The interrupt priority level of the accepted interrupt is set in the IPL.
(7) The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

After the interrupt sequence is completed, the processor resumes executing instructions from the start address of the interrupt routine.

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

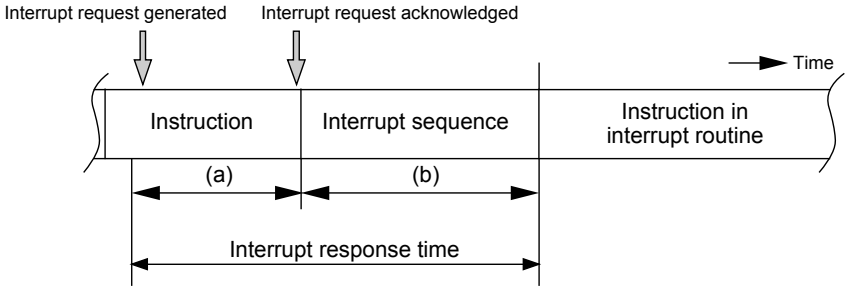


NOTES:
1. The undefined state depends on the instruction queue buffer. A read cycle occurs when the instruction queue buffer is ready to accept instructions.
2. When the stack is in the internal RAM, the $\overline{WR}$ signal indicates the write timing by changing high-level to low-level.

Figure 9.5 Time Required for Executing Interrupt Sequence

9.4.1 Interrupt Response Time

Figure 9.6 shows the interrupt response time. The interrupt response or interrupt acknowledge time denotes time from when an interrupt request is generated till when the first instruction in the interrupt routine is executed. Specifically, it consists of the time from when an interrupt request is generated till when the instruction then executing is completed ((a) in **Figure 9.6**) and the time during which the interrupt sequence is executed ((b) in **Figure 9.6**).



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

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

Interrupt vector address	SP value	Without wait
Even	Even	18 cycles
Even	Odd	19 cycles
Odd	Even	19 cycles
Odd	Odd	20 cycles

Figure 9.6 Interrupt response time

9.4.2 Variation of IPL when Interrupt Request is Accepted

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

When a software interrupt or special interrupt request is accepted, one of the interrupt priority levels listed in **Table 9.5** is set in the IPL. Shown in **Table 9.5** are the IPL values of software and special interrupts when they are accepted.

Table 9.5 IPL Level That is Set to IPL When A Software or Special Interrupt Is Accepted

Interrupt sources	IPL setting
Watchdog timer, $\overline{\text{NMI}}$, Oscillation stop and re-oscillation detection	7
Software, address match, $\overline{\text{DBC}}$, single-step	No change

9.4.3 Saving Registers

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

At this time, the 4 high-order bits of the PC and the 4 high-order (IPL) and 8 low-order bits of the FLG register, 16 bits in total, are saved to the stack first. Next, the 16 low-order bits of the PC are saved. **Figure 9.7** shows the stack status before and after an interrupt request is accepted.

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

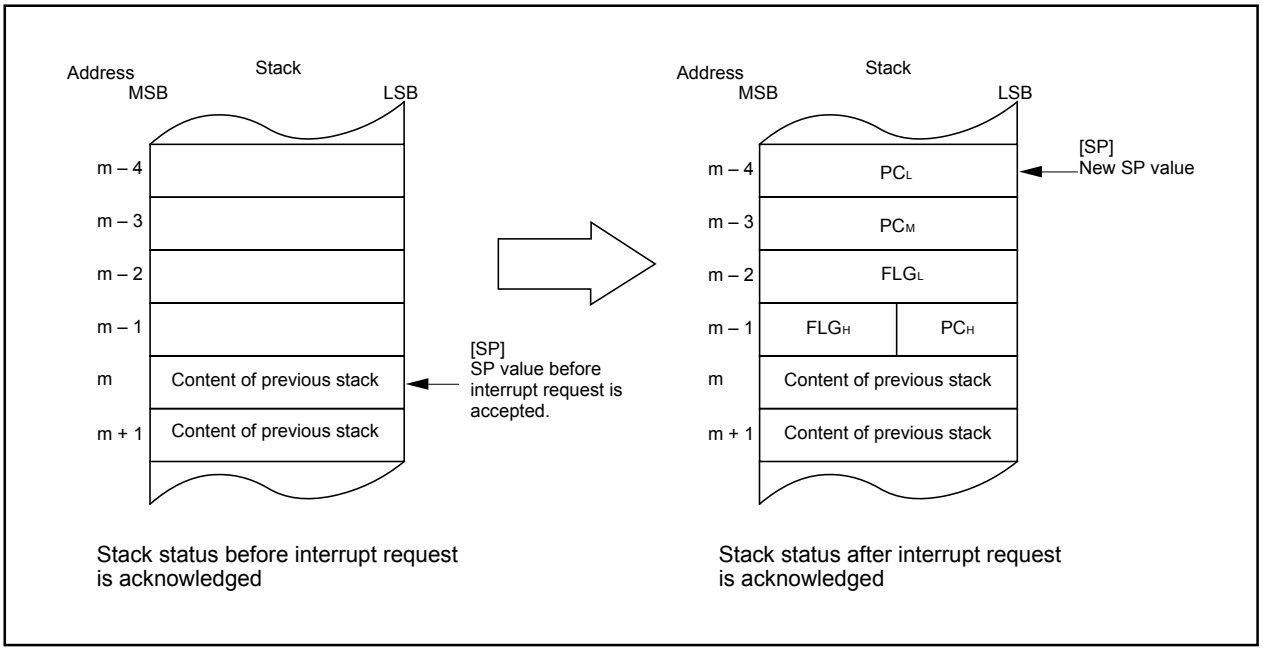


Figure 9.7 Stack Status Before and After Acceptance of Interrupt Request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP(1), at the time of acceptance of an interrupt request, is even or odd. If the stack pointer (1) is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. **Figure 9.8** shows the operation of the saving registers.

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

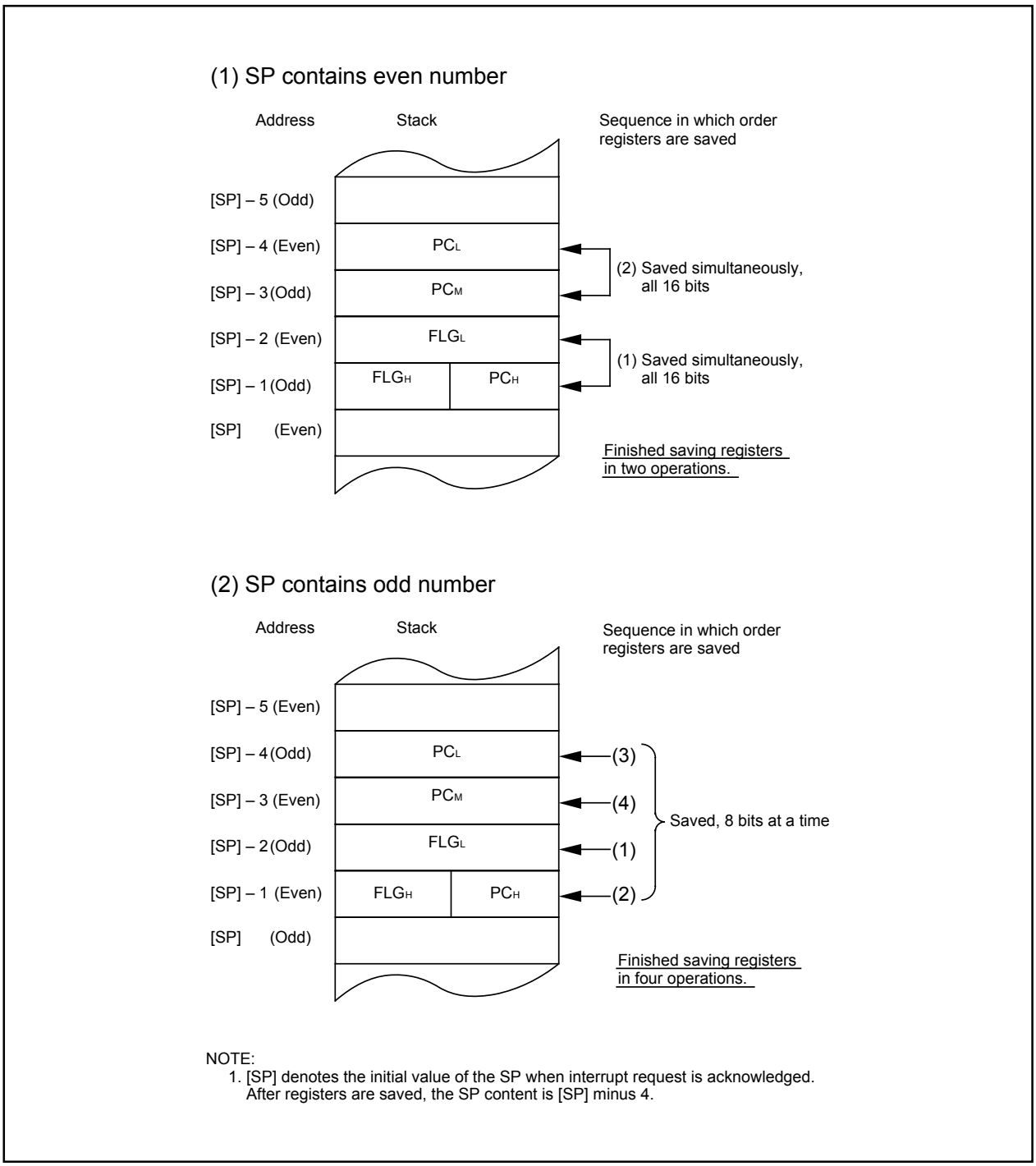


Figure 9.8 Operation of Saving Register

9.4.4 Returning from an Interrupt Routine

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

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

9.5 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt request that has the highest priority is accepted.

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

The watchdog timer and other special interrupts have their priority levels set in hardware. **Figure 9.9** shows the priorities of hardware interrupts.

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

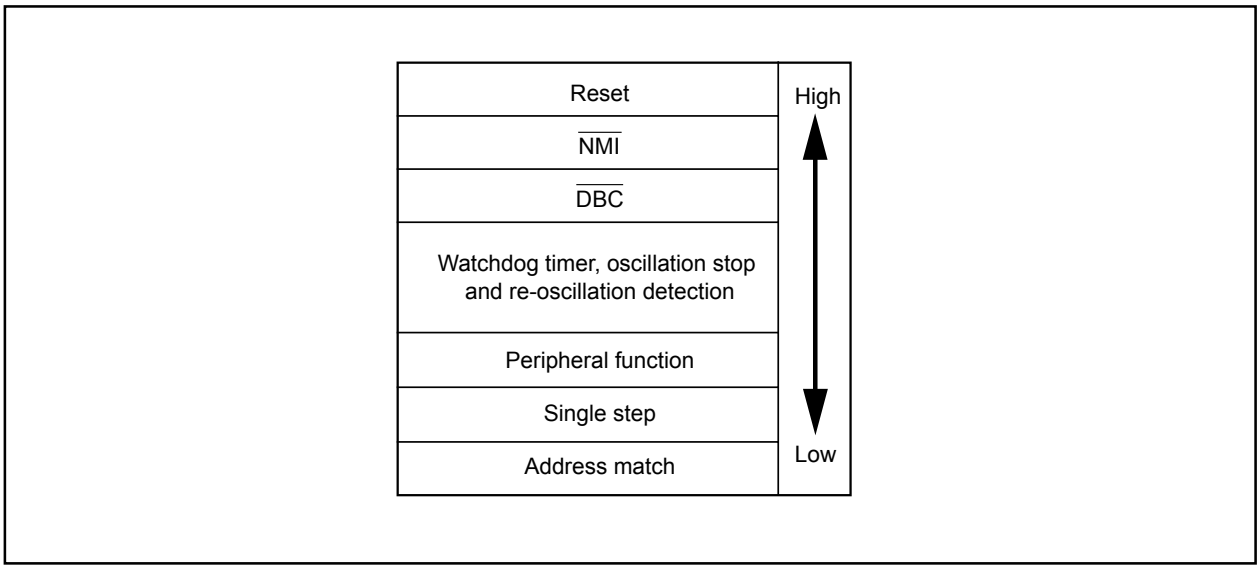


Figure 9.9 Hardware Interrupt Priority

9.5.1 Interrupt Priority Resolution Circuit

The interrupt priority resolution circuit is used to select the interrupt with the highest priority among those requested.

Figure 9.10 shows the circuit that judges the interrupt priority level.

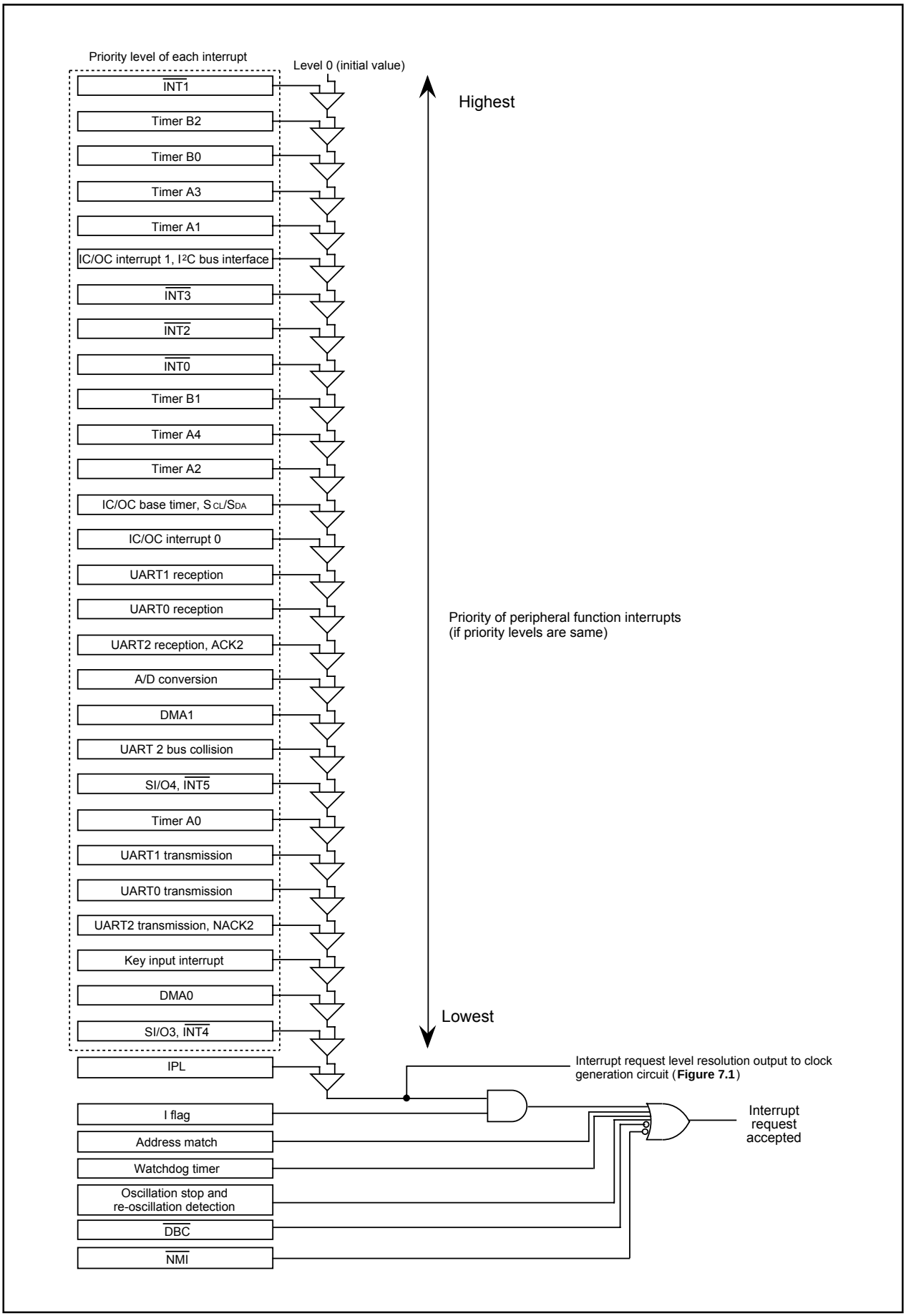


Figure 9.10 Interrupts Priority Select Circuit

9.6 $\overline{\text{INT}}$ Interrupt

$\overline{\text{INT}}$i interrupt (i=0 to 5) is triggered by the edges of external inputs. The edge polarity is selected using the IFSRi bit in the IFSR register.

The $\overline{\text{INT5}}$ input has an effective digital debounce function for a noise rejection. Refer to "**18.6 Digital Debounce function**" for this detail. To use $\overline{\text{INT5}}$ interrupt to exit stop mode, set the P17DDR register to FF_{16} before entering stop mode.

To use the $\overline{\text{INT4}}$ interrupt, set the IFSR6 bit in the IFSR register to 1 ($\overline{\text{INT4}}$). To use the $\overline{\text{INT5}}$ interrupt, set the IFSR7 bit in the IFSR register to 1 ($\overline{\text{INT5}}$).

After modifiying bit IFSR6 or IFSR7, clear the corresponding IR bit to 0 (interrupt not requested) before enabling the interrupt.

Figure 9.11 shows the IFSR registers.

Interrupt Request Cause Select Register

Symbol	Address	After Reset
IFSR	$035F_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR2	INT2 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR3	INT3 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR4	INT4 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR5	INT5 interrupt polarity switching bit	0 : One edge 1 : Both edges (1)	RW
IFSR6	Interrupt request cause select bit	0 : SI/O 3 (2) 1 : $\overline{\text{INT4}}$	RW
IFSR7	Interrupt request cause select bit	0 : SI/O 4 (2) 1 : $\overline{\text{INT5}}$	RW

NOTES:
1. When setting this bit to 1 (both edges), make sure the POL bit in registers INT0IC to INT5IC is set to 0 (falling edge).
2. When setting this bit to 0 (SI/O3, SI/O4), make sure the POL bit in registers S3IC and S4IC is set to 0 (falling edge).

Figure 9.11 IFSR Register

9.7 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt request is generated when input on the $\overline{\text{NMI}}$ pin changes state from high to low, after the $\overline{\text{NMI}}$ interrupt was enabled by writing a 1 to bit 4 in the register PM2. The $\overline{\text{NMI}}$ interrupt is a non-maskable interrupt, once it is enabled.

The input level of this $\overline{\text{NMI}}$ interrupt input pin can be read by accessing the P8_5 bit in the P8 register.

$\overline{\text{NMI}}$ is disabled by default after reset (the pin is a GPIO pin, $P8_5$) and can be enabled using bit 4 in the PM2 register. Once enabled, it can only be disabled by a reset signal.

The $\overline{\text{NMI}}$ input has a digital debounce function for noise rejection. Refer to "**19.6 Digital Debounce function**" for details. To use $\overline{\text{NMI}}$ interrupt to exit stop mode, set the NDDR register to FF_{16} before entering stop mode.

9.8 Key Input Interrupt

A key input interrupt is generated when input on any of the $P10_4$ to $P10_7$ pins which has had bits PD10_7 to PD10_4 in the PD10 register set to 0 (= input) goes low. Key input interrupts can be used for a key-on wakeup function to get the MCU to exit stop or wait modes. However, if you intend to use the key input interrupt, do not use $P10_4$ to $P10_7$ as analog input ports. **Figure 9.12** shows the block diagram of the key input interrupt. Note, however, that while input on any pin which has had bits PD10_7 to PD10_4 set to 0 (= input mode) is pulled low, inputs on all other pins of the port are not detected as interrupts.

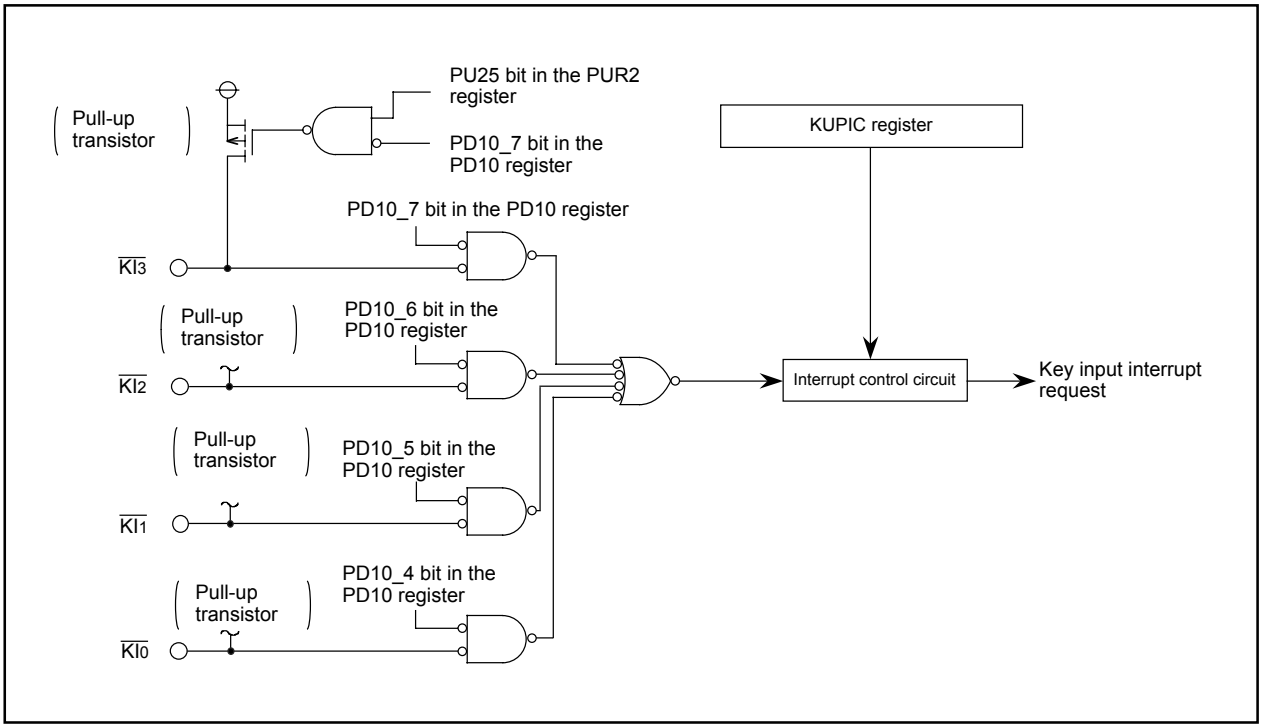


Figure 9.12 Key Input Interrupt

9.9 Address Match Interrupt

An address match interrupt request is generated immediately before executing the instruction at the address indicated by the RMADi register (i=0 to 1). Set the start address of any instruction in the RMADi register. Use bits AIER1 and AIER0 in the AIER register to enable or disable the interrupt. Note that the address match interrupt is unaffected by the I flag and IPL. For address match interrupts, the value of the PC that is saved to the stack area varies depending on the instruction being executed (refer to "**Saving Registers**").

(The value of the PC that is saved to the stack area is not the correct return address.) Therefore, follow one of the methods described below to return from the address match interrupt.

• Rewrite the content of the stack and then use the REIT instruction to return.

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

Table 9.6 shows the value of the PC that is saved to the stack area when an address match interrupt request is accepted.

Figure 9.13 shows registers AIER, RMAD0, and RMAD1.

Table 9.6 Value of the PC that is saved to the stack area when an address match interrupt request is accepted.

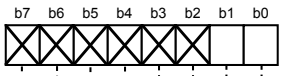
Instruction at the address indicated by the RMADi register	Value of the PC that is saved to the stack area
• 2-byte op-code instruction • 1-byte op-code instructions which are followed: ADD.B:S #IMM8,dest SUB.B:S #IMM8,dest AND.B:S #IMM8,dest OR.B:S #IMM8,dest MOV.B:S #IMM8,dest STZ.B #IMM8,dest STNZ.B #IMM8,dest STZX.B #IMM81,#IMM82,dest CMP.B:S #IMM8,dest PUSHM src POPM dest JMPS #IMM8 JSRS #IMM8 MOV.B:S #IMM,dest (However, dest=A0 or A1)	The address indicated by the RMADi register +2
Instructions other than the above	The address indicated by the RMADi register +1

Value of the PC that is saved to the stack area : Refer to "Saving Registers".
Op-code is an abbreviation of Operation Code. It is a portion of instruction code.
Refer to Chapter 4 Instruction Code/Number of Cycles in M16C/60, M16C/20 Series Software Manual. Op-code is shown as a bold-framed figure directly below the Syntax.

Table 9.7 Relationship Between Address Match Interrupt Sources and Associated Registers

Address match interrupt sources	Address match interrupt enable bit	Address match interrupt register
Address match interrupt 0	AIER0	RMAD0
Address match interrupt 1	AIER1	RMAD1

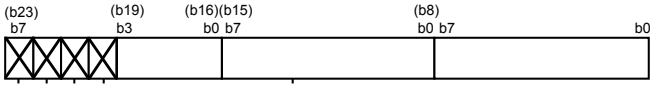
Address Match Interrupt Enable Register



Symbol	Address	After Reset
AIER	0009_{16}	$XXXXXX00_2$

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
$\overline{\text{(b7-b2)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—

Address Match Interrupt Register i (i = 0 to 1)



Symbol	Address	After Reset
RMAD0	0012_{16} to 0010_{16}	$X0000_{16}$
RMAD1	$0016_{16}$to 0014_{16}	$X0000_{16}$

Function	Setting Range	RW
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—

Figure 9.13 AIER Register, RMAD0 and RMAD1 Registers

10. Watchdog Timer

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

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

With main clock source chosen for CPU clock, on-chip oscillator clock, PLL clock

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

With sub-clock chosen for CPU clock

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

For example, when CPU clock is set to 16 MHz and the divide-by-N value for the prescale ris set to 16, the watchdog timer period is approx. 32.8 ms.

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

Write the WDTS register with shorter cycle than the watchdog timer cycle. Set the WDTS register also in the beginning of the watchdog timer interrupt routine.

The watchdog timer and prescaler stop in stop mode and wait mode. When the modes are exited counting is resumed from the held value .

Figure 10.1 shows the block diagram of the watchdog timer. Figure 10.2 shows the WDC register and the WDTS register.

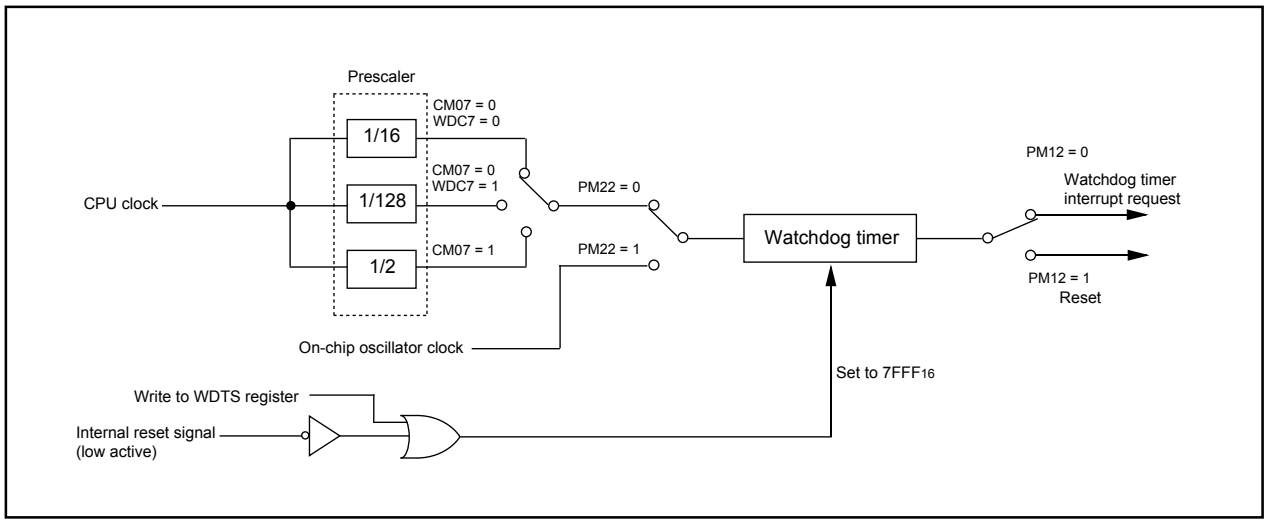


Figure 10.1 Watchdog Timer Block Diagram

Watchdog Timer Control Register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	After Reset
WDC	$000F_{16}$	$00XXXXXX_2$

Bit Symbol	Bit Name	Function	RW
(b4-b0)	High-order bit of watchdog timer		RO
$\overline{(b5)}$	Reserved bit	Set to 0	RW
$\overline{(b6)}$	Reserved bit	Set to 0	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Watchdog Timer Start Register

b7	b0

Symbol	Address	After Reset
WDTS	$000E_{16}$	Undefined

Function	RW
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to $7FFF_{16}$ regardless of whatever value is written.	WO

Figure 10.2 WDC Register and WDTS Register

10.1 Count Source Protective Mode

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

Before this mode can be used, the following register settings are required:

(1) Set the PRC1 bit in the PRCR register to 1 (enable writes to PM1 and PM2 registers).

(2) Set the PM12 bit in the PM1 register to 1 (reset when the watchdog timer underflows).

(3) Set the PM22 bit in the PM2 register to 1 (on-chip oscillator clock used for the watchdog timer count source).

(4) Set the PRC1 bit in the PRCR register to 0 (disable writes to PM1 and PM2 registers).

(5) Write to the WDTS register (watchdog timer starts counting).

Setting the PM22 bit to 1 results in the following conditions

- The on-chip oscillator continues oscillating even if the CM21 bit in the CM2 register is set to "0" (main clock or PLL clock) (system clock of count source selected by the CM21 bit is valid)
- The on-chip oscillator starts oscillating, and the on-chip oscillator clock becomes the watchdog timer count source.

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

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

11. DMAC

Note

Do not use SI/O4 interrupt request as a DMA request in the 64-pin package.

The DMAC (Direct Memory Access Controller) allows data to be transferred without the CPU intervention. Two DMAC channels are included. Each time a DMA request occurs, the DMAC transfers one (8 or 16-bit) data from the source address to the destination address. The DMAC uses the same data bus as used by the CPU. Because the DMAC has higher priority of bus control than the CPU and because it makes use of a cycle steal method, it can transfer one word (16 bits) or one byte (8 bits) of data within a very short time after a DMA request is generated. **Figure 11.1** shows the block diagram of the DMAC. **Table 11.1** shows the DMAC specifications. **Figures 11.2** to **11.4** show the DMAC-related registers.

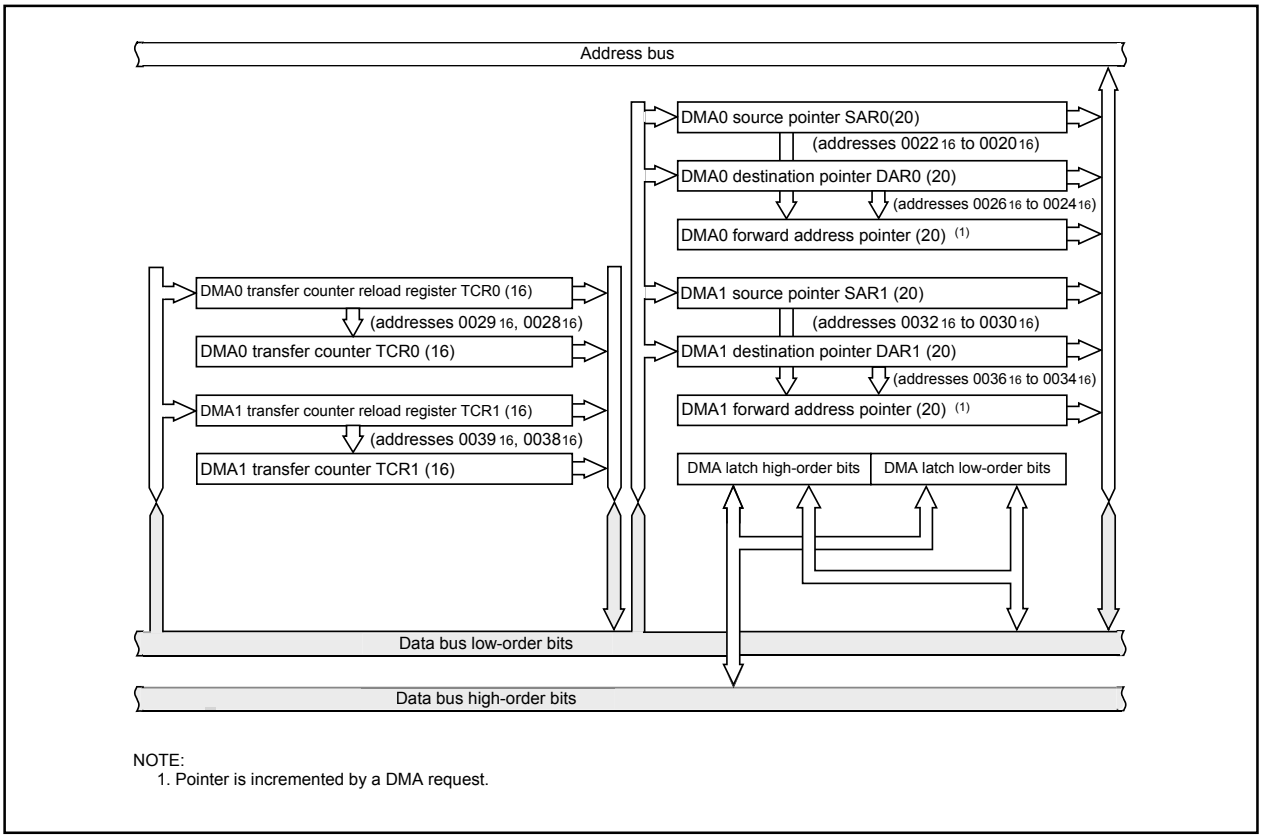


Figure 11.1 DMAC Block Diagram

A DMA request is generated by setting the DSR bit in the DMiSL register (i = 0,1), as well as by an interrupt request which is generated by any function specified by bits DMS, DSEL3, DSEL2, DSEL1, and DSEL0 in the DMiSL register. However, unlike in the case of interrupt requests, DMA requests are not affected by the I flag and the interrupt control register, so that even when interrupt requests are disabled and no interrupt request can be accepted, DMA requests are always accepted. Furthermore, because the DMAC does not affect interrupts, the IR bit in the interrupt control register does not change state due to a DMA transfer.

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

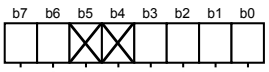
Table 11.1 DMAC Specifications

Item		Specification
No. of channels		2 (cycle steal method)
Transfer memory space		• From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address
Maximum No. of bytes transferred		128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (1, 2)		Falling edge of $\overline{INT0}$ or $\overline{INT1}$ Both edge of $\overline{INT0}$ or $\overline{INT1}$ Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transfer, UART0 reception interrupt requests UART1 transfer, UART1 reception interrupt requests UART2 transfer, UART2 reception interrupt requests SI/O3, SI/O4 interrupt requests A/D conversion interrupt requests Timer S(IC/OC) requests Software triggers
Channel priority		DMA0 > DMA1 (DMA0 takes precedence)
Transfer unit		8 bits or 16 bits
Transfer address direction		forward or fixed (The source and destination addresses cannot both be in the forward direction)
Transfer mode	Single transfer	Transfer is completed when the DMAi transfer counter (i = 0,1) underflows after reaching the terminal count
	Repeat transfer	When the DMAi transfer counter underflows, it is reloaded with the value of the DMAi transfer counter reload register and a DMA transfer is con tinued with it
DMA interrupt request generation timing		When the DMAi transfer counter underflowed
DMA startup the		Data transfer is initiated each time a DMA request is generated when DMAE bit in the DMAiCON register = 1 (enabled)
DMA shutdown	Single transfer	• When the DMAE bit is set to 0 (disabled) • After the DMAi transfer counter underflows
	Repeat transfer	When the DMAE bit is set to 0 (disabled)
Reload timing for forward address pointer and transfer counter		When a data transfer is started after setting the DMAE bit to 1 (en abled), the forward address pointer is reloaded with the value of the SARi or the DARi pointer whichever is specified to be in the forward direction and the DMAi transfer counter is reloaded with the value of the DMAi transfer counter reload register

NOTES:

1. DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the I flag nor by the interrupt control register.
2. The selectable causes of DMA requests differ with each channel.
3. Make sure that no DMAC-related registers (addresses 0020_{16} to $003F_{16}$) are accessed by the DMAC.

DMA0 Request Cause Select Register



Symbol	Address	After Reset
DM0SL	03B8$_{16}$	00$_{16}$

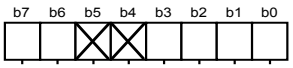
Bit Symbol	Bit Name	Function	RW
DSEL0	DMA request cause select bit	Refer to note (1)	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
—— (b5-b4)	Nothing is assigned. When write, set to 0. When read, their content are 0		——
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to 1 when the DMS bit is 0 (basic cause) and bits DSEL3 to DSEL0 are 0001$_2$ (software trigger). The value of this bit when read is 0	RW

NOTE:
1. The causes of DMA0 requests can be selected by a combination of DMS bit and bits DSEL3 to DSEL0 in the manner described below.

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 0$_2$	Falling edge of $\overline{\text{INT0}}$ pin	IC/OC base timer
0 0 0 1$_2$	Software trigger	–
0 0 1 0$_2$	Timer A0	IC/OC channel 0
0 0 1 1$_2$	Timer A1	IC/OC channel 1
0 1 0 0$_2$	Timer A2	–
0 1 0 1$_2$	Timer A3	–
0 1 1 0$_2$	Timer A4	Two edges of $\overline{\text{INT0}}$ pin
0 1 1 1$_2$	Timer B0	–
1 0 0 0$_2$	Timer B1	–
1 0 0 1$_2$	Timer B2	–
1 0 1 0$_2$	UART0 transmit	IC/OC channel 2
1 0 1 1$_2$	UART0 receive	IC/OC channel 3
1 1 0 0$_2$	UART2 transmit	IC/OC channel 4
1 1 0 1$_2$	UART2 receive	IC/OC channel 5
1 1 1 0$_2$	A/D conversion	IC/OC channel 6
1 1 1 1$_2$	UART1 transmit	IC/OC channel 7

Figure 11.2 DM0SL Register

DMA1 Request Cause Select Register



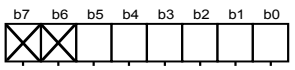
Symbol	Address	After Reset
DM1SL	03BA16	0016

Bit Symbol	Bit Name	Function	RW
DSEL0	DMA request cause select bit	Refer to note (1)	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
—— (b5-b4)	Nothing is assigned. If necessary, set to 0. When read, their contents are 0		——
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to 1 when the DMS bit is 0 (basic cause) and the DSEL3 to DSEL0 bits are 0001 2 (software trigger). The value of this bit when read is 0	RW

NOTES:
1. The causes of DMA1 requests can be selected by a combination of DMS bit and bits DSEL3 to DSEL0 in the manner described below.

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 0 2	Falling edge of INT1 pin	IC/OC base timer
0 0 0 1 2	Software trigger	–
0 0 1 0 2	Timer A0	IC/OC channel 0
0 0 1 1 2	Timer A1	IC/OC channel 1
0 1 0 0 2	Timer A2	–
0 1 0 1 2	Timer A3	SI/O3
0 1 1 0 2	Timer A4	SI/O4
0 1 1 1 2	Timer B0	Two edges of INT1
1 0 0 0 2	Timer B1	–
1 0 0 1 2	Timer B2	–
1 0 1 0 2	UART0 transmit	IC/OC channel 2
1 0 1 1 2	UART0 receive	IC/OC channel 3
1 1 0 0 2	UART2 transmit	IC/OC channel 4
1 1 0 1 2	UART2 receive/ACK2	IC/OC channel 5
1 1 1 0 2	A/D conversion	IC/OC channel 6
1 1 1 1 2	UART1 receive	IC/OC channel 7

DMAi Control Register(i=0,1)



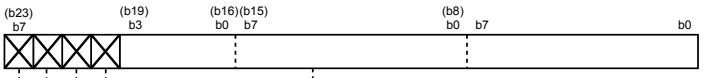
Symbol	Address	After Reset
DM0CON	002C16	00000X002
DM1CON	003C16	00000X002

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

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

Figure 11.3 DM1SL Register, DM0CON Register, and DM1CON Registers

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

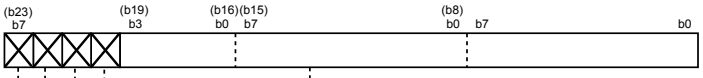


Symbol	Address	After Reset
SAR0	0022₁₆ to 0020₁₆	Undefined
SAR1	0032₁₆ to 0030₁₆	Undefined

Function	Setting Range	RW
Set the source address of transfer	00000₁₆ to FFFFF₁₆	RW
Nothing is assigned. If necessary, set 0. When read, the contents are 0		—

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

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

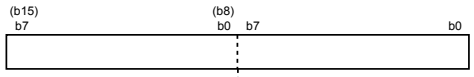


Symbol	Address	After Reset
DAR0	0026₁₆ to 0024₁₆	Undefined
DAR1	0036₁₆ to 0034₁₆	Undefined

Function	Setting Range	RW
Set the destination address of transfer	00000₁₆ to FFFFF₁₆	RW
Nothing is assigned. If necessary, set 0. When read, the contents are 0		—

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

DMAi Transfer Counter (i = 0, 1)



Symbol	Address	After Reset
TCR0	0029₁₆, 0028₁₆	Undefined
TCR1	0039₁₆, 0038₁₆	Undefined

Function	Setting Range	RW
Set the transfer count minus 1. The written value is stored in the DMAi transfer counter reload register, and when the DMAE bit in the DMiCON register is set to 1 (DMA enabled) or the DMAi transfer counter underflows when the DMASL bit in the DMiCON register is 1 (repeat transfer), the value of the DMAi transfer counter reload register is transferred to the DMAi transfer counter. When read, the DMAi transfer counter is read	0000₁₆ to FFFF₁₆	RW

Figure 11.4 SAR0, SAR1, DAR0, DAR1, TCR0, and TCR1 Registers

11.1 Transfer Cycles

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. Furthermore, the bus cycle itself is extended by a software wait.

11.1.1 Effect of Source and Destination Addresses

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

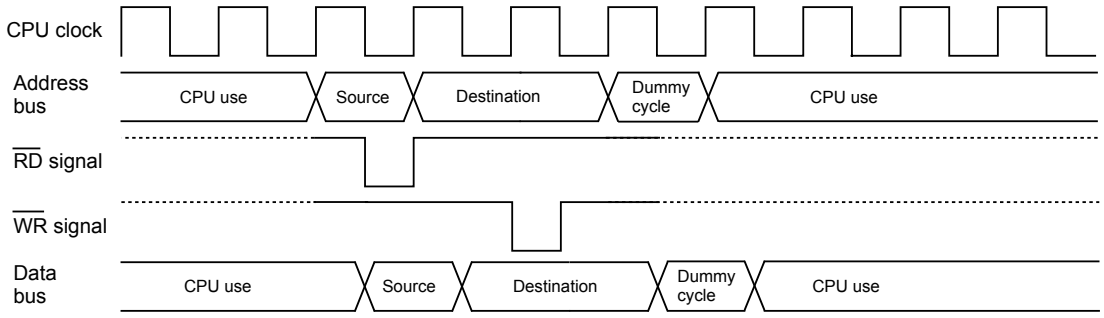
Similarly, if the transfer unit is 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

11.1.2 Effect of Software Wait

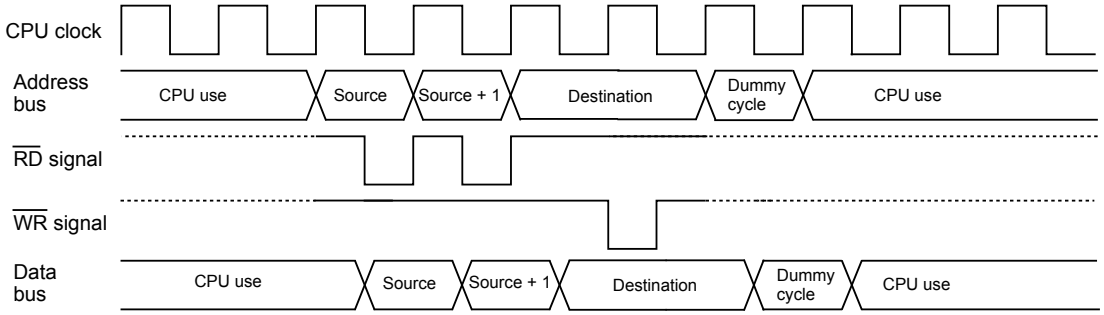
For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

Figure 11.5 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units and when both the source address and destination address are an odd address ((2) in **Figure 11.5**), two source read bus cycles and two destination write bus cycles are required.

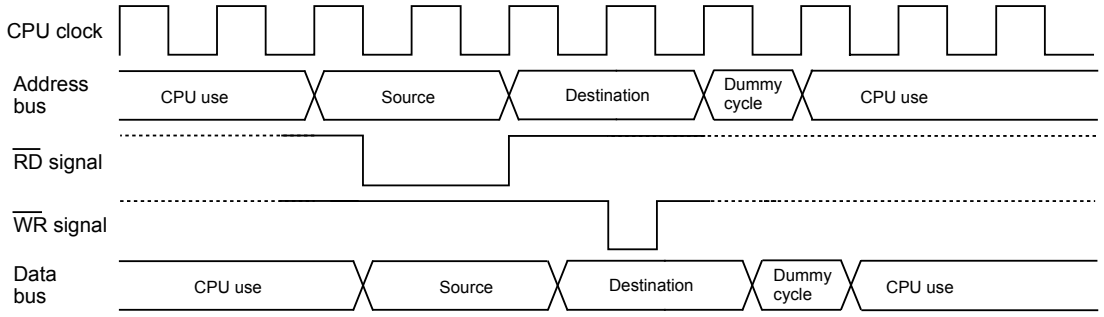
(1) When the transfer unit is 8 or 16 bits and the source of transfer is an even address



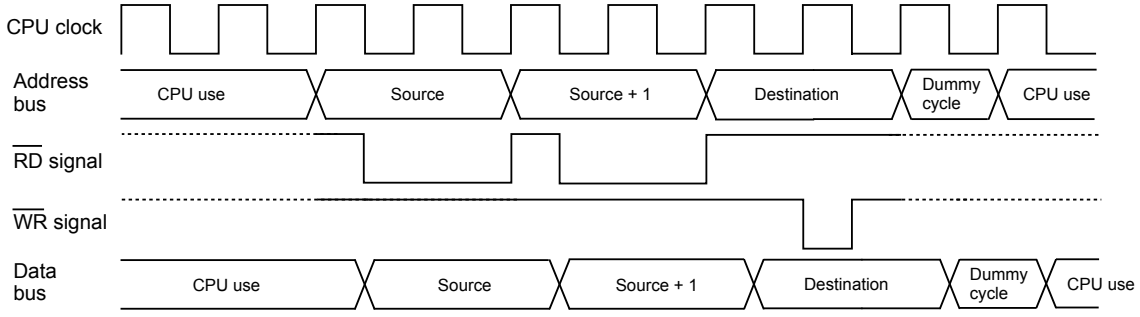
(2) When the transfer unit is 16 bits and the source address of transfer is an odd address.



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



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



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

Figure 11.5 Transfer Cycles for Source Read

11.2 DMA Transfer Cycles

Any combination of even or odd transfer read and write adresses is possible. **Table 11.2** shows the number of DMA transfer cycles. **Table 11.3** shows the Coefficient j, k.
The number of DMAC transfer cycles can be calculated as follows:

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

Table 11.2 DMA Transfer Cycles

Transfer unit	Access address	No. of read cycles	No. of write cycles
8-bit transfers (DMBIT= “1”)	Even	1	1
	Odd	1	1
16-bit transfers (DMBIT= “0”)	Even	1	1
	Odd	2	2

Table 11.3 Coefficient j, k

	Internal Area			
	Internal ROM, RAM		SFR	
	No wait	With wait	1 wait (1)	2 wait (1)
j	1	2	2	3
k	1	2	2	3

NOTE:
1. Depends on the set value of PM20 bit in PM2 register

11.3 DMA Enable

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

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

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

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

(1) Write 1 to bits DMAE and DMAS in DMiCON register simultaneously.

(2) Make sure that the DMAi is in an initial state as described above (a) and (b) by program.

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

11.4 DMA Request

The DMAC can generate a DMA request as triggered by the cause of request that is selected with the DMS bit and bits DSEL3 to DSEL0 in the DMiSL register (i = 0, 1) on either channel. **Table 11.4** shows the timing at which the DMAS bit changes state.

Whenever a DMA request is generated, the DMAS bit is set to 1 (DMA requested) regardless of whether or not the DMAE bit is set. If the DMAE bit was set to 1 (enabled) when this occurred, the DMAS bit is set to 0 (DMA not requested) immediately before a data transfer starts. This bit cannot be set to 1 by program (it can only be set to 0).

The DMAS bit may be set to 1 when the DMS or the DSEL3 to DSEL0 bits change state. Therefore, always be sure to set the DMAS bit to 0 after changing the DMS or the DSEL3 to DSEL0 bits.

Because if the DMAE bit is set to 1, a data transfer starts immediately after a DMA request is generated, the DMAS bit in almost all cases is 0 when read by program. Read the DMAE bit to determine whether the DMAC is enabled.

Table 11.4 Timing at Which the DMAS Bit Changes State

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

11.5 Channel Priority and DMA Transfer Timing

If both DMA0 and DMA1 are enabled and DMA transfer request signals from DMA0 and DMA1 are detected active in the same sampling period (one period from a falling edge to the next falling edge of CPU clock), the DMAS bit on each channel is set to 1 (DMA requested) at the same time. In this case, the DMA requests are arbitrated according to the channel priority, DMA0 > DMA1. The following describes DMAC operation when DMA0 and DMA1 requests are detected active in the same sampling period. **Figure 11.6** shows an example of DMA transfer effected by external factors.

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

In addition, DMA requsts cannot be counted up since each channel has one DMAS bit. Therefore, when DMA requests, as DMA1 in **Figure 11.6** occurs more than one time, the DAMS bit is set to 0 as soon as getting the bus arbitration. The bus arbitration is returned to the CPU when one transfer is completed.

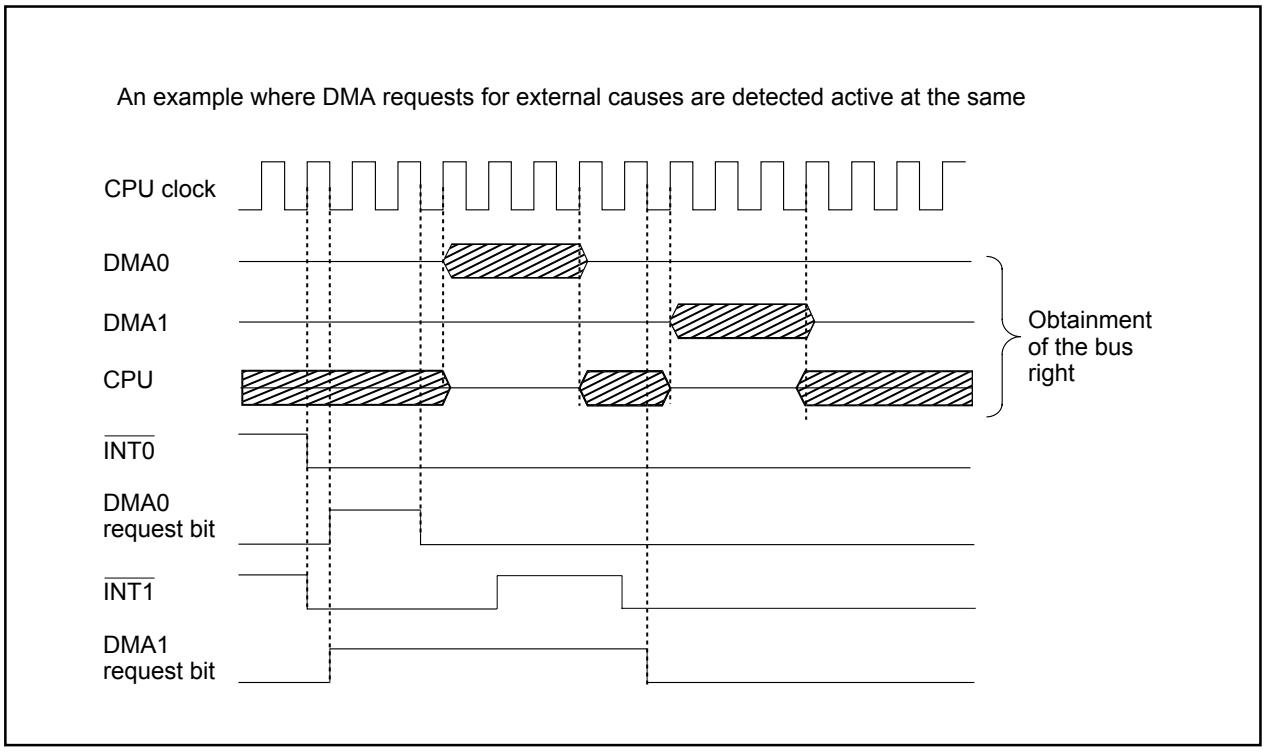


Figure 11.6 DMA Transfer by External Factors

12. Timer

Eight 16-bit timers, each capable of operating independently of the others, can be classified by function as either timer A (five) and timer B (three). The count source for each timer acts as a clock, to control such timer operations as counting, reloading, etc. **Figures 12.1** and **12.2** show block diagrams of timer A and timer B configuration, respectively.

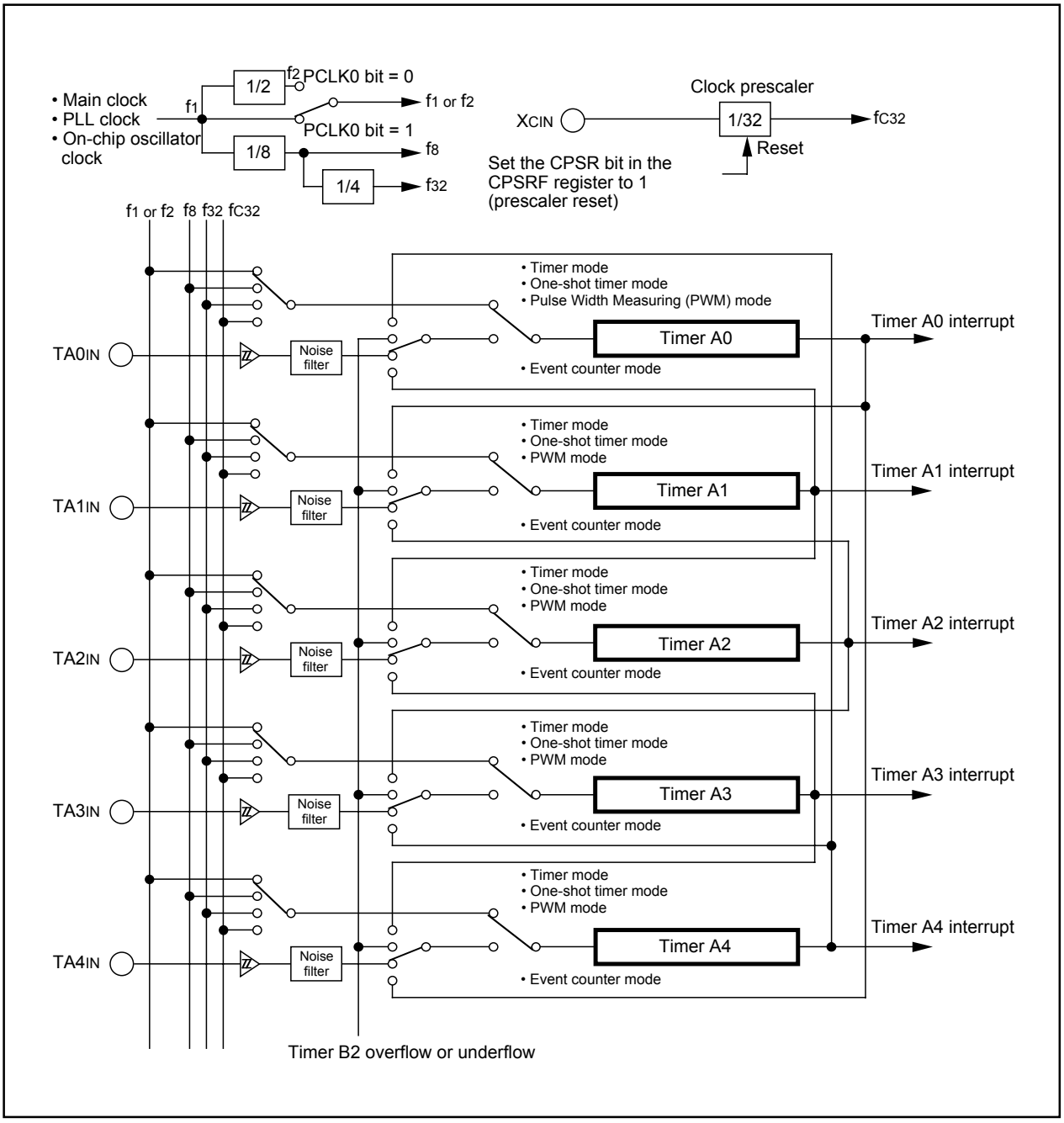


Figure 12.1 Timer A Configuration

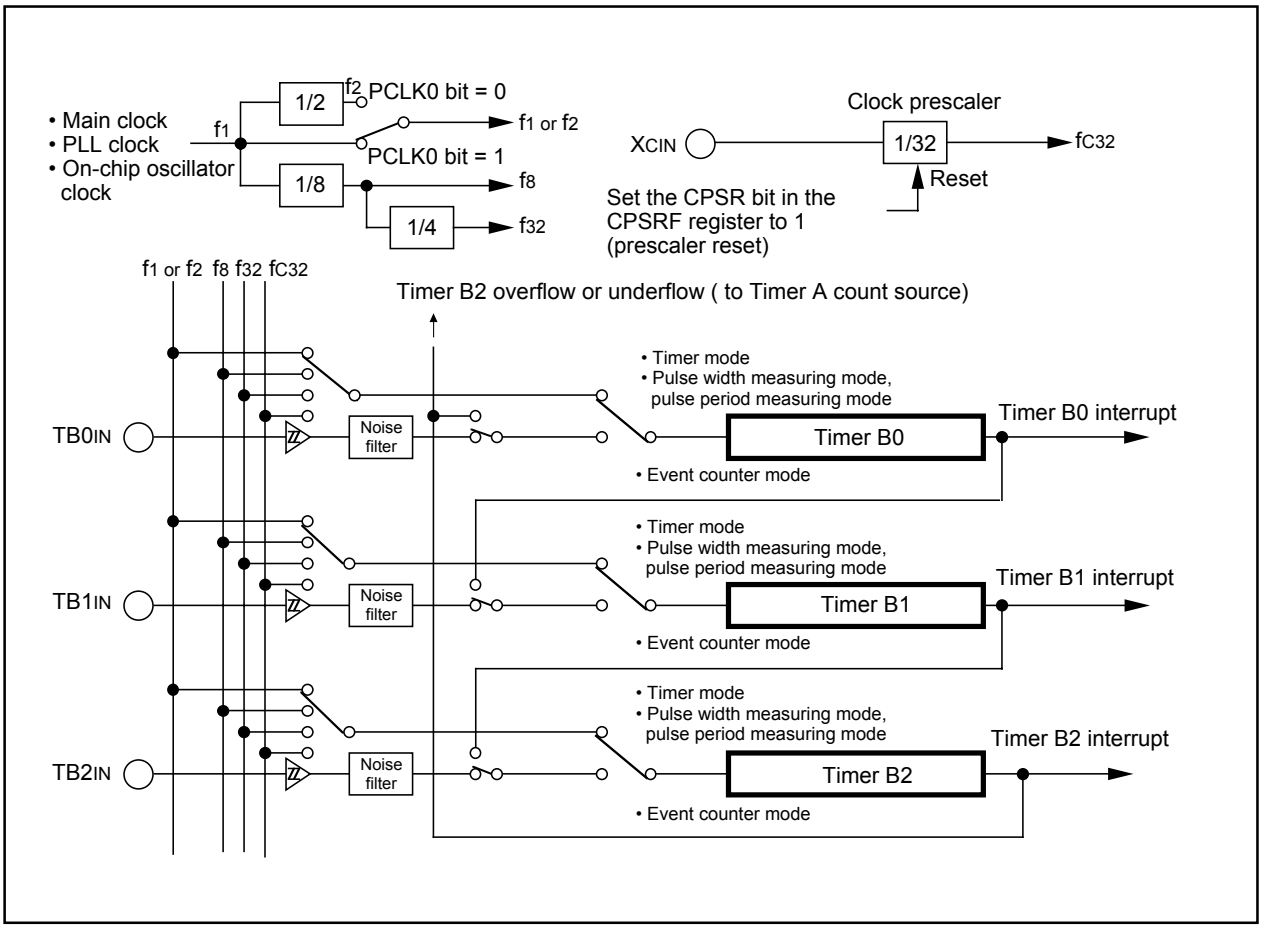


Figure 12.2. Timer B Configuration

12.1 Timer A

Figure 12.3 shows a block diagram of the timer A. **Figures 12.4** to **12.6** show registers related to the timer A. The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use bits TMOD1 to TMOD0 in the TAiMR register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count 0000$_{16}$.
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

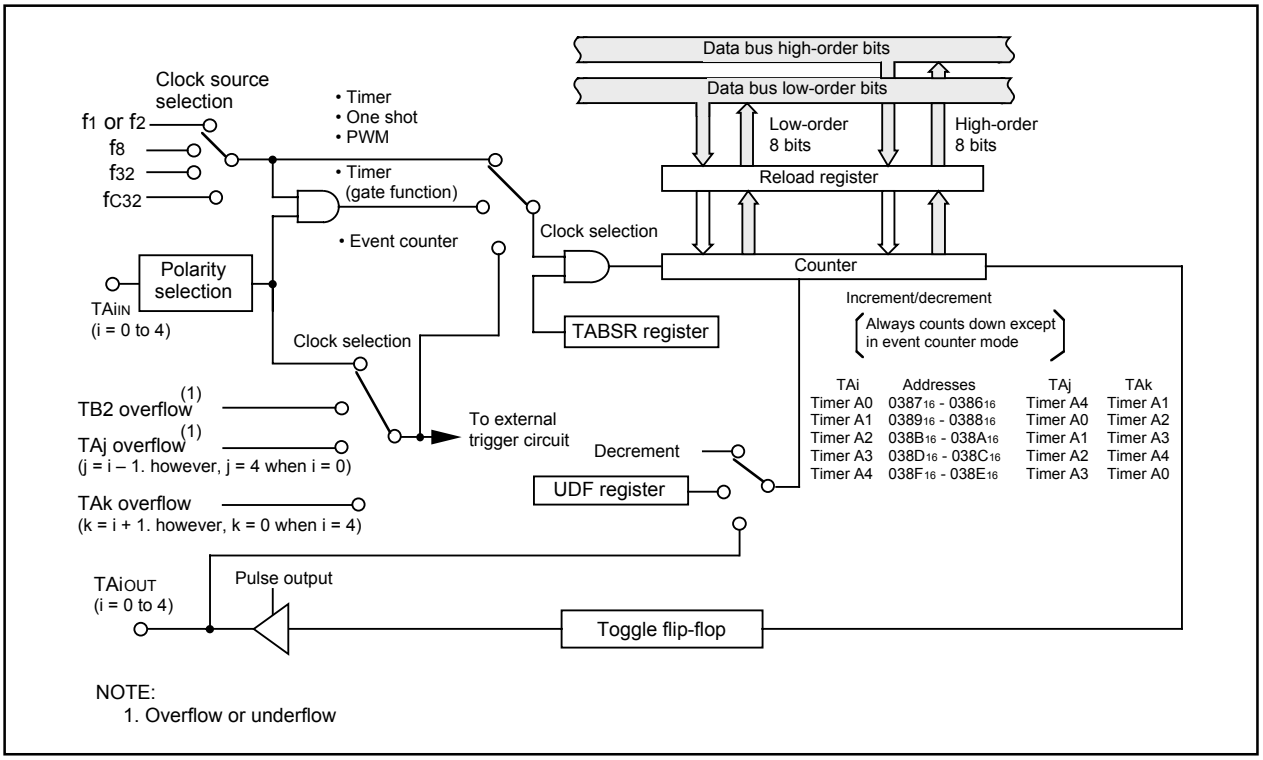


Figure 12.3 Timer A Block Diagram

Timer Ai Mode Register (i=0 to 4)

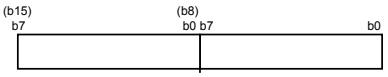
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TA0MR to TA4MR	0396$_{16}$ to 039A$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	RW
TMOD1			RW
MR0	———	Function varies with each operation mode	RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count source select bit	Function varies with each operation mode	RW
TCK1			RW

Figure 12.4 TA0MR to TA4MR Registers

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

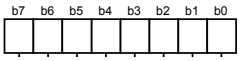


Symbol	Address	After Reset
TA0	0387₁₆, 0386₁₆	Undefined
TA1	0389₁₆, 0388₁₆	Undefined
TA2	038B₁₆, 038A₁₆	Undefined
TA3	038D₁₆, 038C₁₆	Undefined
TA4	038F₁₆, 038E₁₆	Undefined

Mode	Function	Setting Range	RW
Timer mode	Divide the count source by n + 1 where n = set value	0000₁₆ to FFFF₁₆	RW
Event counter mode	Divide the count source by FFFF₁₆ – n + 1 where n = set value when counting up or by n + 1 when counting down(5)	0000₁₆ to FFFF₁₆	RW
One-shot timer mode	Divide the count source by n where n = set value and cause the timer to stop	0000₁₆ to FFFF₁₆ (2, 4)	WO
Pulse width modulation mode (16-bit PWM)	Modify the pulse width as follows: PWM period: (2^{16} – 1) / fj High level PWM pulse width: n / fj where n = set value, fj = count source frequency	0000₁₆ to FFFE₁₆ (3, 4)	WO
Pulse width modulation mode (8-bit PWM)	Modify the pulse width as follows: PWM period: (2^8 – 1) x (m + 1)/ fj High level PWM pulse width: (m + 1)n / fj where n = high-order address set value, m = low-order address set value, fj = count source frequency	00₁₆ to FE₁₆ (High-order address) 00₁₆ to FF₁₆ (Low-order address) (3, 4)	WO

NOTES:
1. The register must be accessed in 16 bit units.
2. If the TAi register is set to 0000₁₆, the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if "pulse output" is selected, no pulses are output from the TAiOUT pin.
3. If the TAi register is set to 0000₁₆, the pulse width modulator does not work, the output level on the TAiOUT pin remains low, and timer Ai interrupt requests are not generated either. The same applies when the 8 high-order bits of the timer TAi register are set to 0000₁₆ while operating as an 8-bit pulse width modulator.
4. Use the MOV instruction to write to the TAi register.
5. The timer counts pulses from an external device or overflows or underflows in other timers.

Count Start Flag



Symbol	Address	After Reset
TABSR	0380₁₆	00₁₆

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

Up/Down Flag (1)

Symbol	Address	After Reset
UDF	0384₁₆	00₁₆

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

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

Figure 12.5 TA0 to TA4 Registers, TABSR Register, and UDF Register

One-shot Start Flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
ONSF	038216	0016

Bit Symbol	Bit Name	Function	RW
TA0OS	Timer A0 one-shot start flag	The timer starts counting by setting this bit to 1 while bits TMOD1 and TMOD0 in the TAiMR register (i = 0 to 4) = 102 (= one-shot timer mode) and the MR2 bit in the TAiMR register = 0 (=TAiOS bit enabled). When read, its content is 0	RW
TA1OS	Timer A1 one-shot start flag		RW
TA2OS	Timer A2 one-shot start flag		RW
TA3OS	Timer A3 one-shot start flag		RW
TA4OS	Timer A4 one-shot start flag		RW
TAZIE	Z-phase input enable bit	0: Z-phase input disabled 1: Z-phase input enabled	RW
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0: Input on TA0IN is selected (1) 0 1: TB2 overflow is selected (2) 1 0: TA4 overflow is selected (2) 1 1: TA1 overflow is selected (2)	RW
TA0TGH			RW

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

Trigger Select Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TRGSR	038316	0016

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

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

Clock Prescaler Reset Flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
CPSRF	038116	0XXXXXXX2

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

Figure 12.6 ONSF Register, TRGSR Register, and CPSRF Register

12.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see **Table 12.1**). **Figure 12.7** shows TAiMR register in timer mode.

Table 12.1 Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Decrement • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TAi register (i= 0 to 4) 0000_{16} to $FFFF_{16}$
Count start condition	Set TAiS bit in the TABSR register to 1 (start counting)
Count stop condition	Set TAiS bit to 0 (stop counting)
Interrupt request generation timing	Timer underflow
TAiIN pin function	I/O port or gate input
TAiOUT pin function	I/O port or pulse output
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Gate function Counting can be started and stopped by an input signal to TAiIN pin • Pulse output function Whenever the timer underflows, the output polarity of TAiOUT pin is inverted. When not counting, the pin outputs a low.

Timer Ai Mode Register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol	Address	After Reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0: Timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0: Pulse is not output (TAiOUT pin is a normal port pin) 1: Pulse is output (TAiOUT pin is a pulse output pin)	RW
MR1	Gate function select bit	b4 b3 0 0: } Gate function not available (TAiIN pin functions as I/O port) 0 1: } 1 0: Counts while input on the TAi IN pin is low (1) 1 1: Counts while input on the TAi IN pin is high (1)	RW
MR2			RW
MR3	Set to 0 in timer mode		RW
TCK0	Count source select bit	b7 b6 0 0: f_1 or f_2 0 1: f_8 1 0: f_{32} 1 1: f_{C32}	RW
TCK1			RW

NOTE:
1. The port direction bit for the TAi IN pin must be set to 0 (input mode).

Figure 12.7 Timer Ai Mode Register in Timer Mode

12.1.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers. Timers A2, A3, and A4 can count two-phase external signals. **Table 12.2** lists specifications in event counter mode (when not processing two-phase pulse signal). **Table 12.3** lists specifications in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4). **Figure 12.8** shows TAiMR register in event counter mode (when not processing two-phase pulse signal). **Figure 12.9** shows TA2MR to TA4MR registers in event counter mode (when processing two-phase pulse signal with the timers A2, A3 and A4).

Table 12.2 Specifications in Event Counter Mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAiIN pin (i=0 to 4) (effective edge can be selected in program) • Timer B2 overflows or underflows, timer Aj (j=i-1, except j=4 if i=0) overflows or underflows, timer Ak (k=i+1, except k=0 if i=4) overflows or underflows
Count operation	• Increment or decrement can be selected by external signal or program • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divided ratio	1/ (FFFF16 - n + 1) for increment 1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16
Count start condition	Set TAiS bit in the TABSR register to 1 (start counting)
Count stop condition	Set TAiS bit to 0 (stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	I/O port or count source input
TAiOUT pin function	I/O port, pulse output, or up/down-count select input
Read from timer	Count value can be read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Whenever the timer underflows or underflows, the output polarity of TAiOUT pin is inverted . When not counting, the pin outputs a low.

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

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

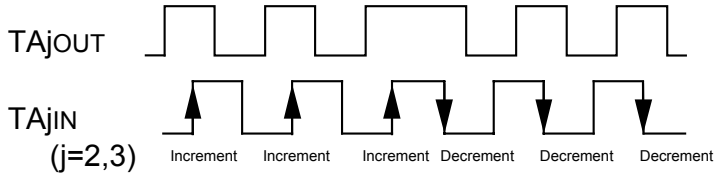
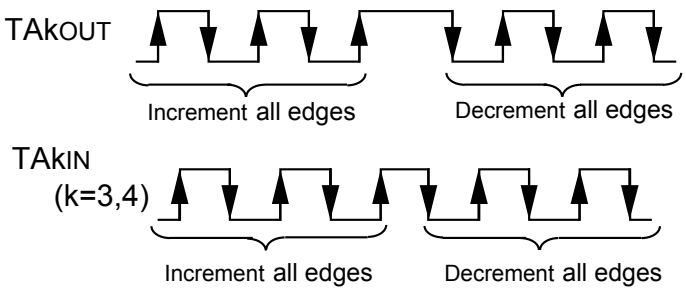
Symbol	Address	After Reset
TA0MR to TA4MR	039616 to 039A16	0016

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

NOTES:
1. During event counter mode, the count source can be selected using registers ONSF and TRGSR.
2. Effective when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are 002 (TAiIN pin input).
3. Decrement when input on TAiOUT pin is low or increment when input on that pin is high. The port direction bit for TAiOUT pin must be set to 0 (input mode).

Figure 12.8 TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing)

Table 12.3 Specifications in Event Counter Mode (when processing two-phase pulse signal with timers A2, A3 and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN or TAiOUT pins (i = 2 to 4)
Count operation	• Increment or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide ratio	1/ (FFFF16 - n + 1) for increment 1/ (n + 1) for down-count n : set value of TAi register 000016 to FFFF16
Count start condition	Set TAiS bit in the TABSR register to 1 (start counting)
Count stop condition	Set TAiS bit to 0 (stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read by reading timer A2, A3 or A4 register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to reload register (Transferred to counter when reloaded next)
Select function (Note)	• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAjIN pin when input signals on TAjOUT pin is “H”.  • Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAkIN(k=3, 4) pin goes “H” when the input signal on TAkOUT pin is “H”, the timer counts up rising and falling edges on TAkOUT and TAkIN pins. If the phase relationship is such that TAkIN pin goes “L” when the input signal on TAkOUT pin is “H”, the timer counts down rising and falling edges on TAkOUT and TAkIN pins.  • Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.

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

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

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

Symbol	Address	After Reset
TA2MR to TA4MR	0398_{16} to $039A_{16}$	00_{16}

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

NOTES:
1. The TCK1 bit is valid for timer A3 mode register. No matter how this bit is set, timers A2 and A4 always operate in normal processing mode and x4 processing mode, respectively.
2. If two-phase pulse signal processing is desired, following register settings are required:
   - Set the TAiP bit in the UDF register to 1 (two-phase pulse signal processing function enabled).
   - Set bits TAiTGH and TAiTGL in the TRGSR register to 00_2 (TAiIN pin input).
   - Set the port direction bits for TAiIN and TAiOUT to 0 (input mode).

Figure 12.9 TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)

12.1.2.1 Counter Initialization by Two-Phase Pulse Signal Processing

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

This function can only be used in timer A3 event counter mode during two-phase pulse signal processing, free-running type, x4 processing, with Z-phase entered from the $\overline{INT2}$ pin.

Counter initialization by Z-phase input is enabled by writing 0000_{16} to the TA3 register and setting the TAZIE bit in ONSF register to 1 (Z-phase input enabled).

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

The counter is initialized at the next count timing after recognizing Z-phase input. **Figure 12.9** shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

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

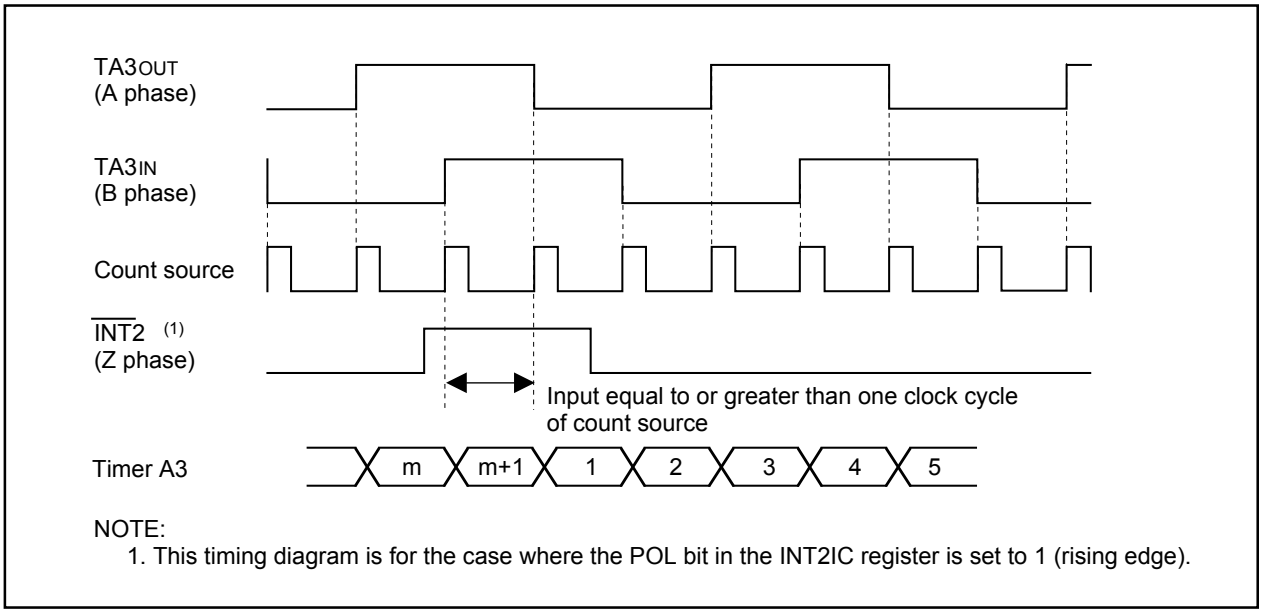


Figure 12.10 Two-phase Pulse (A phase and B phase) and the Z Phase

12.1.3 One-shot Timer Mode

In one-shot timer mode, the timer is activated only once by one trigger. (See **Table 12.4**) When the trigger occurs, the timer starts up and continues operating for a given period. **Figure 12.11** shows the TAiMR register in one-shot timer mode.

Table 12.4 Specifications in One-shot Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Decrement • When the counter reaches 0000_{16}, it stops counting after reloading a new value • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : set value of TAi register 0000_{16} to $FFFF_{16}$ However, the counter does not work if the divide-by-n value is set to 0000_{16}.
Count start condition	TAiS bit in the TABSR register is set to 1 (start counting) and one of the following triggers occurs. • External trigger input from the TAiIN pin • Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1, except k=0 if i=4) overflow or underflow • The TAiOS bit in the ONSF register is set to 1 (timer starts)
Count stop condition	• When the counter is reloaded after reaching 0000_{16} • TAiS bit is set to “0” (stop counting)
Interrupt request generation timing	When the counter reaches 0000_{16}
TAiIN pin function	I/O port or trigger input
TAiOUT pin function	I/O port or pulse output
Read from timer	An undefined value is read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)
Select function	• Pulse output function The timer outputs a low when not counting and a high when counting.

Timer Ai Mode Register (i=0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol	Address	After Reset
TA0MR to TA4MR	396_{16} to $039A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0: One-shot timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0: Pulse is not output (TAiOUT pin functions as I/O port) 1: Pulse is output (TAiOUT pin functions as a pulse output pin)	RW
MR1	External trigger select bit (1)	0: Falling edge of input signal to TAiIN pin (2) 1: Rising edge of input signal to TAiIN pin (2)	RW
MR2	Trigger select bit	0: TAiOS bit is enabled 1: Selected by bits TAiTGH to TAiTGL	RW
MR3	Set to 0 in one-shot timer mode		RW
TCK0	Count source select bit	b7 b6 0 0: f_1 or f_2 0 1: f_8 1 0: f_{32} 1 1: fC_{32}	RW
TCK1			RW

NOTES:
1. Effective when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are 00_2 (TAiIN pin input).
2. The port direction bit for the TAiIN pin must be set to 0 (input mode).

Figure 12.11 TAiMR Register in One-shot Timer Mode

12.1.4. Pulse Width Modulation (PWM) Mode

In PWM mode, the timer outputs pulses of a given width in succession (see **Table 12.5**). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. **Figure 12.12** shows TAiMR register in pulse width modulation mode. **Figures 12.13** and **12.14** show examples of how a 16-bit pulse width modulator operates and how an 8-bit pulse width modulator operates.

Table 12.5 Specifications in Pulse Width Modulation Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Decrement (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new value at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	• High level width n / fj n : set value of TAi register (i=o to 4) • Cycle time $(2^{16}-1)$ / fj fixed fj: count source frequency (f_1, f_2, f_8, f_{32}, f_{C32})
8-bit PWM	• High level width n x (m+1) / fj n : set value of TAi register high-order address • Cycle time (2^8-1) x (m+1) / fj m : set value of TAi register low-order address
Count start condition	• TAiS bit in the TABSR register is set to “1” (= start counting) • The TAiS bit = 1 and external trigger input from the TAiIN pin • The TAiS bit = 1 and one of the following external triggers occurs • Timer B2 overflow or underflow, timer Aj (j=i-1, except j=4 if i=0) overflow or underflow, timer Ak (k=i+1, except k=0 if i=4) overflow or underflow
Count stop condition	TAiS bit is set to 0 (stop counting)
Interrupt request generation timing	PWM pulse goes “L”
TAiIN pin function	I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	An undefined value is read by reading TAi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to only reload register (Transferred to counter when reloaded next)

Timer Ai Mode Register (i= 0 to 4)

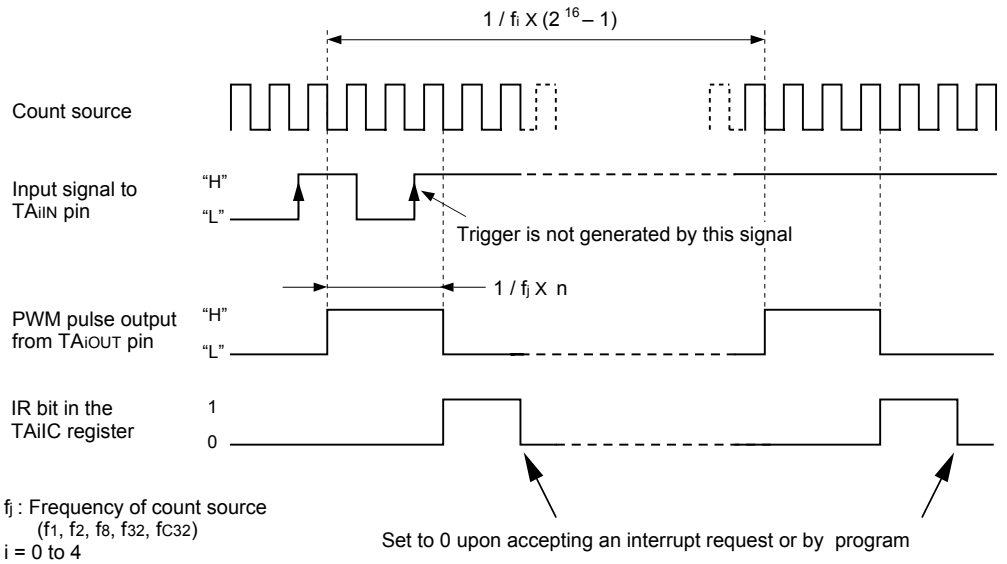
b7	b6	b5	b4	b3	b2	b1	b0
						1	1

Symbol	Address	After Reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 1: PWM mode	RW
TMOD1			RW
MR0	Pulse output funcion select bit	0: Pulse is not output (TAiOUT pin functions as I/O port) 1: Pulse is output (TAiOUT pin functions as a pulse output pin)	RW
MR1	External trigger select bit (1)	0: Falling edge of input signal to TAiIN pin(2) 1: Rising edge of input signal to TAiIN pin(2)	RW
MR2	Trigger select bit	0: Write 1 to TAiS bit in the TASF register 1: Selected by bits TAiTGH to TAiTGL	RW
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	RW
TCK0	Count source select bit	b7 b6 0 0: f_1 or f_2 0 1: f_8 1 0: f_{32} 1 1: fc_{32}	RW
TCK1			RW

NOTES:
1. Effective when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are 00_2 (TAiIN pin input).
2. The port direction bit for the TAiIN pin must be set to 0 (input mode).

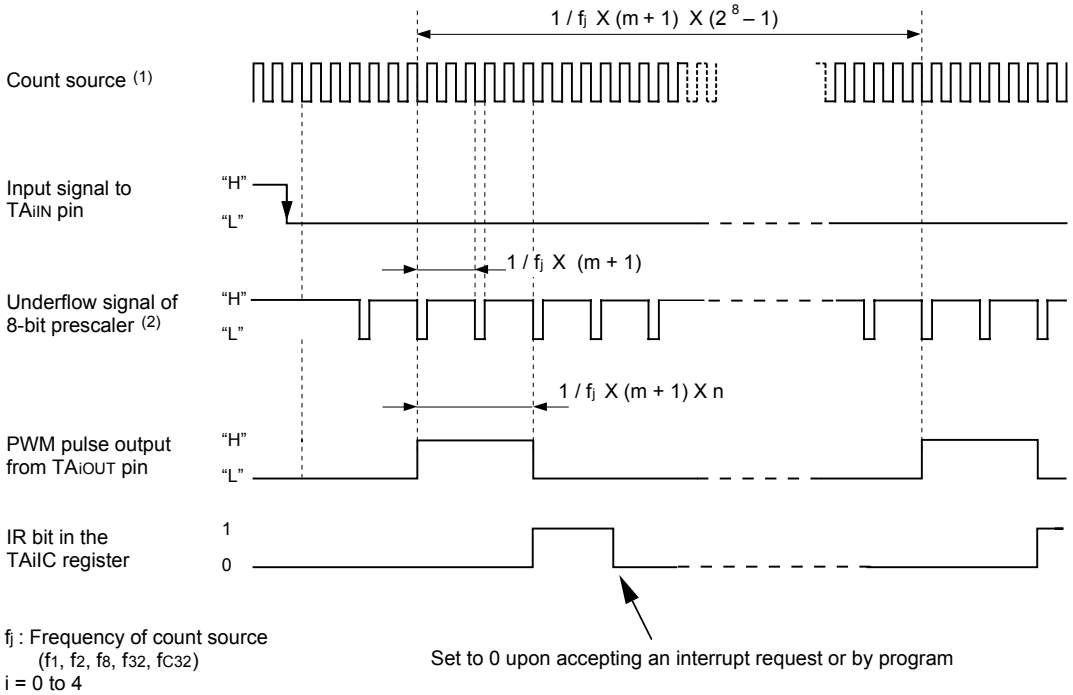
Figure 12.12 TAiMR Register in Pulse Width Modulation Mode



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

NOTES:
1. n = 0000 16 to FFFE 16.
2. This timing diagram is for the case where the TAi register is 0003 16, bits TAiTGH and TAiTGL in the ONSF or TRGSR register is set to 00 2 (TAiIN pin input), the MR1 bit in the TAiMR register is set to 1 (rising edge), and the MR2 bit in the TAiMR register is set to 1 (trigger selected by TAiTGH and TAiTGL bits).

Figure 12.13 Example of 16-bit Pulse Width Modulator Operation



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

NOTES:
1. The 8-bit prescaler counts the count source.
2. The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
3. m = 00 16 to FF 16; n = 00 16 to FE 16.
4. This timing diagram is for the case where the TAi register is 0202 16, bits TAiTGH and TAiTGL in the ONSF or TRGSR register is set to 00 2 (TAiIN pin input), the MR1 bit in the TAiMR register is set to 0 (falling edge), and the MR2 bit in the TAiMR register is set to 1 (trigger selected by bits TAiTGH and TAiTGL).

Figure 12.14 Example of 8-bit Pulse Width Modulator Operation

12.2 Timer B

Figure 12.15 shows a block diagram of the timer B. **Figures 12.16** and **12.17** show registers related to the timer B.

Timer B supports the following four modes. Use bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 2) to select the desired mode.

- Timer mode: The timer counts the internal count source.
- Event counter mode: The timer counts the external pulses or overflows and underflows of other timers.
- Pulse period/pulse width measurement mode: The timer measures the pulse period or pulse width of external signal.
- A/D trigger mode: The timer starts counting by one trigger until the count value becomes 0000_{16}. This mode is used together with simultaneous sample sweep mode or delayed trigger mode 0 of A/D converter to start A/D conversion.

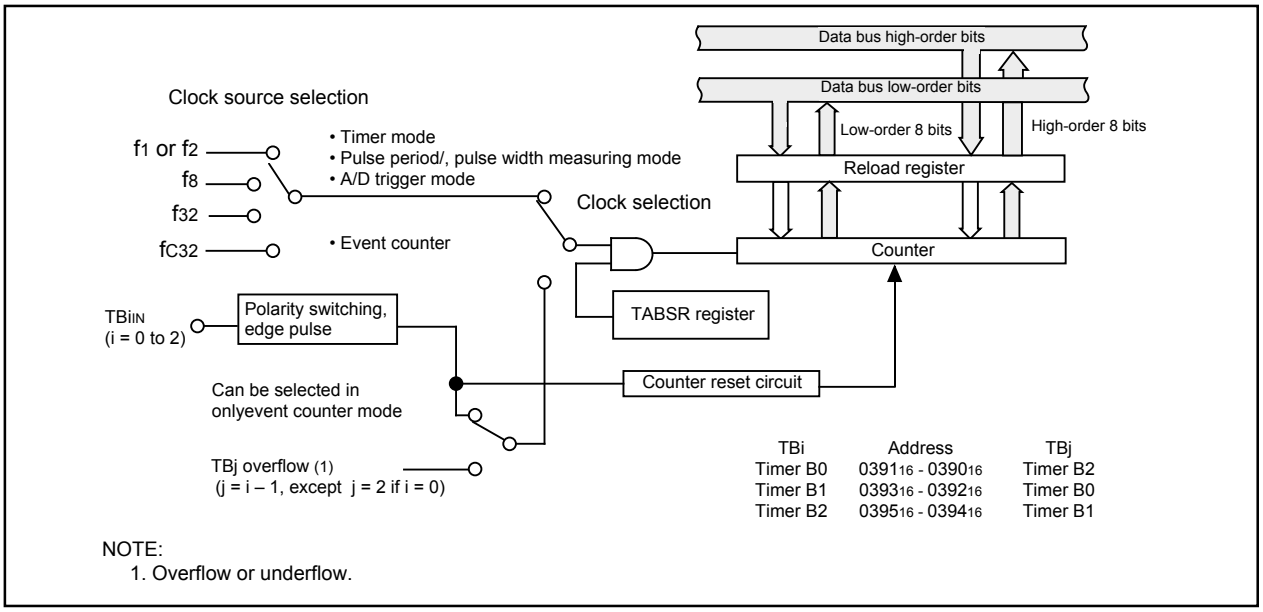


Figure 12.15 Timer B Block Diagram

Timer Bi Mode Register (i=0 to 2)

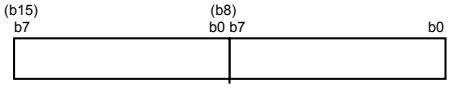
Symbol	Address	After Reset
TB0MR to TB2MR	039B16 to 039D16	00XX0000₂

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode or A/D trigger mode 0 1 : Event counter mode 1 0 : Pulse period measurement mode, pulse width measurement mode 1 1 : Do not set	RW
TMOD1			RW
MR0	———	Function varies with each operation mode	RW
MR1			RW
MR2			RW(1) —— (2)
MR3			RO
TCK0	Count source select bit	Function varies with each operation mode	RW
TCK1			RW

NOTES:
1. Timer B0.
2. Timer B1, Timer B2.

Figure 12.16 TB0MR to TB2MR Registers

Timer Bi Register (i=0 to 2)(1)

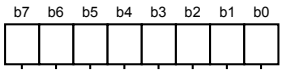


Symbol	Address	After Reset
TB0	039116, 039016	Undefined
TB1	039316, 039216	Undefined
TB2	039516, 039416	Undefined

Mode	Function	Setting Rrange	RW
Timer mode	Divide the count source by n + 1 where n = set value	000016 to FFFF16	RW
Event counter mode	Divide the count source by n + 1 where n = set value (2)	000016 to FFFF16	RW
Pulse period modulation mode, Pulse width modulation mode	Measures a pulse period or width	——	RO
A/D trigger mode (3)	Divide the count source by n + 1 where n = set value and cause the timer stop	000016 to FFFF16	RW

NOTES:
1. The register must be accessed in 16 bit units
2. The timer counts pulses from an external device or overflows or underflows of other timers.
3. When this mode is used combining delayed trigger mode 0, set the larger value than the value in the timer B0 register to the timer B1 register.

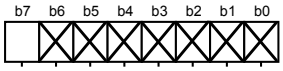
Count Start Flag



Symbol	Address	After Reset
TABSR	038016	0016

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

Clock Prescaler Reset flag



Symbol	Address	After Reset
CPSRF	038116	0XXXXXXX2

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

Figure 12.17 TB0 to TB2 Registers, TABSR Register, CPSRF Register

12.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see **Table 12.6**). **Figure 12.18** shows TBiMR register in timer mode.

Table 12.6 Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Decrement • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TBi register (i= 0 to 2) 0000_{16} to $FFFF_{16}$
Count start condition	Set TBiS bit(1) to 1 (start counting)
Count stop condition	Set TBiS bit to 0 (stop counting)
Interrupt request generation timing	Timer underflow
TBi_{IN} pin function	I/O port
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

NOTE:
1. Bits TB0S to TB2S are assigned to the bit 7 to bit 5 in the TABSR register.

Timer Bi Mode Register (i= 0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol	Address	After Reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0: Timer mode or A/D trigger mode	RW
TMOD1			RW
MR0	No effect in timer mode Can be set to 0 or 1		RW
MR1			RW
MR2	TB0MR register Set to 0 in timer mode		RW
	TB1MR, TB2MR registers Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
MR3	When write in timer mode, set to 0. When read in timer mode, the content is undefined		RO
TCK0	Count source select bit	b7 b6 0 0: f_1 or f_2 0 1: f_8 1 0: f_{32} 1 1: f_{C32}	RW
TCK1			RW

Figure 12.18 TBiMR Register in Timer Mode

12.2.2 Event Counter Mode

In event counter mode, the timer counts pulses from an external device or overflows and underflows of other timers (see **Table 12.7**). **Figure 12.19** shows the TBiMR register in event counter mode.

Table 12.7 Specifications in Event Counter Mode

Item	Specification
Count source	• External signals input to TBiIN pin (i=0 to 2) (effective edge can be selected in program) • Timer Bj overflow or underflow (j=i-1, except j=2 if i=0)
Count operation	• Decrement • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TBi register 0000$_{16}$ to FFFF$_{16}$
Count start condition	Set TBiS bit(1) to 1 (start counting)
Count stop condition	Set TBiS bit to 0 (stop counting)
Interrupt request generation timing	Timer underflow
TBiIN pin function	Count source input
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

NOTE:
1. Bits TB2S to TB0S are assigned to the bit 7 to bit 5 in the TABSR register.

Timer Bi Mode Register (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol	Address	After Reset
TB0MR to TB2MR	039B$_{16}$ to 039D$_{16}$	00XX0000$_2$

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1: Event counter mode	RW
TMOD1			RW
MR0	Count polarity select bit (1)	b3 b2 0 0: Counts external signal's falling edges 0 1: Counts external signal's rising edges 1 0: Counts external signal's falling and rising edges 1 1: Do not set	RW
MR1			RW
MR2	TB0MR register Set to 0 in timer mode		RW
	TB1MR, TB2MR registers Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
MR3	When write in event counter mode, set to 0. When read in event counter mode, the content is undefined		RO
TCK0	No effect in event counter mode Can be set to 0 or 1		RW
TCK1	Event clock select	0 : Input from TBiIN pin (2) 1 : TBj overflow or underflow (j = i – 1, except j = 2 if i = 0)	RW

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

Figure 12.19 TBiMR Register in Event Counter Mode

12.2.3 Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal (see **Table 12.8**). **Figure 12.20** shows the TBiMR register in pulse period and pulse width measurement mode. **Figure 12.21** shows the operation timing when measuring a pulse period. **Figure 12.22** shows the operation timing when measuring a pulse width.

Table 12.8 Specifications in Pulse Period and Pulse Width Measurement Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Increment • Counter value is transferred to reload register at an effective edge of measurement pulse. The counter value is set to 0000_{16} to continue counting.
Count start condition	Set TBiS (i=0 to 2) bit (3) to 1 (start counting)
Count stop condition	Set TBiS bit to 0 (stop counting)
Interrupt request generation timing	• When an effective edge of measurement pulse is input (1) • Timer overflow. When an overflow occurs, MR3 bit in the TBiMR register is set to 1 (overflowed) simultaneously. MR3 bit is cleared to 0 (no overflow) by writing to TBiMR register at the next count timing or later after MR3 bit was set to 1. At this time, make sure TBiS bit is set to 1 (start counting).
TBi_{IN} pin function	Measurement pulse input
Read from timer	Contents of the reload register (measurement result) can be read by reading TBi register (2)
Write to timer	Value written to TBi register is written to neither reload register nor counter

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

Timer Bi Mode Register (i=0 to 2)

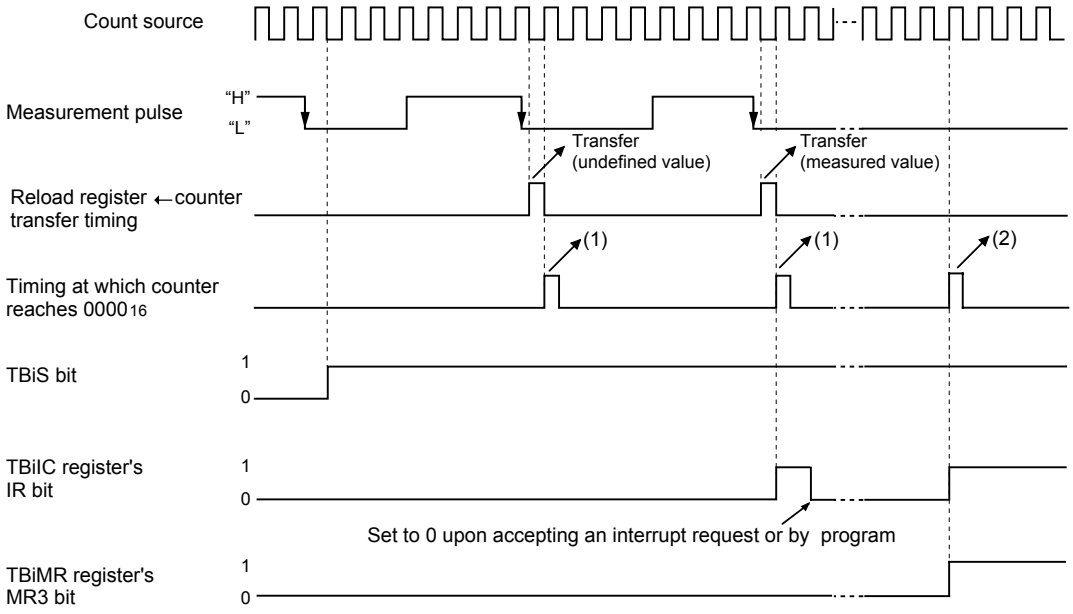
b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Symbol	Address	After Reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement mode select bit	b3 b2 0 0: Pulse period measurement (Measurement between a falling edge and the next falling edge of measured pulse) 0 1: Pulse period measurement (Measurement between a rising edge and the next rising edge of measured pulse) 1 0: Pulse width measurement (Measurement between a falling edge and the next rising edge of measured pulse and between a rising edge and the next falling edge) 1 1: Do not set.	RW
MR1			RW
MR2	TB0MR register Set to 0 in pulse period and pulse width measurement mode		RW
	TB1MR, TB2MR registers Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
MR3	Timer Bi overflow flag (1)	0 : Timer did not overflow 1 : Timer has overflowed	RO
TCK0	Count source select bit	b7 b6 0 0: f_1 or f_2 0 1: f_8 1 0: f_{32} 1 1: f_{C32}	RW
TCK1			RW

NOTE:
1. This flag is undefined after reset. When the TBiS bit is set to 1 (start counting), the MR3 bit is cleared to 0 (no overflow) by writing to the TBiMR register at the next count timing or later after the MR3 bit was set to 1 (overflowed). The MR3 bit cannot be set to 1 by program. Bits TB0S to TB2S are assigned to the bit 5 to bit 7 in the TABSR register.

Figure 12.20 TBiMR Register in Pulse Period and Pulse Width Measurement Mode

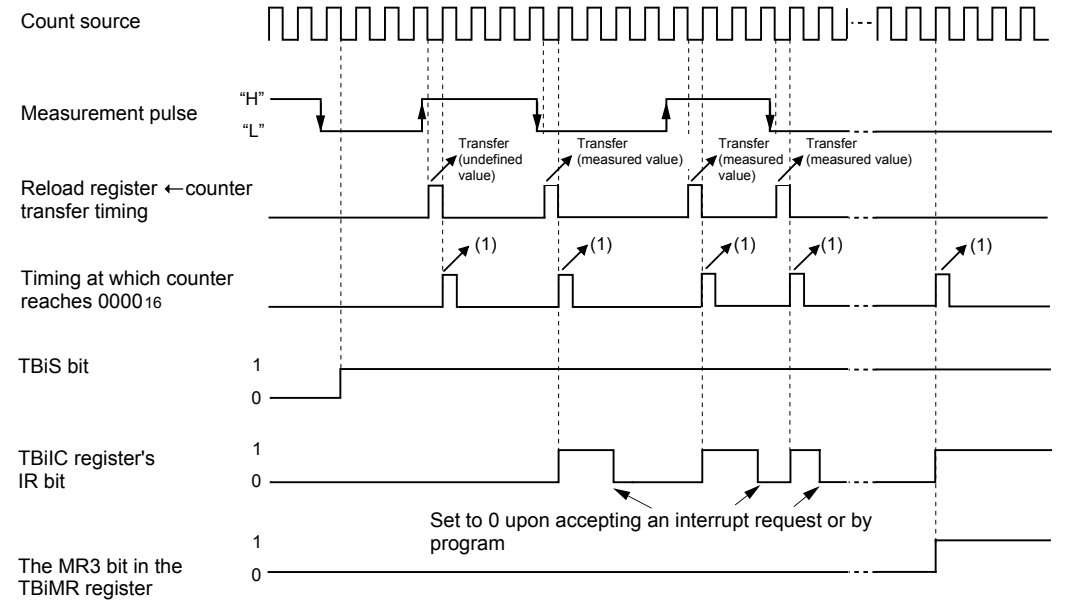


Bits TB0S to TB2S are assigned to the bit 5 to bit 7 in the TABSR register.

i = 0 to 2

NOTES:
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where bits MR1 and MR0 in the TBiMR register are 00_2 (measure the interval from falling edge to falling edge of the measurement pulse).

Figure 12.21 Operation timing when measuring a pulse period



Bits TB2S to TB0S are assigned to the bit 7 to bit 5 in the TABSR register.

i = 0 to 2

NOTES:
1. Counter is initialized at completion of measurement.
2. Timer has overflowed.
3. This timing diagram is for the case where bits MR1 to MR0 in the TBiMR register are 10_2 (measure the interval from a falling edge to the next rising edge and the interval from a rising edge to the next falling edge of the measurement pulse).

Figure 12.22 Operation timing when measuring a pulse width

12.2.4 A/D Trigger Mode

A/D trigger mode is used together with simultaneous sample sweep mode or delayed trigger mode 0 of A/D conversion to start A/D conversion. It is used in timer B0 and timer B1 only. In this mode, the timer starts counting by one trigger until the count value becomes 0000_{16}. **Figure 12.23** shows the TBiMR register in A/D trigger mode and **Figure 12.24** shows the TB2SC register.

Table 12.9 Specifications in A/D Trigger Mode

Item	Specification
Count Source	f_1, f_2, f_8, f_{32}, and f_{C32}
Count Operation	• Decrement • When the timer underflows, reload register contents are reloaded before stopping counting • When a trigger is generated during the count operation, the count is not affected
Divide Ratio	1/(n+1) n: Setting value of TBi register (i=0,1) 0000_{16}-$FFFF_{16}$
Count Start Condition	When the TBiS (i=0,1) bit in the TABSR register is 1(count started), TBiEN(i=0,1) in TB2SC register is 1 (A/D trigger mode) and the following trigger is generated.(Selection based on bits TB2SEL in the TB2SC) • Timer B2 interrupt • Underflow of Timer B2 interrupt generation frequency counter setting
Count Stop Condition	• After the count value is 0000_{16} and reload register contents are reloaded • Set the TBiS bit to 0 (count stopped)
Interrupt Request Generation Timing	Timer underflows (1)
TBiIN Pin Function	I/O port
Read From Timer	Count value can be read by reading TBi register
Write To Timer (2)	• When writing in the TBi register during count stopped. Value is written to both reload register and counter • When writing in the TBi register during count. Value is written to only reload register (Transfered to counter when reloaded next)

NOTES:

1: A/D conversion is started by the timer underflow. For details refer to **15. A/D Converter**.

2: When using in delayed trigger mode 0, set the larger value than the value of the timer B0 register to the timer B1 register.

Timer Bi Mode Register (i= 0 to 1)

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol: TB0MR to TB1MR
Address: 039B16 to 039C16
After Reset: 00XX00002

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0: Timer mode or A/D trigger mode	RW
TMOD1			RW
MR0	Invalid in A/D trigger mode Either 0 or 1 is enabled		RW
MR1			RW
MR2	TB0MR register Set to 0 in A/D trigger mode		RW
	TB1MR register Nothing is assigned. If necessary, set to 0. When read, its content is undefined		—
MR3	When write in A/D trigger mode, set to 0. When read in A/D trigger mode, the content is undefined		RO
TCK0	Count source select bit (1)	b7 b6 0 0: f1 or f2 0 1: f8 1 0: f32 1 1: fC32	RW
TCK1			RW

NOTE:
1. When this bit is used in delayed trigger mode 0, set the same count source to the timer B0 and timer B1.

Figure 12.23 TBiMR Register in A/D Trigger Mode

Timer B2 special mode register (1)

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

Symbol: TB2SC
Address: 039E16
After Reset: X00000002

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 reload timing switch bit (2)	0: Timer B2 underflow 1: Timer A output at odd-numbered	RW
IVPCR1	Three-phase output port $\overline{SD}$ control bit 1 (3, 4, 7)	0: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled 1: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled	RW
TB0EN	Timer B0 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode (5)	RW
TB1EN	Timer B1 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode (5)	RW
TB2SEL	Trigger select bit (6)	0: TB2 interrupt 1: Underflow of TB2 interrupt generation frequency setting counter [ICTB2]	RW
$\overline{\text{(b6-b5)}}$	Reserved bits	Set to 0	RW
— (b7)	Nothing is assigned. If necessary, set to 0. When read, its content is 0		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. If the INV11 bit is 0 (three-phase mode 0) or the INV06 bit is 1 (triangular wave modulation mode), set this bit to 0 (timer B2 underflow).
3. When setting the IVPCR1 bit to 1 (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), set the PD8_5 bit to 0 (= input mode).
4. Related pins are U(P80), $\overline{U}$(P81), V(P72), $\overline{V}$(P73), W(P74), $\overline{W}$(P75). After forcible cutoff, input "H" to the P85/$\overline{NMI}$/$\overline{SD}$ pin. Set the IVPCR1 bit to 0, and this forcible cutoff will be reset. If "L" is input to the P85/$\overline{NMI}$/$\overline{SD}$ pin, a three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is 0, the target pins changes to programmable I/O port. When the IVPCR1 bit is 1, the target pins changes to high-impedance state regardless of which functions of those pins are used.
5. When this bit is used in delayed trigger mode 0, set bits TB0EN and TB1EN to 1 (A/D trigger mode).
6. When setting the TB2SEL bit to 1 (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), set the INV02 bit to 1 (three-phase motor control timer function).
7. Refer to "**18.6 Digital Debounce function**" for the $\overline{SD}$ input.

Figure 12.24 TB2SC Register in A/D Trigger Mode

12.3 Three-phase Motor Control Timer Function

Timers A1, A2, A4 and B2 can be used to output three-phase motor drive waveforms. **Table 12.10** lists the specifications of the three-phase motor control timer function. **Figure 12.25** shows the block diagram for three-phase motor control timer function. Also, the related registers are shown on **Figures 12.26** to **12.32.**

Table 12.10 Three-phase Motor Control Timer Function Specifications

Item	Specification
Three-phase waveform output pin	Six pins (U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$)
Forced cutoff input (1)	Input "L" to $\overline{SD}$ pin
Used Timers	Timer A4, A1, A2 (used in the one-shot timer mode) Timer A4: U- and $\overline{U}$-phase waveform control Timer A1: V- and $\overline{V}$-phase waveform control Timer A2: W- and $\overline{W}$-phase waveform control Timer B2 (used in the timer mode) Carrier wave cycle control Dead time timer (3 eight-bit timer and shared reload register) Dead time control
Output waveform	Triangular wave modulation, Sawtooth wave modification Enable to output "H" or "L" for one cycle Enable to set positive-phase level and negative-phase level respectively
Carrier wave cycle	Triangular wave modulation: count source x (m+1) x 2 Sawtooth wave modulation: count source x (m+1) m: Setting value of TB2 register, 0 to 65535 Count source: f_1, f_2, f_8, f_{32}, f_{C32}
Three-phase PWM output width	Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: Setting value of TA4, TA1 and TA2 register (of TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11 bit to 1), 1 to 65535 Count source: f_1, f_2, f_8, f_{32}, f_{C32}
Dead time	Count source x p, or no dead time p: Setting value of DTT register, 1 to 255 Count source: f_1, f_2, f_1 divided by 2, f_2 divided by 2
Active level	Eable to select "H" or "L"
Positive and negative-phase concurrent	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function
Interrupt frequency	For Timer B2 interrupt, select a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis

NOTE:

1. When the INV02 bit in the INVC0 register is set to 1 (three-phase motor control timer function), the $\overline{SD}$ function of the $P8_5/\overline{SD}$ pin is enabled. At this time, the $P8_5$ pin cannot be used as a programmable I/O port. When the $\overline{SD}$ function is not used, apply "H" to the $P8_5/\overline{SD}$ pin.
   When the IVPCR1 bit in the TB2SC register is set to 1 (enable three-phase output forced cutoff by $\overline{SD}$ pin input), and "L" is applied to the $\overline{SD}$ pin, the related pins enter high-impedance state regardless of the functions which are used. When the IVPCR1 bit is set to 0 (disabled three-phase output forced cutoff by $\overline{SD}$ pin input) and "L" is applied to the $\overline{SD}$ pin, the related pins can be selected as a programmable I/O port and the setting of the port and port direction registers are enable.
   Related pins: $P7_2$/CLK2/TA1OUT/V/RXD1 $P7_3/\overline{CTS_2}/\overline{RTS_2}$/TA1IN/$\overline{V}$/TXD1
   $P7_4$/TA2OUT/W $P7_5$/TA2IN/$\overline{W}$
   $P8_0$/TA4OUT/U $P8_1$/TA4IN/$\overline{U}$

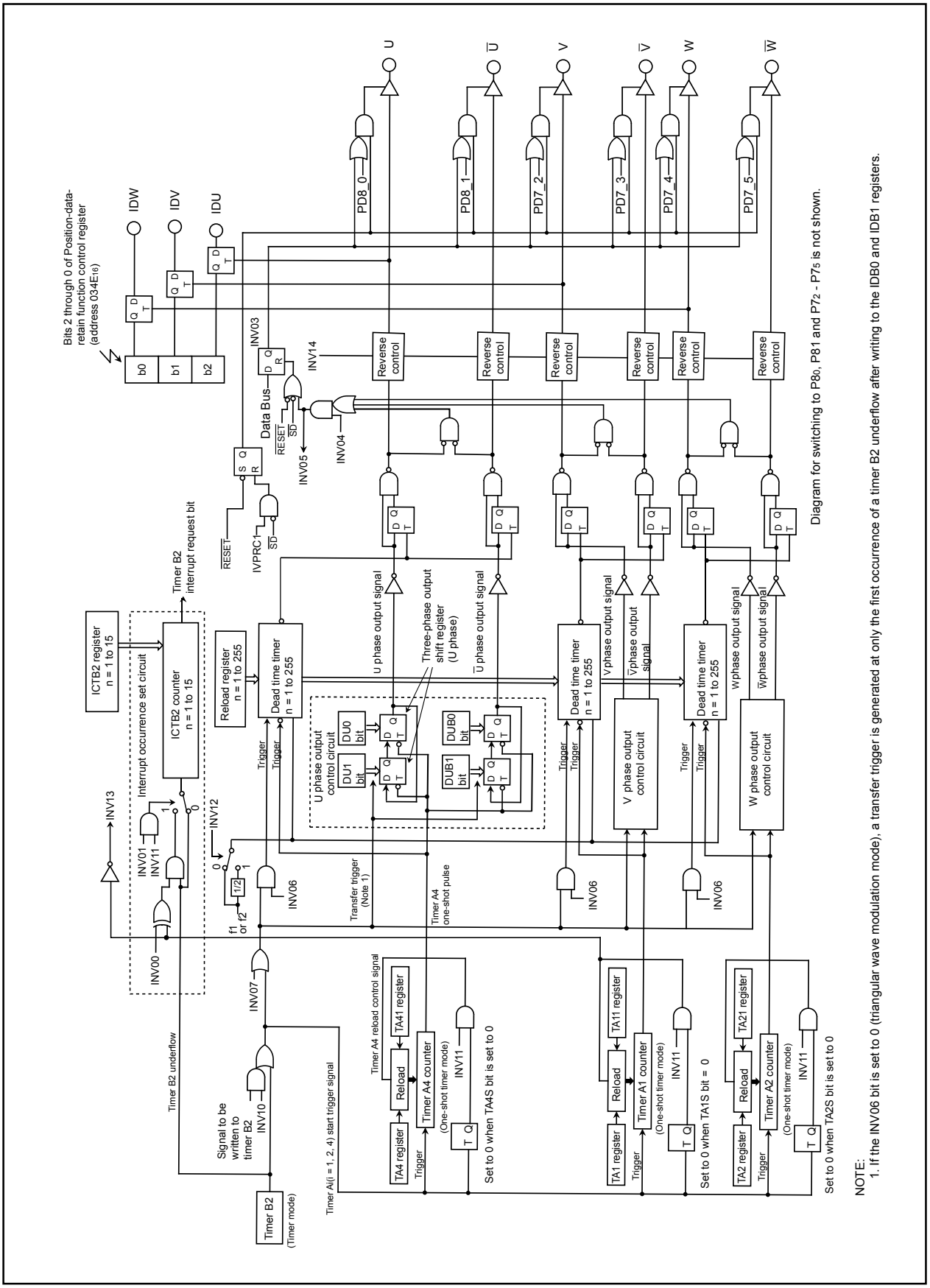


Figure 12.25 Three-phase Motor Control Timer Functions Block Diagram

Three-phase PWM Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
INVC0	034816	0016

Bit Symbol	Bit Name	Function	RW
INV00	Effective interrupt output polarity select bit(3)	0: ICTB2 counter is incremented by 1 on the rising edge of timer A1 reload control signal 1: ICTB2 counter is incremented by 1 on the falling edge of timer A1 reload control signal	RW
INV01	Effective interrupt output specification bit(2, 3)	0: ICTB2 counter incremented by 1 at a timer B2 underflow 1: Selected by INV00 bit	RW
INV02	Mode select bit(4)	0: Three-phase motor control timer function unused 1: Three-phase motor control timer function (5)	RW
INV03	Output control bit(6)	0: Three-phase motor control timer output disabled (5) 1: Three-phase motor control timer output enabled	RW
INV04	Positive and negative phases concurrent output disable bit	0: Simultaneous active output enabled 1: Simultaneous active output disabled	RW
INV05	Positive and negative phases concurrent output detect flag	0: Not detected yet 1: Already detected (7)	RW
INV06	Modulation mode select bit(8)	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode (9)	RW
INV07	Software trigger select bit	Setting this bit to 1 generates a transfer trigger. If the INV06 bit is 1, a trigger for the dead time timer is also generated. The value of this bit when read is 0	RW

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enable). Note also that bits INV00 to INV02, bits INV04 and INV06 can only be rewritten when timers A1, A2, A4 and B2 are idle.
2. If this bit needs to be set to 1, set any value in the ICTB2 register before writing to it.
3. Effective when the INV11 bit in the INV1 register is 1 (three-phase mode 1). If INV11 is set to 0 (three-phase mode 0), the ICTB2 counter is incremented by 1 each time the timer B2 underflows, regardless of whether the INV00 and INV01 bits are set. When setting the INV01 bit to 1, the first interrupt is generated when the timer B2 underflows n-1 times, if n is the value set in the ICTB2 counter. Subsequent interrupts are generated every n times the timer B2 underflow.
4. Setting the INV02 bit to 1 activates the dead time timer, U/V/W-phase output control circuits and ICTB2 counter.
5. When the INV02 bit is set to 1 and the INV03 bit is set to 0, $\overline{U}$, U, $\overline{V}$, V, $\overline{W}$, W pins, including pins shared with other output functions, enter a high-impedance state. When INV03 is set to 1, U/V/W corresponding pins generate the three-phase PWM output.
6. The INV03 bit is set to 0 in the following cases:
   - When reset
   - When positive and negative go active (INV05 = 1) simultaneously while INV04 bit is 1
   - When set to 0 by program
   - When input on the $\overline{SD}$ pin changes state from “H” to “L” regardless of the value of the INVCR1 bit. (The INV03 bit cannot be set to 1 when $\overline{SD}$ input is “L”.) INV03 is set to 0 when both bits INV05 and INV04 are set to 1.

Item	INV06=0	INV06=1
Mode	Triangular wave modulation mode	Sawtooth wave modulation mode
Timing at which transferred from registers IDB0 to IDB1 to three-phase output shift register	Transferred only once synchronously with the transfer trigger after writing to registers IDB0 to IDB1	Transferred every transfer trigger
Timing at which dead time timer trigger is generated when INV16 bit is 0	Synchronous with the falling edge of timer A1, A2, or A4 one-shot pulse	Synchronous with the transfer trigger and the falling edge of timer A1, A2, or A4 one-shot pulse
INV13 bit	Effective when INV11 is set to 1 and INV06 is set to 0	No effect

Transfer trigger: Timer B2 underflow, write to the INV07 bit or write to the TB2 register when the INV10 bit is set to 1.

9: If the INV06 bit is set to 1, set the INV11 bit to 0 (three-phase mode 0) and set the PWCON bit to 0 (timer B2 reloaded by a timer B2 underflow)

10. When the PFCi (i = 0 to 5) bit in the PFCR register is set to 1 (three-phase PWM output), individual pins are enabled to output.

Figure 12.26 INVC0 Register

Three-phase PWM Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	After Reset
INVC1	034916	0016

Bit Symbol	Bit Name	Function	RW
INV10	Timer A1, A2, A4 start trigger signal select bit	0: Timer B2 underflow 1: Timer B2 underflow and write to the TB2 register(2)	RW
INV11	Timer A1-1, A2-1, A4-1 control bit (3)	0: Three-phase mode 0 (4) 1: Three-phase mode 1	RW
INV12	Dead time timer count source select bit	0: f1 or f2 1: f1 divided by 2 or f2 divided by 2	RW
INV13	Carrier wave detect flag (5)	0: Timer Reload control signal is set to 0 1: Timer Reload control signal is set to 1	RO
INV14	Output polarity control bit	0 : Output waveform “L” active 1 : Output waveform “H” active	RW
INV15	Dead time invalid bit	0: Dead time timer enabled 1: Dead time timer disabled	RW
INV16	Dead time timer trigger select bit	0: Falling edge of timer A4, A1 or A2 one-shot pulse 1: Rising edge of three-phase output shift register (U, V or W phase) output(6)	RW
(b7)	Reserved bit	Set to 0	RW

NOTES:

1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enable). Note also that this register can only be rewritten when timers A1, A2, A4 and B2 are idle.
2. A start trigger is generated by writing to the TB2 register only while timer B2 stops.
3. The effects of the INV11 bit are described in the table below.

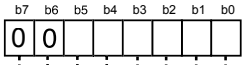
Item	INV11=0	INV11=1
Mode	Three-phase mode 0	Three-phase mode 1
TA11, TA21, TA41 registers	Not Used	Used
INV00 bit, INV01 bit	Has no effect. ICTB2 counted every time timer B2 underflows regardless of whether bits INV00 and INV01 are set	Effect
INV13 bit	Has no effect	Effective when INV11 bit is 1 and INV06 bit is 0

4. If the INV06 bit is 1 (sawtooth wave modulation mode), set this bit to 0 (three-phase mode 0). Also, if the INV11 bit is 0, set the PWCON bit to 0 (timer B2 reloaded by a timer B2 underflow).
5. The INV13 bit is effective only when the INV06 bit is set to 0 (triangular wave modulation mode) and the INV11 bit is set to 1 (three-phase mode 1).
6. If all of the following conditions hold true, set the INV16 bit to 1 (dead time timer triggered by the rising edge of three-phase output shift register output)
   - The INV15 bit is 0 (dead time timer enabled)
   - When the INV03 bit is set to 1 (three-phase motor control timer output enabled), the Dij bit and DiBj bit (i:U, V, or W, j: 0 to 1) have always different values (the positive-phase and negative-phase always output different levels during the period other than dead time).

Conversely, if either one of the above conditions holds false, set the INV16 bit to 0 (dead time timer triggered by the falling edge of one-shot pulse).

Figure 12.27 INVC1 Register

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

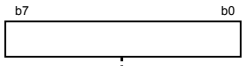


Symbol	Address	After Reset
IDB0	$034A_{16}$	00111111_2
IDB1	$034B_{16}$	00111111_2

Bit Symbol	Bit Name	Function	RW
DUi	U phase output buffer i	Write the output level 0: Active level 1: Inactive level When read, these bits show the three-phase output shift register value.	RW
DUBi	$\overline{U}$ phase output buffer i		RW
DVi	V phase output buffer i		RW
DVBi	$\overline{V}$ phase output buffer i		RW
DWi	W phase output buffer i		RW
DWBi	$\overline{W}$ phase output buffer i		RW
$\overline{(b7\text{-}b6)}$	Nothing is assigned. If necessary, set to 0. When read, these contents are 0		RO

NOTE:
1. Registers IDB0 and IDB1 values are transferred to the three-phase shift register by a transfer trigger. The value written to the IDB0 register aftera transfer trigger represents the output signal of each phase, and the next value written to the IDB1 register at the falling edge of the timer A1, A2, or A4 one-shot pulse represents the output signal of each phase.

Dead Time Timer (1, 2)

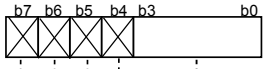


Symbol	Address	After Reset
DTT	$034C_{16}$	Undefined

Function	Setting Range	RW
Assuming the set value = n, upon a start trigger the timer starts counting the count souce selected by the INV12 bit and stops after counting it n times. The positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops.	1 to 255	WO

NOTES:
1. Use MOV instruction to write to this register.
2. Effective when the INV15 bit is set to 0 (dead time timer enable). If the INV15 bit is set to 1, the dead time timer is disabled and has no effect.

Timer B2 Interrupt Occurrences Frequency Set Counter



Symbol	Address	After Reset
ICTB2	$034D_{16}$	Undefined

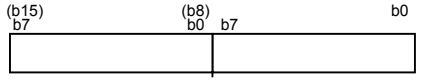
Function	Setting Range	RW
If the INV01 bit is 0 (ICTB2 counter counted every time timer B2 underflows), assuming the set value = n, a timer B2 interrupt is generated at every *n*th occurrence of a timer B2 underflow. If the INV01 bit is 1 (ICTB2 counter count timing selected by the INV00 bit), assuming the set value = n, a timer B2 interrupt is generated at every *n*th occurrence of a timer B2 underflow that meets the condition selected by the INV00 bit. (1)	1 to 15	WO
Nothing is assigned. When write, set to "0". When read, the content is undefined.		—

NOTE:
1. Use MOV instruction to write to this register.
If the INV01 bit is set to 1, make sure the TB2S bit also is set to 0 (timer B2 count stopped) when writing to this register. If the INV01 bit is set to 0, although this register can be written even when the TB2S bit is set to 1 (timer B2 count start), do not write synchronously with a timer B2 underflow.

Figure 12.28 IDB0 Register, IDB1Register, DTT Register, and ICTB2 Register

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

Symbol	Address	After reset
TA1	038916-038816	Undefined
TA2	038B16-038A16	Undefined
TA4	038F16-038E16	Undefined
TA11(6,7)	034316-034216	Undefined
TA21(6,7)	034516-034416	Undefined
TA41(6,7)	034716-034616	Undefined



Function	Setting Range	RW
Assuming the set value = n, upon a start trigger the timer starts counting the count source and stops after counting it n times. The positive and negative phases change at the same time timer A, A2 or A4 stops.	000016 to FFFF16	WO

NOTES:
1. The register must be accessed in 16 bit units.
2. When the timer Ai register is set to 000016, the counter does not operate and a timer Ai interrupt does not occur.
3. Use MOV instruction to write to these registers.
4. If the INV15 bit is 0 (dead time timer enable), the positive or negative phase whichever is going from an inactive to an active level changes at the same time the dead time timer stops.
5. If the INV11 bit is 0 (three-phase mode 0), the TAi register value is transferred to the reload register by a timer Ai (i = 1, 2 or 4) start trigger.
   If the INV11 bit is 1 (three-phase mode 1), the TAi1 register value is transferred to the reload register by a timer Ai start trigger first and then the TAi register value is transferred to the reload register by the next timer Ai start trigger. Thereafter, the TAi1 register and TAi register values are transferred to the reload register alternately.
6. Do not write to TAi1 registers synchronously with a timer B2 underflow In three-phase mode 1.
7. Write to the TAi1 register as follows:
   (1) Write a value to the TAi1 register
   (2) Wait for one cycle of timer Ai count source.
   (3) Write the same value to the TAi1 register again.

Figure 12.29 TA1, TA2, TA4, TA11, TA21, and TA41 Registers

Timer B2 Special Mode Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	0	0					

Symbol	Address	After Reset
TB2SC	039E16	X0000000₂

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 reload timing switch bit (2)	0: Timer B2 underflow 1: Timer A output at odd-numbered	RW
IVPCR1	Three-phase output port $\overline{SD}$ control bit 1 (3, 4, 7)	0: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled 1: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled	RW
TB0EN	Timer B0 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode (5)	RW
TB1EN	Timer B1 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode (5)	RW
TB2SEL	Trigger select bit (6)	0: TB2 interrupt 1: Underflow of TB2 interrupt generation frequency setting counter [ICTB2]	RW
$\overline{(b6\text{-}b5)}$	Reserved bits	Set to 0	RW
— (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is 0.		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. If the INV11 bit is 0 (three-phase mode 0) or the INV06 bit is 1 (triangular wave modulation mode), set this bit to 0 (timer B2 underflow).
3. When setting the IVPCR1 bit to 1 (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), Set the PD8$_5$ bit to 0 (= input mode).
4. Related pins are U(P8$_0$), $\overline{U}$(P8$_1$), V(P7$_2$), $\overline{V}$(P7$_3$), W(P7$_4$), $\overline{W}$(P7$_5$). When a high-level ("H") signal is applied to the $\overline{SD}$ pin and set the IVPCR1 bit to 0 after forcible cutoff, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are exit from the high-impedance state. If a low-level ("L") signal is applied to the $\overline{SD}$ pin, three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is 0, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ become programmable I/O ports. When the IVPCR1 bit is set to 1, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are placed in a high-impedance state regardless of which function of those pins is used.
5. When this bit is used in delayed trigger mode 0, set bits TB0EN and TB1EN to 1 (A/D trigger mode).
6. When setting the TB2SEL bit to 1 (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), set the INV02 bit to 1 (three-phase motor control timer function).
7. Refer to "**18.6 Digital Debounce Function**" for the $\overline{SD}$ input.

The effect of $\overline{SD}$ pin input is below.
1.Case of INV03 = 1(Three-phase motor control timer output enabled)

IVPCR1 bit	$\overline{SD}$ pin inputs(3)	Status of U/V/W pins	Remarks
1 (Three-phase output forcible cutoff enable)	H	Three-phase PWM output	
	L(1)	High impedance(4)	Three-phase output forcible cutoff
0 (Three-phase output forcible cutoff disable)	H	Three-phase PWM output	
	L(1)	Input/output port(2)	

NOTES:
1. When "L" is applied to the $\overline{SD}$ pin, INV03 bit is changed to 0 at the same time.
2. The value of the port register and the port direction register becomes effective.
3. When $\overline{SD}$ function is not used, set to 0 (Input) in PD8$_5$ and pullup to "H" in $\overline{SD}$ pin from outside.
4. To leave the high-impedance state and restart the three-phase PWM signal output after the three-phase PWM signal output forced cutoff, set the IVPCR1 bit to 0 after the $\overline{SD}$ pin input level becomes high ("H").

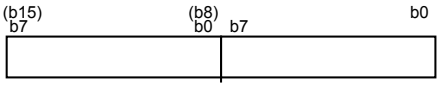
2.Case of INV03 = 0(Three-phase motor control timer output disabled)

IVPCR1 bit	$\overline{SD}$ pin inputs	Status of U/V/W pins	Remarks
1 (Three-phase output forcible cutoff enable)	H	Peripheral input/output or input/output port	
	L	High impedance	Three-phase output forcible cutoff(1)
0 (Three-phase output forcible cutoff disable)	H	Peripheral input/output or input/output port	
	L	Peripheral input/output or input/output port	

NOTE:
1. The three-phase output forcible cutoff function becomes effective if the INPCR1 bit is set to 1 (three-phase output forcible cutoff function enable) even when the INV03 bit is 0 (three-phase motor control timer output disalbe)

Figure 12.30 TB2SC Register

Timer B2 Register (1)

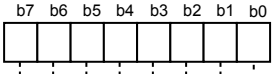


Symbol	Address	After Reset
TB2	0395$_{16}$-0394$_{16}$	Undefined

Function	Setting Range	RW
Divide the count source by n + 1 where n = set value. Timer A1, A2 and A4 are started at every occurrence of underflow.	0000$_{16}$ to FFFF$_{16}$	RW

NOTE:
1. Access the register by 16 bit units.

Trigger Select Register

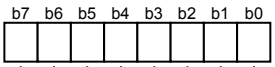


Symbol	Address	After Reset
TRGSR	0383$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	To use the V-phase output control circuit, set these bits to "01$_2$"(TB2 underflow).	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	To use the W-phase output control circuit, set these bits to "01$_2$"(TB2 underflow).	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3$_{IN}$ is selected (1) 0 1 : TB2 is selected (2) 1 0 : TA2 is selected (2) 1 1 : TA4 is selected (2)	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	To use the U-phase output control circuit, set these bits to "01$_2$"(TB2 underflow).	RW
TA4TGH			RW

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

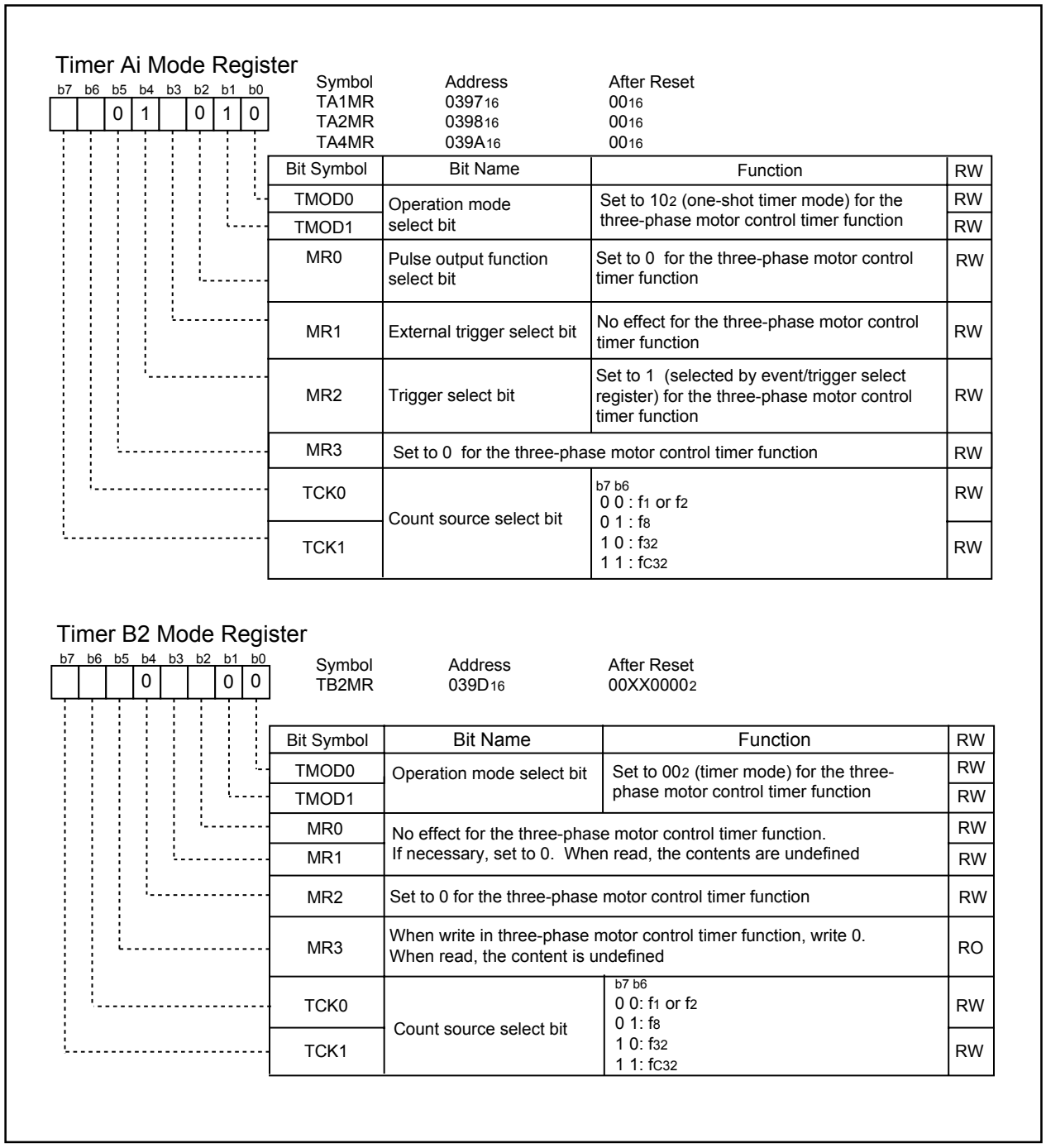
Count Start Flag



Symbol	Address	After reset
TABSR	0380$_{16}$	00$_{16}$

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

Figure 12.31 TB2 Register, TRGSR Register, and TABSR Register



Timer Ai Mode Register

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

Symbol	Address	After Reset
TA1MR	039716	0016
TA2MR	039816	0016
TA4MR	$039A16$	0016

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	Set to 102 (one-shot timer mode) for the three-phase motor control timer function	RW
TMOD1			RW
MR0	Pulse output function select bit	Set to 0 for the three-phase motor control timer function	RW
MR1	External trigger select bit	No effect for the three-phase motor control timer function	RW
MR2	Trigger select bit	Set to 1 (selected by event/trigger select register) for the three-phase motor control timer function	RW
MR3	Set to 0 for the three-phase motor control timer function		RW
TCK0	Count source select bit	b7 b6 0 0 : f_1 or f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	RW
TCK1			RW

Timer B2 Mode Register

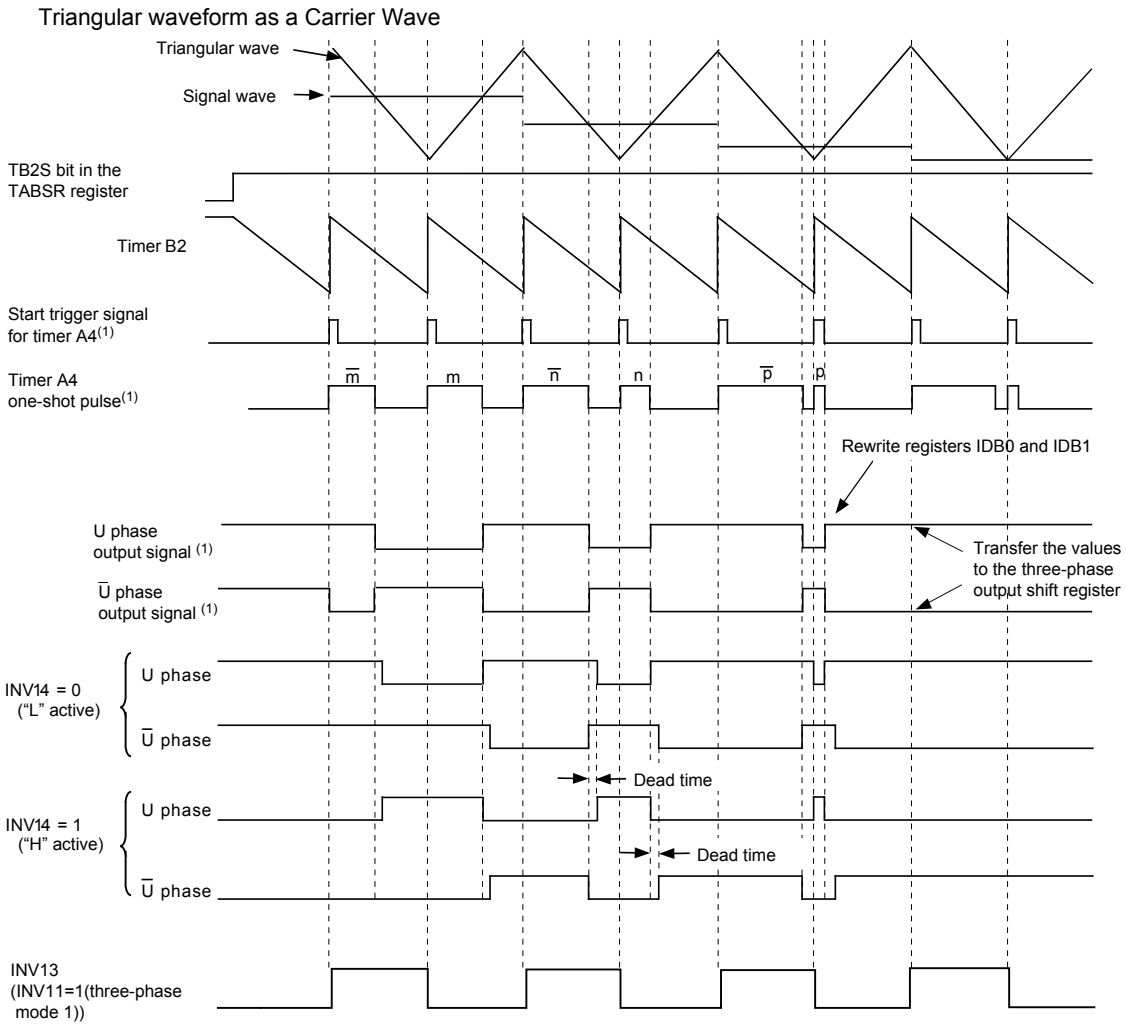
b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

Symbol	Address	After Reset
TB2MR	$039D16$	$00XX0000_2$

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	Set to 002 (timer mode) for the three-phase motor control timer function	RW
TMOD1			RW
MR0	No effect for the three-phase motor control timer function. If necessary, set to 0. When read, the contents are undefined		RW
MR1			RW
MR2	Set to 0 for the three-phase motor control timer function		RW
MR3	When write in three-phase motor control timer function, write 0. When read, the content is undefined		RO
TCK0	Count source select bit	b7 b6 0 0: f_1 or f_2 0 1: f_8 1 0: f_{32} 1 1: f_{C32}	RW
TCK1			RW

Figure 12.32 TA1MR, TA2MR, TA4MR, and TB2MR Registers

The three-phase motor control timer function is enabled by setting the INV02 bit in the INVC0 register to 1. When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$). The dead time is controlled by a dedicated dead-time timer. **Figure 12.33** shows the example of triangular modulation waveform, and **Figure 12.34** shows the example of sawtooth modulation waveform.



NOTE:
1. Internal signals. See **Figure 12.25**.

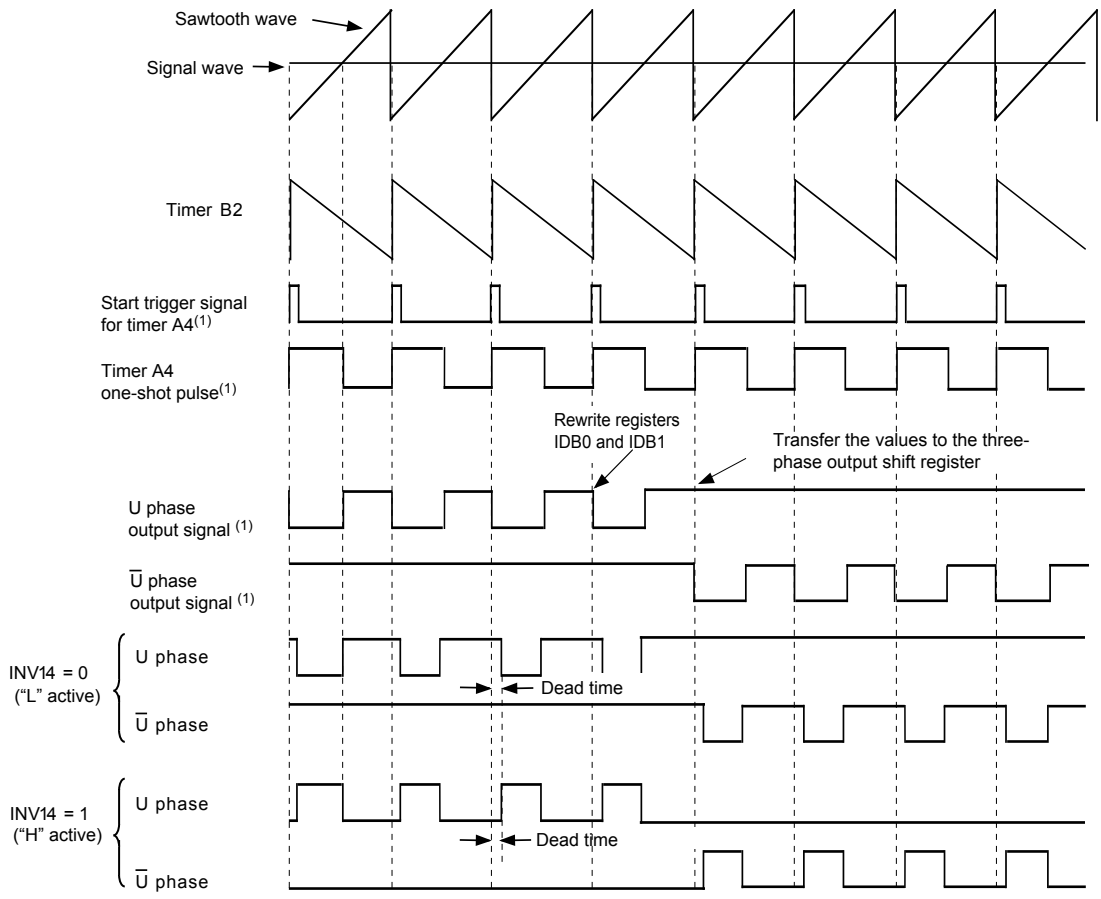
The above applies under the following conditions:
INVC0 = 00XX11XX2 (X varies depending on each system) and INVC1 = 010XXXX02.
Examples of PWM output change are:

(1)When INV11 = 1 (three-phase mode 1)
- INV01 = 0 and ICTB2 = 2$_{16}$ (the timer B2 interrupt is generated every two times the timer B2 underflows), or INV01 = 1, INV00 = 1, and ICTB2=1$_{16}$ (the timer B2 interrupt is generated at the falling edge of the timer A1 reload control signal.)
- Default value of the timer: TA41 = $\overline{m}$, TA4 = m. Registers TA4 and TA41 are changed whenever the timer B2 interrupt is generated. First time, TA41 = $\overline{n}$, TA4 = n. Second time, TA41 = $\overline{p}$, TA4 = p.
- Default values of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1. They are changed to DU0 = 1, DUB0 = 0, DU1= 1 and DUB1 = 0 when the third timer B2 interrupt is generated.

(2)When INV11 = 0 (three-phase mode 0)
- INV01 = 0, ICTB2 = 1$_{16}$ (the timer B2 interrupt is generated whenever timer B2 underflows)
- Default value of the timer: TA4 = $\overline{m}$. The TA4 register is changed whenever the timer B2 interrupt is generated. First time: TA4 = m. Second tim:, TA4 = $\overline{n}$. Third time: TA4 = n. Fourth time: TA4 = $\overline{p}$. Fifth time: TA4 = p.
- Default values of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1. They are changed to DU0 = 1, DUB0 = 0, DU1 = 1, and DUB1 = 0 when the sixth timer B2 interrupt is generated.

The value written to registers TA4 and TA41 becomes effective at the rising edge of the timer A1 reload control signal.

Figure 12.33 Triangular Wave Modulation Operation



NOTE:
1. Internal signals. See **Figure 12.25**.

The above applies under the following conditions:
INVC0 = 01XX110X2 (X varies depending on each system) and INVC1 = 010XXX002.

Examples of PWM output change are:
• Default value of registers IDB0 and IDB1: DU0=0, DUB0=1, DU1=1, DUB1=1.
They are changed to DU0=1, DUB0=0, DU1=1, DUB1=1 when the timer B2 interrupt is generated.

Figure 12.34 Sawtooth Wave Modulation Operation

12.3.1 Position-Data-Retain Function

This function is used to retain the position data synchronously with the three-phase waveform output.There are three position-data input pins for U, V, and W phases.

A trigger to retain the position data (hereafter, this trigger is referred to as "retain trigger") can be selected by the PDRT bit in the PDRF register. This bit selects the retain trigger to be the falling edge of each positive phase, or the rising edge of each positive phase.

12.3.1.1 Operation of the Position-data-retain Function

Figure 12.35 shows a usage example of the position-data-retain function (U phase) when the retain trigger is selected as the falling edge of the positive signal.

(1) At the falling edge of the U-phase waveform ouput, the state at pin IDU is transferred to the PDRU bit in the PDRF register.

(2) Until the next falling edge of the Uphase waveform output,the above value is retained.

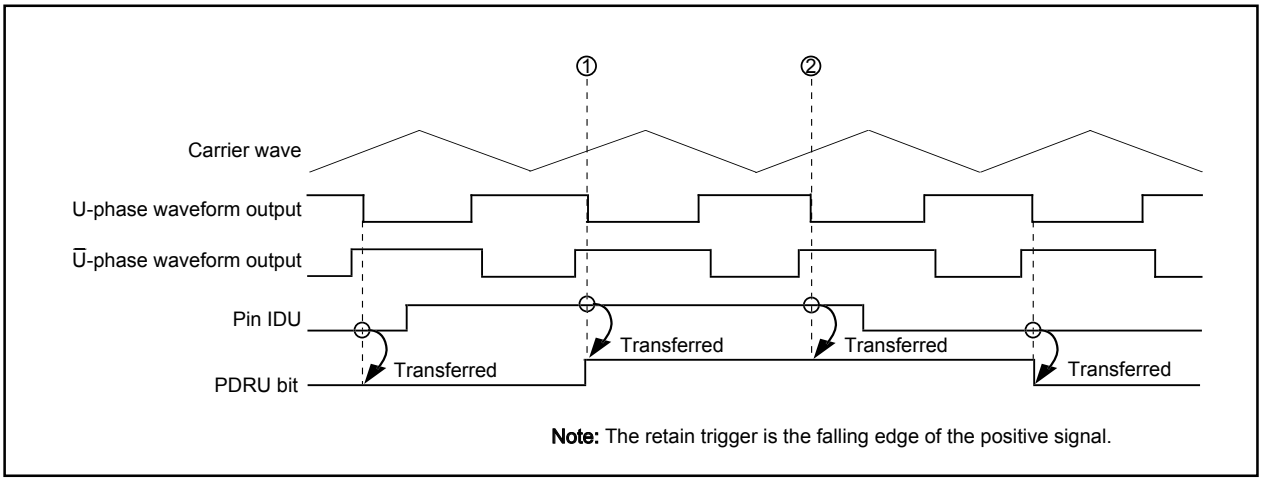


Figure 12.35 Usage Example of Position-data-retain Function (U phase)

12.3.1.2 Position-data-retain Function Control Register

Figure 12.36 shows the structure of the position-data-retain function contol register.

Position-data-retain Function Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
×	×	×	×				

Symbol	Address	After Reset
PDRF	$034E_{16}$	$XXXX\ 0000_2$

Bit Symbol	Bit Name	Function	RW
PDRW	W-phase position data retain bit	Input level at pin IDW is read out. 0: "L" level 1: "H" level	RO
PDRV	V-phase position data retain bit	Input level at pin IDV is read out. 0: "L" level 1: "H" level	RO
PDRU	U-phase position data retain bit	Input level at pin IDU is read out. 0: "L" level 1: "H" level	RO
PDRT	Retain-trigger polarity select bit	0: Rising edge of positive phase 1: Falling edge of positive phase	RW
(b7-b4)	Nothing is assigned. When write, set to 0. When read, contents are undefined		—

NOTE:
1.This register is valid only in the three-phase mode.

Figure 12.36 PDRF Register

12.3.1.2.1 W-phase Position Data Retain Bit (PDRW)

This bit is used to retain the input level at pin IDW.

12.3.1.2.2 V-phase Position Data Retain Bit (PDRV)

This bit is used to retain the input level at pin IDV.

12.3.1.2.3 U-phase Position Data Retain Bit (PDRU)

This bit is used to retain the input level at pin IDU.

12.3.1.2.4 Retain-trigger Polarity Select Bit (PDRT)

This bit is used to select the trigger polarity to retain the position data.

When this bit is set to 0, the rising edge of each positive phase selected.

When this bit is set to 1, the falling edge of each pocitive phase selected.

12.3.2 Three-phase/Port Output Switch Function

When the INVC03 bit in the INVC0 register set to 1 (Timer output enabled for three-phase motor control) and setting the PFCi (i=0 to 5) in the PFCR register to 0 (I/O port), the three-phase PWM output pin (U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$) functions as I/O port. Each bit of the PFCi bits (i=0 to 5) is applicable for each one of three-phase PWM output pins. **Figure 12.37** shows the example of three-phase/port output switch function. **Figure 12.38** shows the PFCR register and the three-phase protect control register.

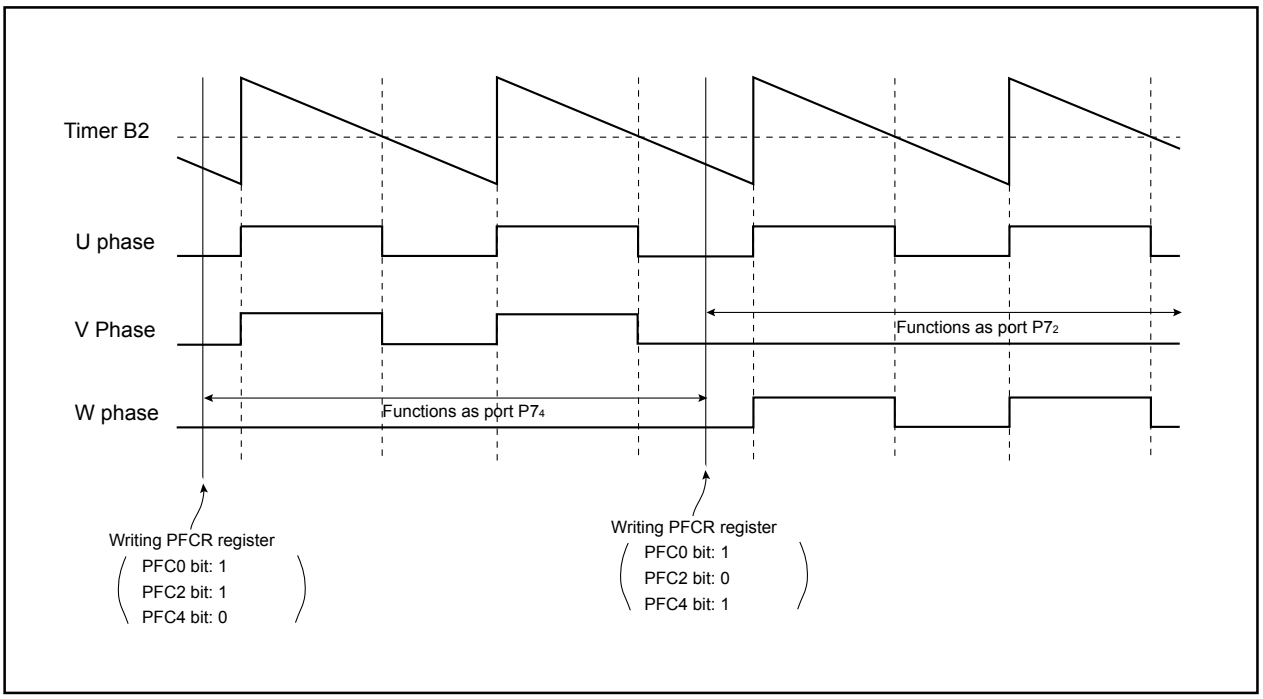
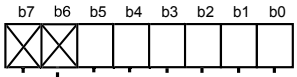


Figure 12.37 Usage Example of Three-phse/Port Output Switch Function

Port Function Control Register (1)

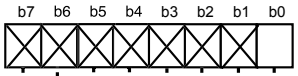


Symbol	Address	After Reset
PFCR	0358_{16}	$0011\ 1111_2$

Bit Symbol	Bit Name	Function	RW
PFC0	Port $P8_0$ output function select bit	0: Input/Output port $P8_0$ 1: Three-phase PWM output (U phase output)	RW
PFC1	Port $P8_1$ output function select bit	0: Input/Output port $P8_1$ 1: Three-phase PWM output ($\overline{U}$ phase output)	RW
PFC2	Port $P7_2$ output function select bit	0: Input/Output port $P7_2$ 1: Three-phase PWM output (V phase output)	RW
PFC3	Port $P7_3$ output function select bit	0: Input/Output port $P7_3$ 1: Three-phase PWM output ($\overline{V}$ phase output)	RW
PFC4	Port $P7_4$ output function select bit	0: Input/Output port $P7_4$ 1: Three-phase PWM output (W phase output)	RW
PFC5	Port $P7_5$ output function select bit	0: Input/Output port $P7_5$ 1: Three-phase PWM output ($\overline{W}$ phase output)	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. When write, set to 0. When read, these contents are 0		—

NOTE:
1. This register is valid only when the INVC03 bit in the INVC0 register is 1 (Three-phase motor control timer.

Three-phase Protect Control Register



Symbol	Address	After Reset
TPRC	$025A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TPRC0	Three-phase protect control bit	Enable write to PFCR register 0: Write protected 1: Write enabled	RW
$\overline{\text{(b7-b1)}}$	Nothing is assigned. If necessary, set to 0. When read, the contents are 0		—

Figure 12.38 PFCR Register, and TPRC Register

13. Timer S

The Timer S (Input Capture/Output Compare : here after, Timer S is referred to as "IC/OC".) is a high-performance I/O port for time measurement and waveform generation.
The IC/OC has one 16-bit base timer for free-running operation and eight 16-bit registers for time measurement and waveform generation.
Table 13.1 lists functions and channels of the IC/OC.

Table 13.1 IC/OC Functions and Channels

Function		Description
Time measurement (1)		8 channels
	Digital filter	8 channels
	Trigger input prescaler	2 channels
	Trigger input gate	2 channels
Waveform generation (1)		8 channels
	Single-phase waveform output	Available
	Phase-delayed waveform output	Available
	Set/Reset waveform output	Available

NOTE:
1. The time measurement function and the waveform generating function share a pin.
The time measurement function or waveform generating function can be selected for each channel.

Figure 13.1 shows the block diagram of the IC/OC.

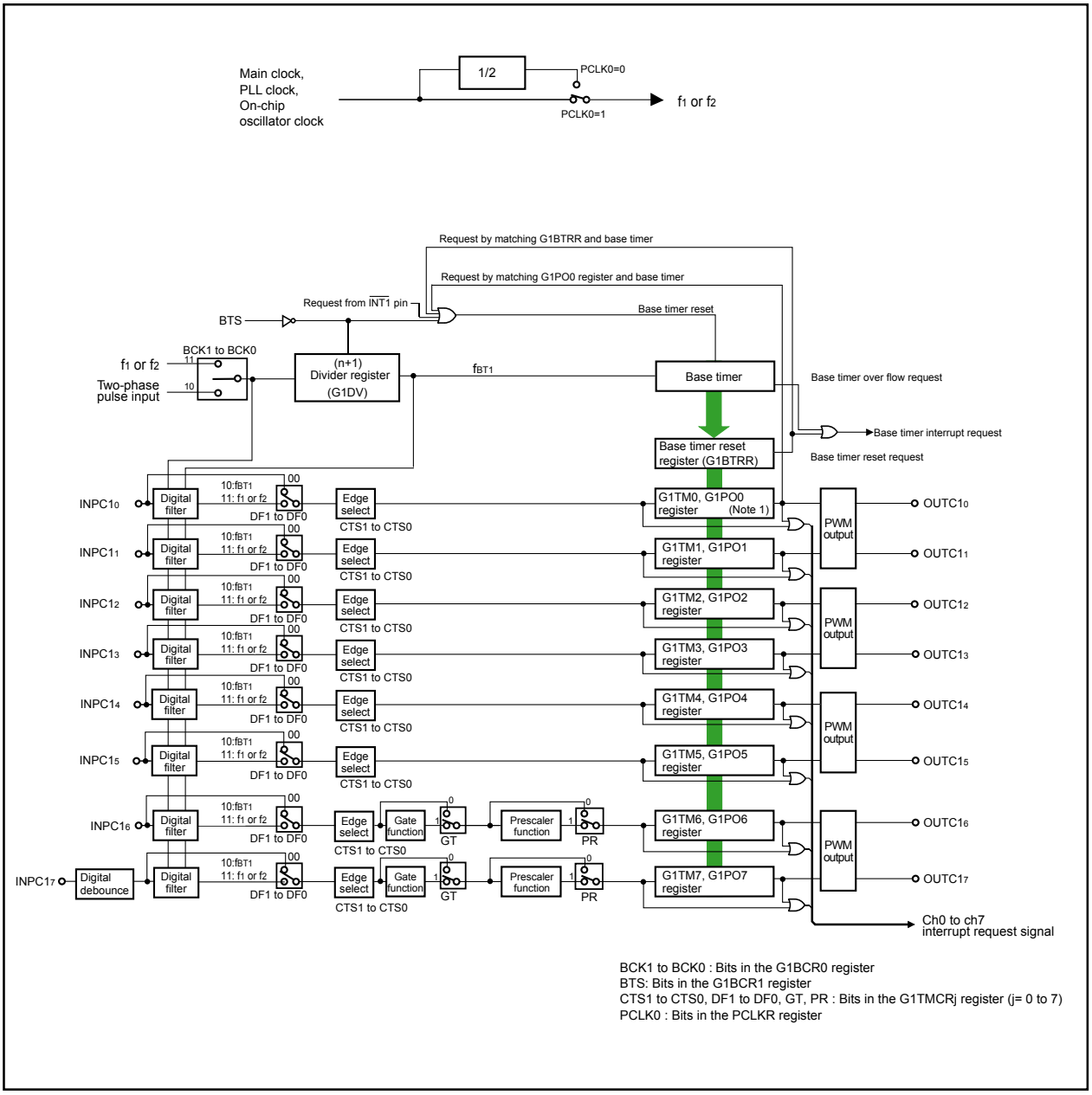
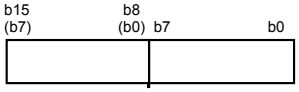


Figure 13.1 IC/OC Block Diagram

Figures 13.2 to **13.10** show registers associated with the IC/OC base timer, the time measurement function, and the waveform generating function.

Base Timer Register(1)

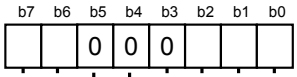


Symbol	Address	After Reset
G1BT	0321_{16} - 0320_{16}	Indeterminate

Function	Setting Range	RW
• When the base timer is operating: When read, the value of base timer plus 1 can be read. When write, the counter starts counting from the value written. When the base timer is reset, this register is set to "0000_{16}". (2) • When the base timer is reset: This register is set to "0000_{16}" but a value read is indeterminate. No value is written (2)	0000_{16} to $FFFF_{16}$	RW

NOTES:
1. The G1BT register reflects the value of the base timer, synchronizing with the count source f_{BT1} cycles.
2. This base timer stops only when the BCK1 to BCK0 bits in the G1BCR0 register are set to "00_2" (count source clock stop). The base timer operates when the BCK1 to BCK0 bits are set to other than "00_2". When the BTS bit in the G1BCR1 register is set to "0", the base timer is reset continuously, and remaining set to "0000_{16}". When the BTS bit is set to "1", this state is cleared and the timer starts counting.

Base Timer Control Register 0



Symbol	Address	After Reset
G1BCR0	0322_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
BCK0	Count source select bit	b1b0 0 0: Clock stop 0 1: Do not set to this value 1 0: Two-phase input (1) 1 1: f_1 or f_2 (2)	RW
BCK1			RW
RST4	Base timer reset cause select bit 4	0: Do not reset Base timer by matching G1BTRR 1: Reset Base timer by matching G1BTRR(3)	RW
(b5-b3)	Reserved bit	Set to 0	RW
CH7INSEL	Channel 7 input select bit	0: P2$_7$/OUTC1$_7$/INPC1$_7$ pin 1: P1$_7$/$\overline{\text{INT5}}$/INPC1$_7$/IDU pin	RW
IT	Base timer interrupt bit	0: Bit 15 overflow 1: Bit 14 overflow	RW

NOTES:
1. This setting can be used when bits UD1 to UD0 in the G1BCR1 register are set to 10_2 (two-phase signal processing mode). Do not set bits BCK1 to BCK0 to 10_2 in other modes.
2. When the PCLK0 bit in the PCLKR register is set to 0, the count source is f_2 cycles. And when the PCLK0 bit is set to set to 1, the count source is f_1 cycles.
3. When the RST4 bit is set to 1, set the RST1 bit in the G1BCR1 register to 0.

Figure 13.2 G1BT and G1BCR0 Registers

Divider Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
G1DV	$032A_{16}$	00_{16}

Function	Setting range	RW
Divide f_1, f_2 or two-phase pulse input by (n+1) for f_{BT1} clock cycles generation. n: the setting value of the G1DV register	00_{16} to FF_{16}	RW

Base Timer Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
0				0			0

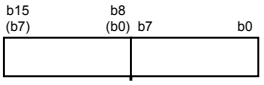
Symbol	Address	After Reset
G1BCR1	0323_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
$\overline{\text{(b0)}}$	Reserved bit	Set to 0	RW
RST1	Base timer reset cause select bit 1	0: The base timer is not reset by matching the G1PO0 register 1: The base timer is reset by matching with the G1PO0 register (1)	RW
RST2	Base timer reset cause select bit 2	0: The base timer is not reset by applying "L" to the $\overline{\text{INT1}}$ pin 1: The base timer is reset by applying "L" to the $\overline{\text{INT1}}$ pin	RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
BTS	Base timer start bit	0: Base timer is reset 1: Base timer starts counting	RW
UD0	Counter increment/ decrement control bit	b6b5 0 0: Counter increment mode 0 1: Counter increment/decrement mode 1 0: Two-phase pulse signal processing mode 1 1: Do not set to this value	RW
UD1			RW
$\overline{\text{(b7)}}$	Reserved bit	Set to 0	RW

NOTE:
1. The base timer is reset two f_{BT1} clock cycles after the base timer matches the value set in the G1PO0 register. (See **Figure 13.7** for details on the G1PO0 register) When the RST1 bit is set to 1, the value of the G1POj register (j=1 to 7) for the waveform generating function must be set to a value smaller than that of the G1PO0 register.
When the RST1 bit is set to 1, set the RST4 bit in the G1BCR0 register to 0.

Figure 13.3 G1DV Register and G1BCR1 Register

Base Timer Reset Register(1)



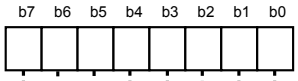
Symbol	Address	After Reset
G1BTRR	0329_{16} - 0328_{16}	Undefined

Function	Setting Range	RW
When enabled by the RST4 bit in the G1BCR0 register, the base timer is reset by matching the G1BTRR register setting value and the base timer setting value.	0000_{16} to $FFFF_{16}$	RW

NOTE:
1. The G1BTRR register reflects the value of the base timer, synchronizing with the count source f_{BT1} cycles.

Figure 13.4 G1BTRR Register

Time Measurement Control Register j (j=0 to 7)

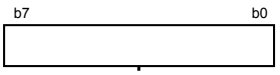


Symbol	Address	After Reset
G1TMCR0 to G1TMCR3	0318_{16}, 0319_{16}, $031A_{16}$, $031B_{16}$	00_{16}
G1TMCR4 to G1TMCR7	$031C_{16}$, $031D_{16}$, $031E_{16}$, $031F_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
CTS0	Time measurement trigger select bit	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 function select bit	b3 b2 0 0: No digital filter 0 1: Do not set to this value 1 0: f_{BT1} 1 1: f_1 or f_2 (1)	RW
DF1			RW
GT	Gate function select bit (2)	0: Gate function is not used 1: Gate function is used	RW
GOC	Gate function clear select bit (2, 3, 4)	0: Not cleared 1: The gate is cleared when the base timer matches the G1POk register	RW
GSC	Gate function clear bit (2, 3)	The gate is cleared by setting the GSC bit to 1	RW
PR	Prescaler function select bit (2)	0: Not used 1: Used	RW

NOTES:
1. When the PCLK0 bit in the PCLKR register is set to 0, the count source is f_2 cycles. And when the PCLK0 bit is set to 1, the count source is f_1 cycles.
2. These bits are in registers G1TMCR6 and G1TMCR7. Set all bits 4 to 7 in registers G1TMCR0 to G1TMCR5 to 0.
3. These bits are enabled when the GT bit is set to 1.
4. The GOC bit is set to 0 after the gate function is cleared. See **Figure 13.7** for details on the G1POk register (k=4 when j=6 and k=5 when j=7).

Time Measurement Prescale Register j (j=6,7)(1)



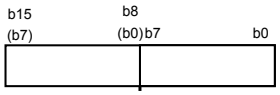
Symbol	Address	After Reset
G1TPR6 to G1TPR7	0324_{16}, 0325_{16}	00_{16}

Function	Setting Range	RW
As the setting value is n, time is measured whenever a trigger input is counted by *n+1* (2)	00_{16} to FF_{16}	RW

NOTES:
1. The G1TPR6 to G1TPR7 registers reflect the base timer value, synchronizing with the count source f_{BT1} cycles.
2. The first prescaler, after the PR bit in the G1TMCRj register is changed from 0 (not used) to 1 (used), may be divided by *n*, rather than *n+1*. The subsequent prescaler is divided by *n+1*.

Figure 13.5 G1TMCR0 to G1TMCR7 Registers, and G1TPR6 to G1TPR7 Registers

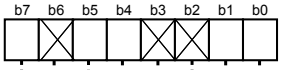
Waveform Generation Register j (j=0 to 7)



Symbol	Address	After Reset
G1TM0 to G1TM2	0301$_{16}$-0300$_{16}$, 0303$_{16}$-0302$_{16}$, 0305$_{16}$-0304$_{16}$	Indeterminte
G1TM3 to G1TM5	0307$_{16}$-0306$_{16}$, 0309$_{16}$-0308$_{16}$, 030B$_{16}$-030A$_{16}$	Indeterminte
G1TM6 to G1TM7	030D$_{16}$-030C$_{16}$, 030F$_{16}$-030E$_{16}$	Indeterminte

Function	Setting Range	RW
The base timer value is stored every measurement timing	———	RO

Waveform Generation Control Register j (j=0 to 7)



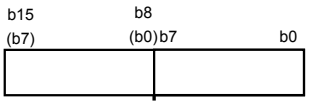
Symbol	Address	After Reset
G1POCR0 to G1POCR3	0310$_{16}$, 0311$_{16}$, 0312$_{16}$, 0313$_{16}$	0X00 XX00$_2$
G1POCR4 to G1POCR7	0314$_{16}$, 0315$_{16}$, 0316$_{16}$, 0317$_{16}$	0X00 XX00$_2$

Bit Symbol	Bit Name	Function	RW
MOD0	Operating mode select bit	b1b0 00: Single waveform output mode 01: SR waveform output mode (1) 10: Phase-delayed waveform output mode 11: Do not set to this value	RW
MOD1			RW
——— (b3-b2)	Nothing is assigned. If necessary, set to 0. When read, their contents are undefined		—
IVL	Output initial value select bit(4)	0: "L" output as a default value 1: "H" output as a default value	RW
RLD	G1POj register value reload timing select bit	0: Reloads the G1POj register when value is written 1: Reloads the G1POj register when the base timer is reset	RW
——— (b6)	Nothing is assigned. If necessary, set to 0. When read, its content is undefined		—
INV	Inverse output function select bit (2)	0: Output is not inversed 1: Output is inversed	RW

NOTES :
1. This setting is enabled only for even channels. In SR waveform output mode, values written to the corresponding odd channel (next channel after an even channel) are ignored. Even channels provide waveform output. Odd channels provide no waveform output.
2. The inverse output function is the final step in waveform generating process. When the INV bit is set to 1, and "H" signal is provided a default output by setting the IVL bit to 0, and an "L" signal is provided by setting it to 1.
3. In the SR waveform output mode, set not only the even channel but also the correspoinding even channel (next channel after the even channel).
4. To provide either "H" or "L" signal output set in the IVL bit, set the FSCj bit in the G1FS register to 0 (select waveform generating function) and IFEj bit in the G1FE register to 1 (functions for channel j enabled). Then set the IVL bit to 0 or 1.

Figure 13.6 G1TM0 to G1TM7 Registers, and G1POCR0 to G1POCR7 Registers

Waveform Generation Register j (j=0 to 7)



Symbol	Address	After Reset
G1PO0 to G1PO2	0301_{16}-0300_{16}, 0303_{16}-0302_{16}, 0305_{16}-0304_{16}	Undefined
G1PO3 to G1PO5	0307_{16}-0306_{16}, 0309_{16}-0308_{16}, $030B_{16}$-$030A_{16}$	Undefined
G1PO6 to G1PO7	$030D_{16}$-$030C_{16}$, $030F_{16}$-$030E_{16}$	Undefined

Function	Setting Range	RW
• When the RLD bit in the G1POCRj register is set to 0, value written is immediately reloaded into the G1POj register for output, for example, a waveform output,reflecting the value. • When the RLD bit is set to 1, value reloaded while the base timer is reset. The value written can be read until reloaded	0000_{16} to $FFFF_{16}$	RW

Figure 13.7 G1PO0 to G1PO7 Registers

Function Select Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
G1FS	0327_{16}	00_{16}

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

Function Enable Register(1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
G1FE	0326_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
IFE0	Channel 0 function enable bit	0 : Disable function s for channel j (2) 1 : Enable functions for channel j (j=0 to 7)	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

NOTES:
1. The G1FE register reflects the base timer value, synchronizing with the count source f_{BT1} cycles.
2. When functions for the channel j are disabled, each pin functions as an I/O port.

Figure 13.8 G1FS and G1FE Registers

Interrupt Request Register (1)

b7	b6	b5	b4	b3	b2	b1	b0

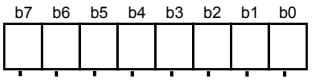
Symbol	Address	After Reset
G1IR	0330_{16}	Undefined

Bit Symbol	Bit Name	Function	RW
G1IR0	Interrupt request, Ch0	0 : No interrupt request 1 : Interrupt requested	RW
G1IR1	Interrupt request, Ch1		RW
G1IR2	Interrupt request, Ch2		RW
G1IR3	Interrupt request, Ch3		RW
G1IR4	Interrupt request, Ch4		RW
G1IR5	Interrupt request, Ch5		RW
G1IR6	Interrupt request, Ch6		RW
G1IR7	Interrupt request, Ch7		RW

NOTE:
1. When writing 0 to each bit in the G1IR register, use the following instruction:
AND, BCLR

Figure 13.9 G1IR Register

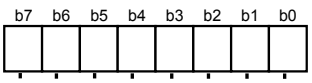
Interrupt Enable Register 0



Symbol	Address	After Reset
G1IE0	0331_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
G1IE00	Interrupt enable 0, CH0	0 : IC/OC interrupt 0 request disable 1 : IC/OC interrupt 0 request enable	RW
G1IE01	Interrupt enable 0, CH1		RW
G1IE02	Interrupt enable 0, CH2		RW
G1IE03	Interrupt enable 0, CH3		RW
G1IE04	Interrupt enable 0, CH4		RW
G1IE05	Interrupt enable 0, CH5		RW
G1IE06	Interrupt enable 0, CH6		RW
G1IE07	Interrupt enable 0, CH7		RW

Interrupt Enable Register 1



Symbol	Address	After Reset
G1IE1	0332_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
G1IE10	Interrupt enable 1, CH0	0 : IC/OC interrupt 1 request disable 1 : IC/OC interrupt 1 request enable	RW
G1IE11	Interrupt enable 1, CH1		RW
G1IE12	Interrupt enable 1, CH2		RW
G1IE13	Interrupt enable 1, CH3		RW
G1IE14	Interrupt enable 1, CH4		RW
G1IE15	Interrupt enable 1, CH5		RW
G1IE16	Interrupt enable 1, CH6		RW
G1IE17	Interrupt enable 1, CH7		RW

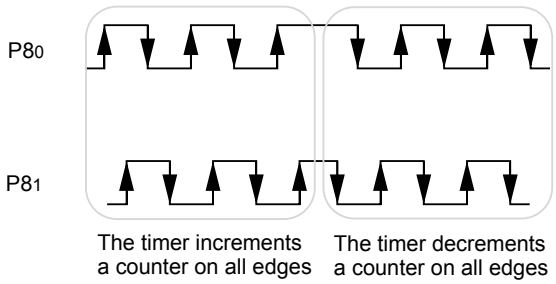
Figure 13.10 G1IE0 and G1IE1 Registers

13.1 Base Timer

The base timer is a free-running counter that counts an internally generated count source.

Table 13.2 lists specifications of the base timer. **Table 13.3** shows registers associated with the base timer. **Figure 13.11** shows a block diagram of the base timer. **Figure 13.12** shows an example of the base timer in counter increment mode. **Figure 13.13** shows an example of the base timer in counter increment/decrement mode. **Figure 13.14** shows an example of two-phase pulse signal processing mode.

Table 13.2 Base Timer Specifications

Item	Specification
Count source(f_{BT1})	f_1 or f_2 divided by *(n+1)*, two-phase pulse input divided by *(n+1)* n: determined by the DIV7 to DIV0 bits in the G1DV register. n=0 to 255 However, no division when n=0
Counting operation	The base timer increments the counter value The base timer increments/decrements the counter value Two-phase pulse signal processing
Count start condition	The BTS bit in the G1BCR1 register is set to 1 (base timer starts counting)
Count stop condition	The BTS bit in the G1BCR1 register is set to 0 (base timer reset)
Base timer reset condition	(1) The value of the base timer matches the value of the G1BTRR register (2) The value of the base timer matches the value of G1PO0 register. (3) Apply a low-level signal ("L") to external interrupt pin, $\overline{INT1}$ pin
Value for base timer reset	0000_{16}
Interrupt request	The base timer interrupt request is generated: (1) When the bit 14 or bit 15 in the base timer overflows (2) The value of the base timer value matches the value of the base timer reset register
Read from timer	• The G1BT register indicates a counter value while the base timer is running • The G1BT register is undefined when the base timer is reset
Write to timer	When a value is written while the base timer is running, the timer counter immediately starts counting from this value. No value can be written while the base timer is reset.
Selectable function	• Counter increment/decrement mode The base timer starts counting from 0000_{16}. After incrementing to $FFFF_{16}$, the timer counter is then decremented back to 0000_{16}. The base timer increments the counter value again when the timer counter reaches 0000_{16}. (See **Figure 13.13**) • Two-phase pulse processing mode Two-phase pulse signals from pins $P8_0$ and $P8_1$ are counted (See **Figure 13.14**) 

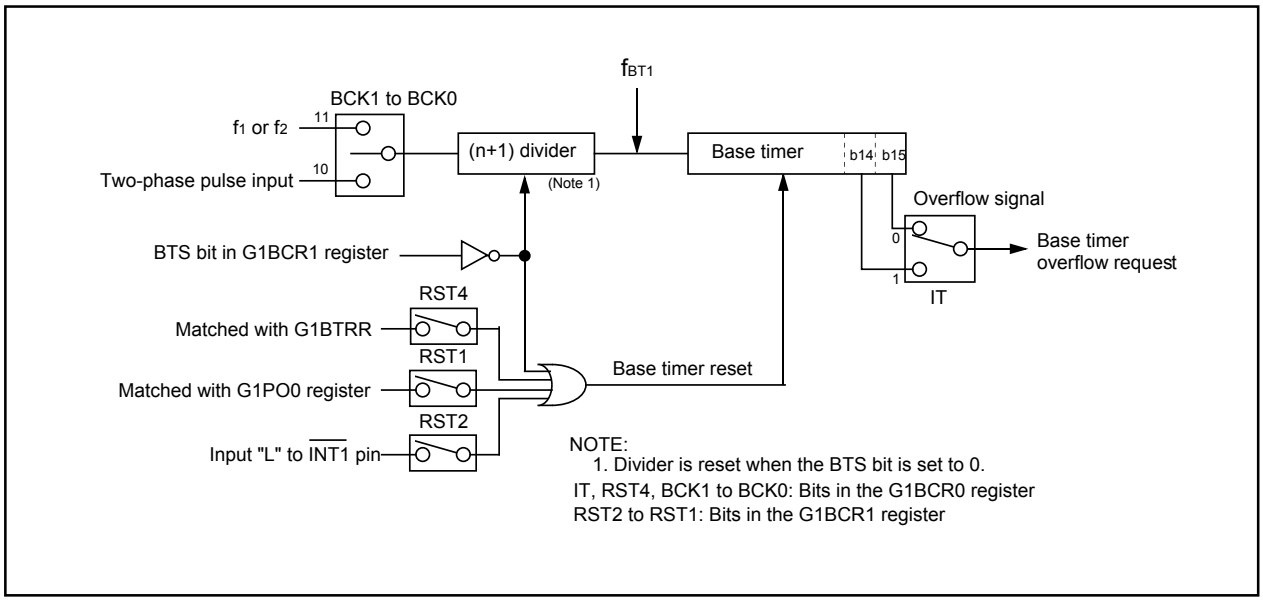


Figure 13.11 Base Timer Block Diagram

Table 13.3 Base Timer Associated Register Settings (Time Measurement Function, Waveform Generation Function, Communication Function)

Register	Bit	Function
G1BCR0	BCK1 to BCK0	Select a count source
	RST4	Select base timer reset timing
	IT	Select the base timer overflow
G1BCR1	RST2 to RST1	Select base timer reset timing
	BTS	Used to start the base timer
	UD1 to UD0	Select how to count
G1BT	-	Read or write base timer value
G1DV	-	Divide ratio of a count source

Set the following registers to set the RST1 bit to 1 (base timer reset by matching the base timer with the G1PO0 register)

G1POCR0	MOD1 to MOD0	Set to 00_2 (single-phase waveform output mode)
G1PO0	-	Set reset cycle
G1FS	FSC0	Set to 0 (waveform generating function)
G1FE	IFE0	Set to 1 (channel operation start)

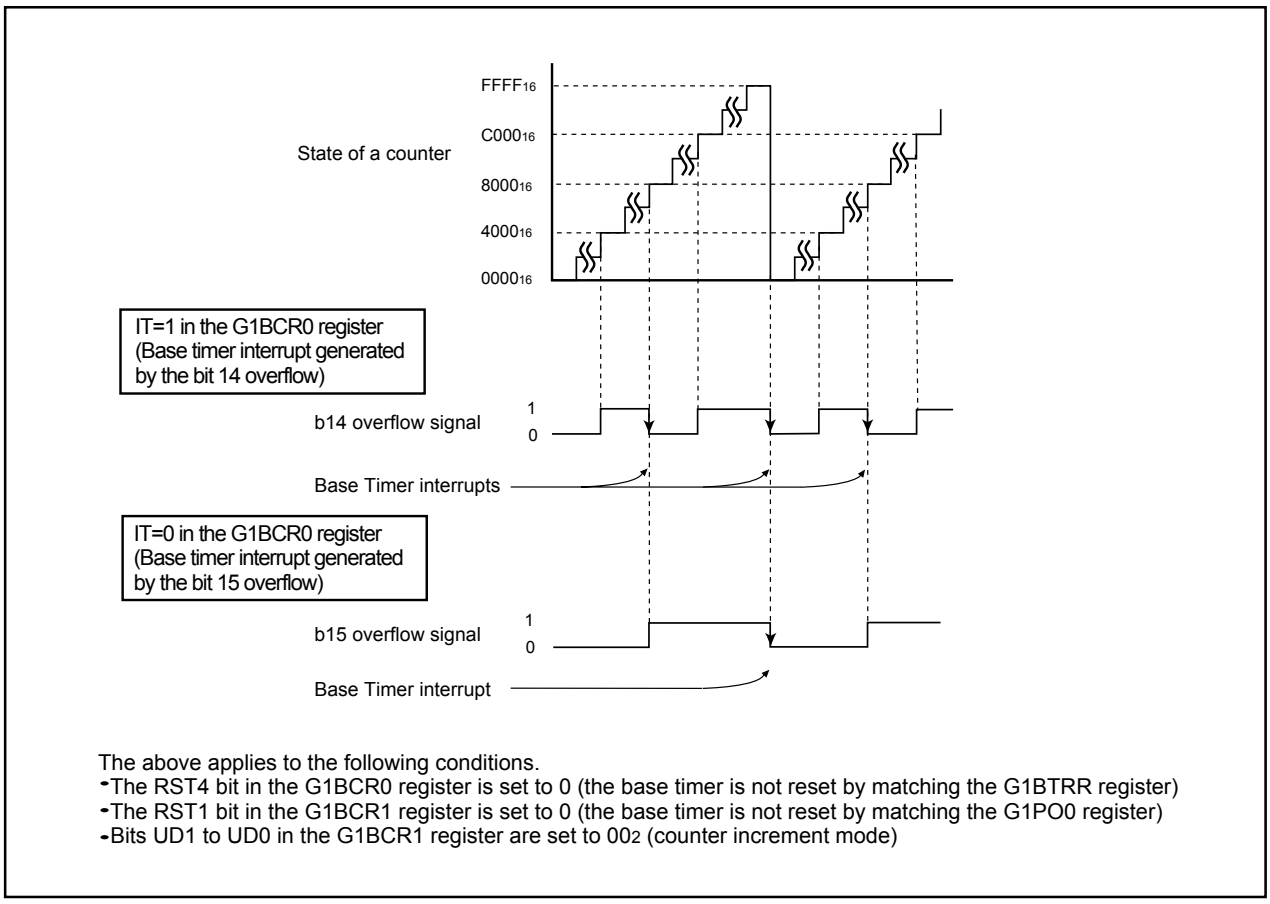


Figure 13.12 Counter Increment Mode

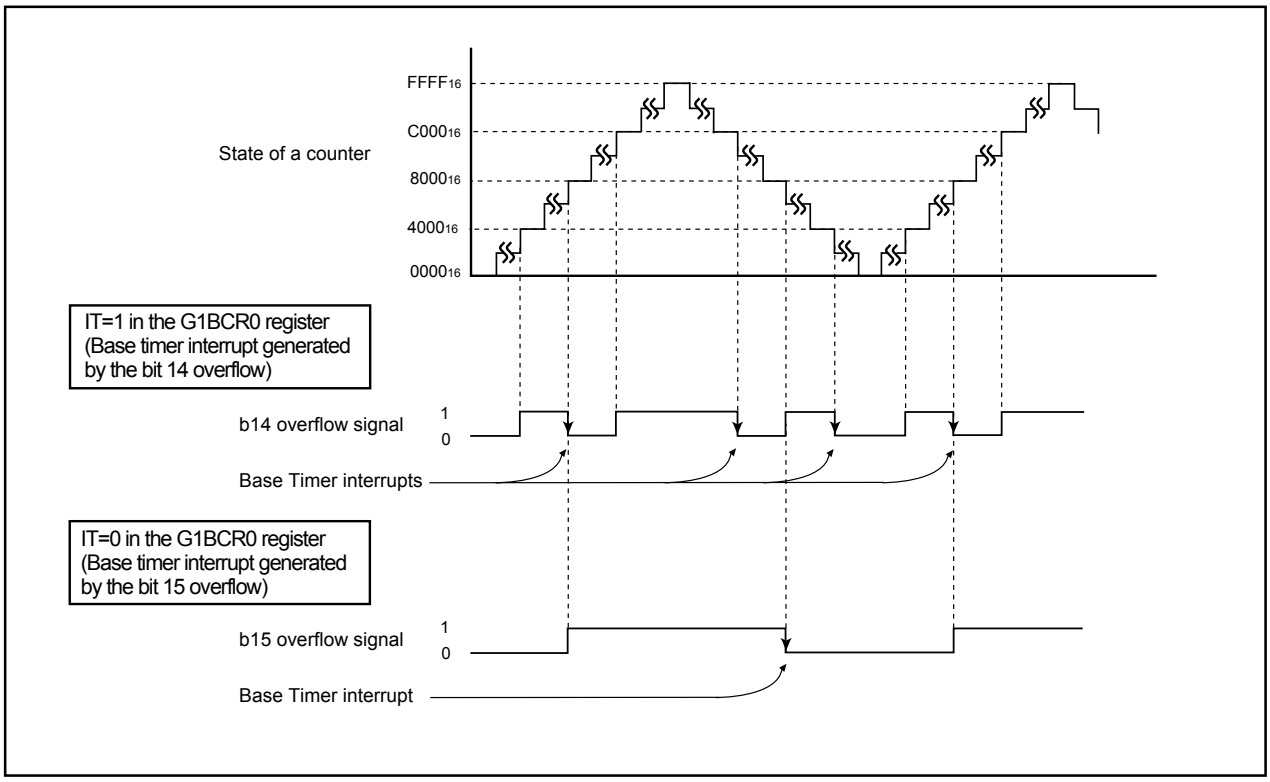
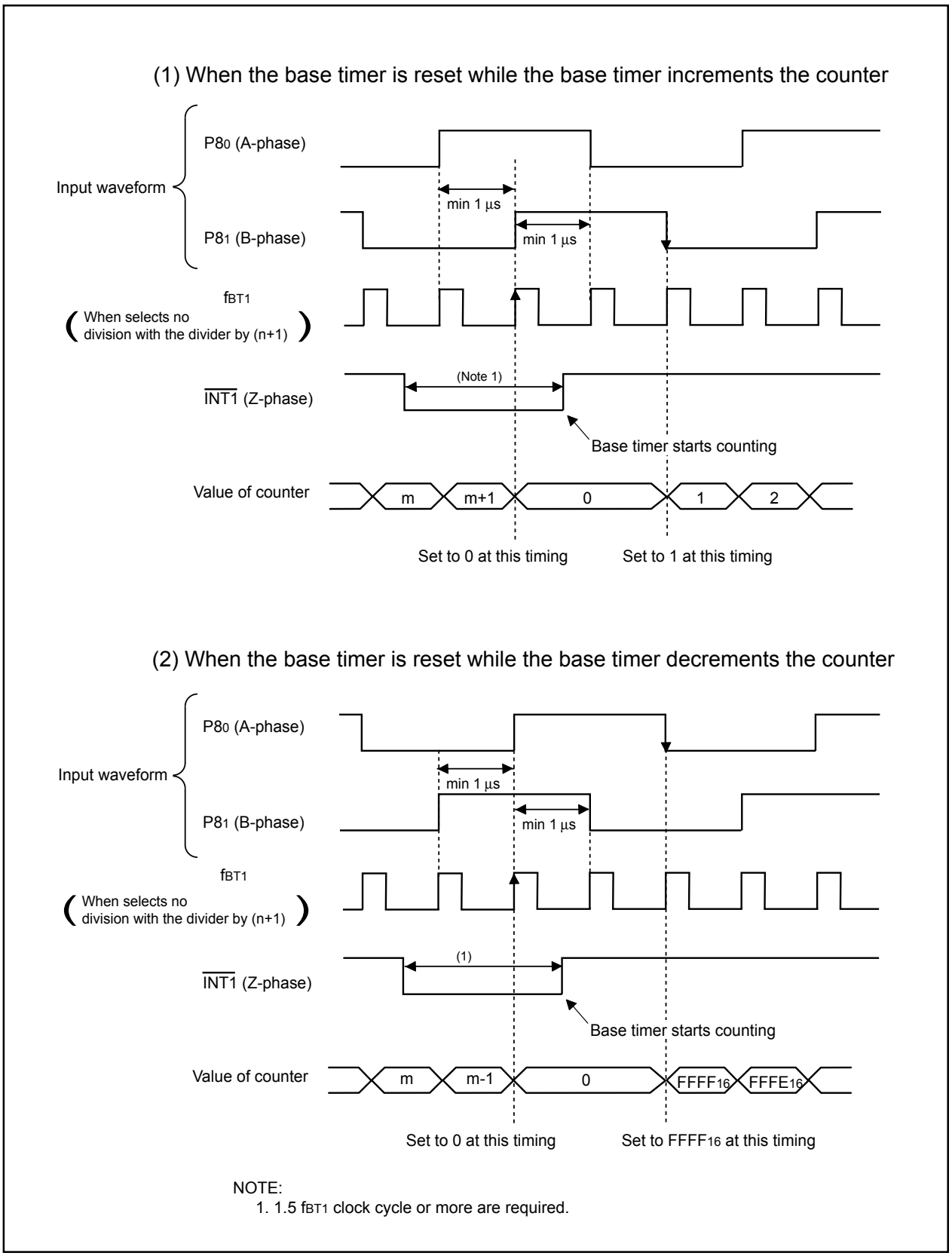


Figure 13.13 Counter Increment/Decrement Mode

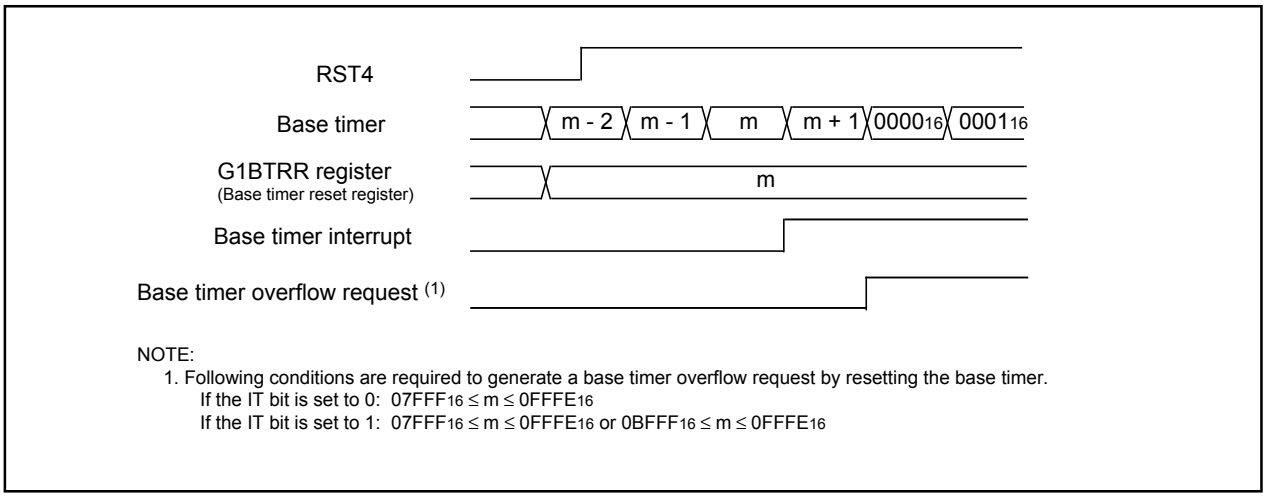


NOTE:
1. 1.5 f_{BT1} clock cycle or more are required.

Figure 13.14 Base Timer Operation in Two-phase Pulse Signal Processing Mode

13.1.1 Base Timer Reset Register(G1BTRR)

The G1BTRR register provides the capability to reset the base timer when the base timer count value matches the value stored in the G1BTRR register. The G1BTRR register is enabled by the RST4 bit in the G1BCR0 register. This function is identical in operation to the G1PO0 base timer reset that is enabled by the RST1 bit in the G1BCR0 reigster. If the free-running operation is not selected, the channel 0 can be used for a waveform generation when the base timer is reset by the G1BTRR register. Do not enable bits RST1 and RST4 simultaneously.



NOTE:
1. Following conditions are required to generate a base timer overflow request by resetting the base timer.
If the IT bit is set to 0: $07FFF_{16} \le m \le 0FFFE_{16}$
If the IT bit is set to 1: $07FFF_{16} \le m \le 0FFFE_{16}$ or $0BFFF_{16} \le m \le 0FFFE_{16}$

Figure 13.15 Base Timer Reset operation by Base Timer Reset Register

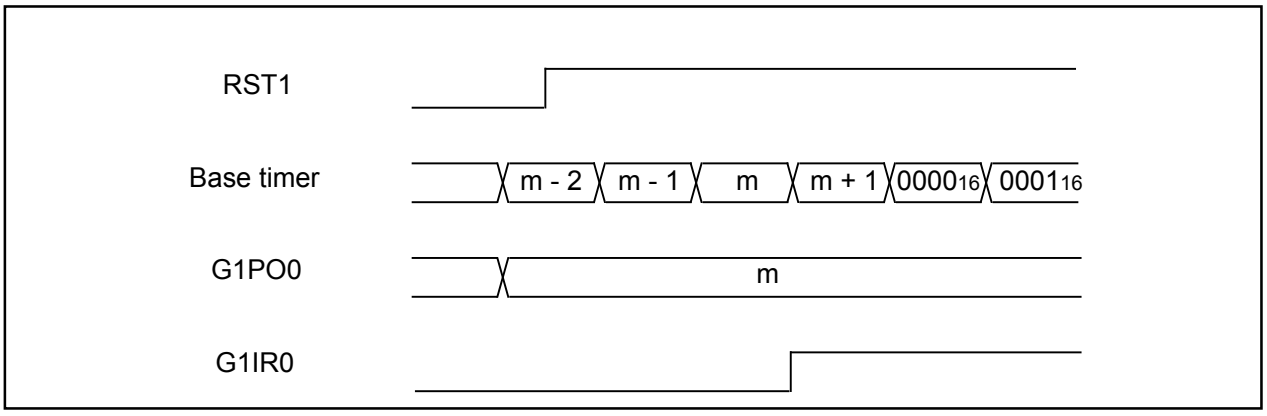
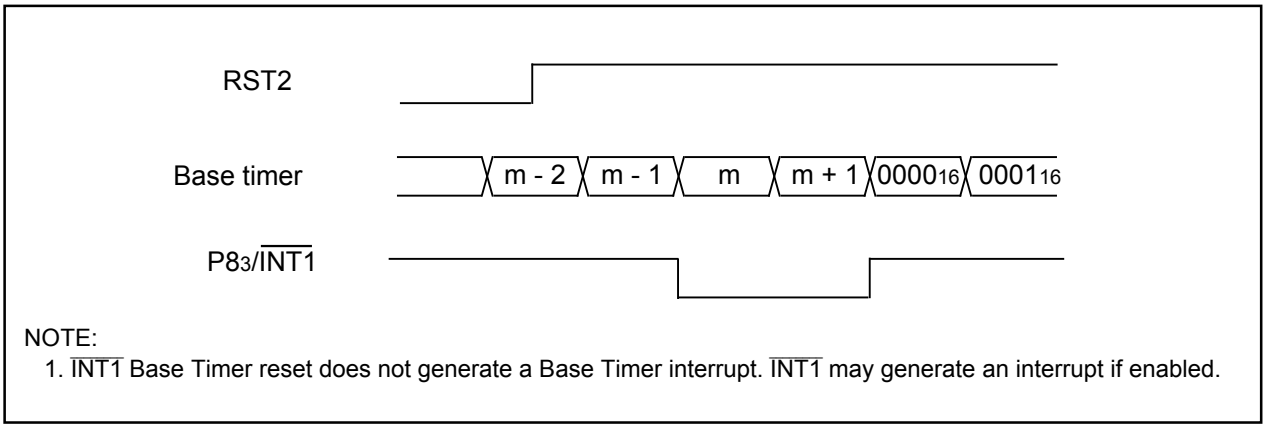


Figure 13.16 Base Timer Reset operation by G1PO0 register



NOTE:
1. $\overline{INT1}$ Base Timer reset does not generate a Base Timer interrupt. $\overline{INT1}$ may generate an interrupt if enabled.

Figure 13.17 Base Timer Reset operation by $\overline{INT1}$

13.2 Interrupt Operation

The IC/OC interrupt contains several request causes. **Figure 13.18** shows the IC/OC interrupt block diagram and **Table 13.4** shows the IC/OC interrupt assignation.

When either the base timer reset request or base timer overflow request is generated, the IR bit in the BTIC register corresponding to the IC/OC base timer interrupt is set to 1 (with an interrupt request). Also when an interrupt request in each eight channels (channel i) is generated, the bit i in the G1IR register is set to 1 (with an interrupt request). At this time, if the bit i in the G1IE0 register is 1 (IC/OC interrupt 0 request enabled), the IR bit in the ICOC0IC register corresponding to the IC/OC interrupt 0 is set to 1 (with an interrupt request). And if the bit i in the G1IE1 register is 1 (IC/OC interrupt 1 request enabled), the IR bit in the ICOC1IC register corresponding to the IC/OC interrupt 1 is set to 1(with an interrupt request).

Additionally, because each bit in the G1IR register is not automatically set to 0 even if the interrupt is acknowledged, set to 0 by program. If these bits are left as 1, all IC/OC channel interrupt causes, which are generated after setting the IR bit to 1, will be disabled.

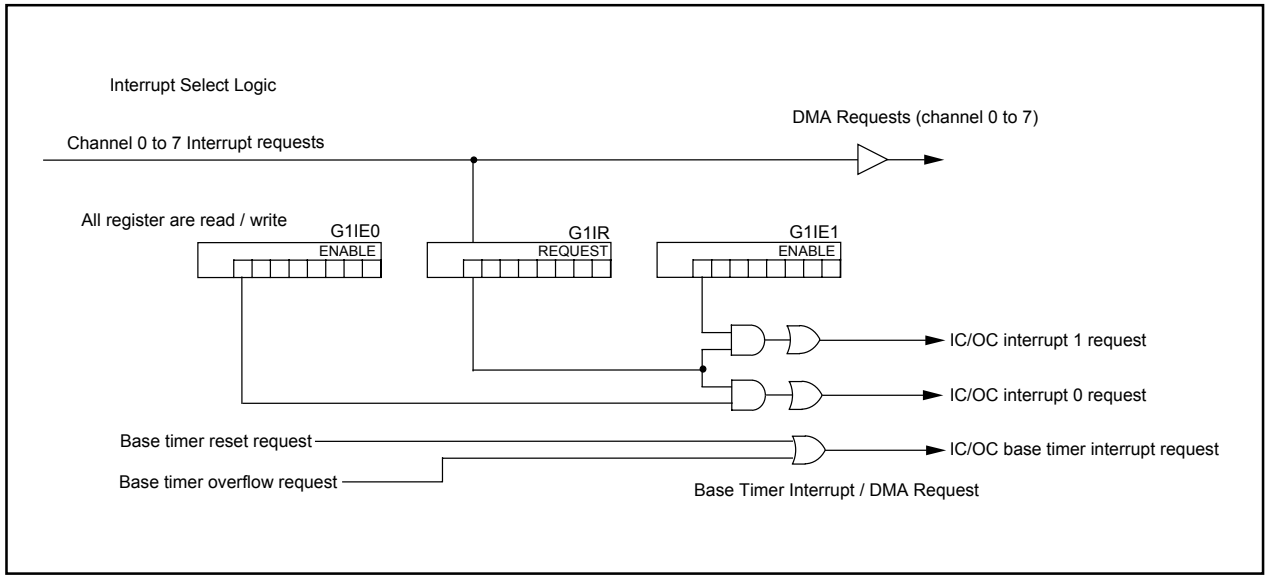


Figure 13.18 IC/OC Interrupt and DMA request generation

Table 13.4 Interrupt Assignment

Interrupt	Interrupt control register
IC/OC base timer interrupt	BTIC(0047_{16})
IC/OC interrupt 0	ICOC0IC(0045_{16})
IC/OC interrupt 1	ICOC0IC(0046_{16})

13.3 DMA Support

Each of the interrupt sources - the eight IC/OC channel interrupts and the one Base Timer interrupt - are capable of generating a DMA request.

13.4 Time Measurement Function

In synchronization with an external trigger input, the value of the base timer is stored into the G1TMj register (j=0 to 7). **Table 13.5** shows specifications of the time measurement function. **Table 13.6** shows register settings associated with the time measurement function. **Figures 13.19** and **13.20** display operational timing of the time measurement function. **Figure 13.21** shows operational timing of the prescaler function and the gate function.

Table 13.5 Time Measurement Function Specifications

Item	Specification
Measurement channel	Channels 0 to 7
Selecting trigger input polarity	Rising edge, falling edge, both edges of the INPC1j pin (1)
Measurement start condition	The IFEj bit in the G1FE register should be set to 1 (channels j function enabled) when the FSCj bit (j=0 to 7) in the G1FS register is set to 1 (time measurement function selected).
Measurement stop condition	The IFEj bit should be set to 0 (channel j function disabled)
Time measurement timing	•No prescaler : every time a trigger signal is applied •Prescaler (for channel 6 and channel 7): every *G1TPRk (k=6,7) register value +1* times a trigger signal is applied
Interrupt request generation timing	The G1IRi bit (i=0 to 7) in the interrupt request register (See **Figure 13.9**) is set to 1 at time measurement timing
INPC1j pin function (1)	Trigger input pin
Selectable function	• Digital filter function The digital filter samples a trigger input signal level every f_1, f_2 or f_{BT1} cycles and passes pulse signal matching trigger input signal level three times • Prescaler function (for channel 6 and channel 7) Time measurement is executed every *G1TPRk register value +1* times a trigger signal is applied • Gate function (for channel 6 and channel 7) After time measurement by the first trigger input, trigger input cannot be accepted. However, while the GOC bit in the G1TMCRk register is set to 1 (gate cleared by matching the base timer with the G1POp register (p=4 when k=6, p=5 when k=7)), trigger input can be accepted again by matching the base timer value with the G1POp register setting • Digital Debounce function (for channel7) See **13.6.2 Digital Debounce Function for P17/$\overline{\text{INT5}}$/INPC17** and **18.6 Digital Debounce Function** for details

NOTE:
1. The $INPC1_0$ to $INPC1_7$ pins

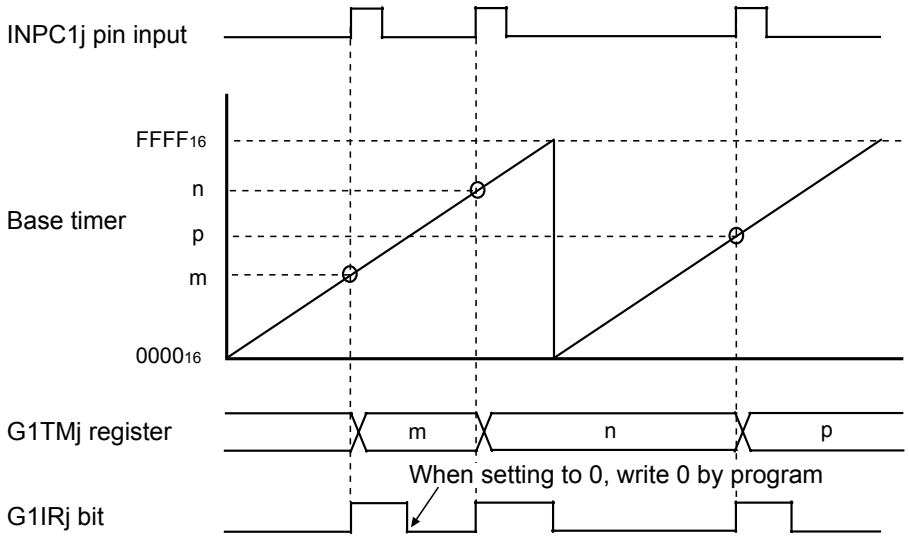
Table 13.6 Register Settings Associated with the Time Measurement Function

Register	Bit	Function
G1TMCRj	CTS1 to CTS0	Select time measurement trigger
	DF1 to DF0	Select the digital filter function
	GT, GOC, GSC	Select the gate function
	PR	Select the prescaler function
G1TPRk	-	Setting value of prescaler
G1FS	FSCj	Set to 1 (time measurement function)
G1FE	IFEj	Set to 1 (channel j function enabled)

j = 0 to 7 k = 6, 7

Bit configurations and function varys with channels used.

Registers associated with the time measurement function must be set after setting registers associated with the base timer.



j = 0 to 7
G1IRj bit: Bits in the G1IR register

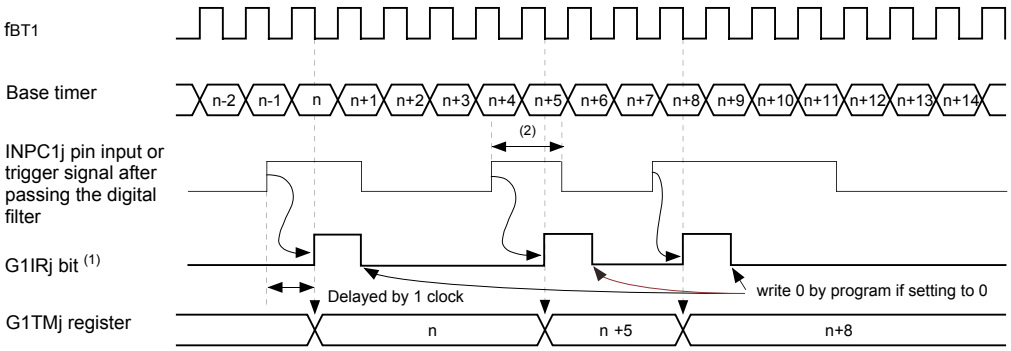
The above applies to the following condition.

- Bits CTS1 to CTS0 in the G1TMCRj registers are set to 01_2 (rising edge). The PR bit is set to 0 (no prescaler used) and the GT bit is set to 0 (no gate function used).
- Bits RTS4, RTS2, and RTS1 in registers G1BCR0 and G1BCR1 are set to 0 (no base timer reset). Bits UD1 to UD0 bits are set to 00_2 (counter increment mode).

Set the base timer to 0000_{16} (setting the RST1 bit to 1, and bits RST4 and RST2 to 0), when the base timer value matches the G1PO0 register setting. The base timer is set to 0000_{16} after it reaches *the G1PO0 register value + 2*.

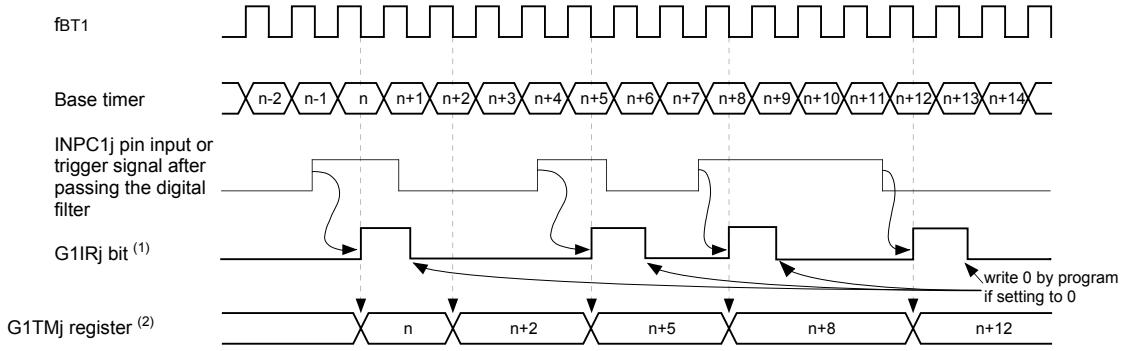
Figure 13.19 Time Measurement Function (1)

(a) When selecting the rising edge as a timer measurement trigger
(Bits CTS1 and CTS0 in the G1TMCRj register (j=0 to 7)=01$_2$)



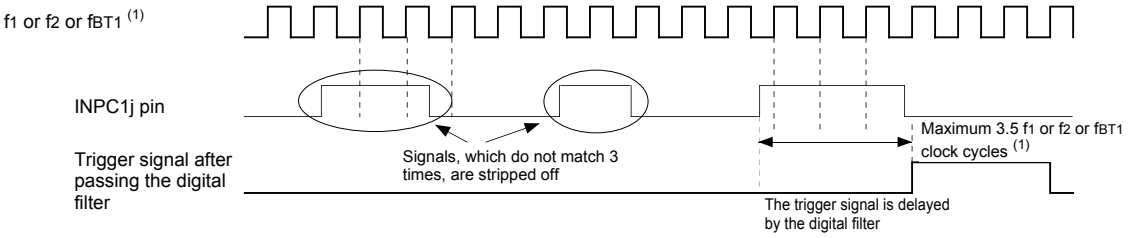
NOTES :
1. Bits in the G1IR register.
2. Input pulse applied to the INPC1j pin requires 1.5 fBT1 clock cycles or more.

(b) When selecting both edges as a timer measurement trigger
(Bits CTS1 and CTS0 = 11$_2$)



NOTES :
1. Bits in the G1IR register.
2. No interrupt is generated if the MCU receives a trigger signal when the G1IRj bit is set to 1. However, the value of the G1TMj register is updated.

(c) Trigger signal when using digital filter
(Bits DF1 to DF0 in the G1TMCRj register =10$_2$ or 11$_2$)

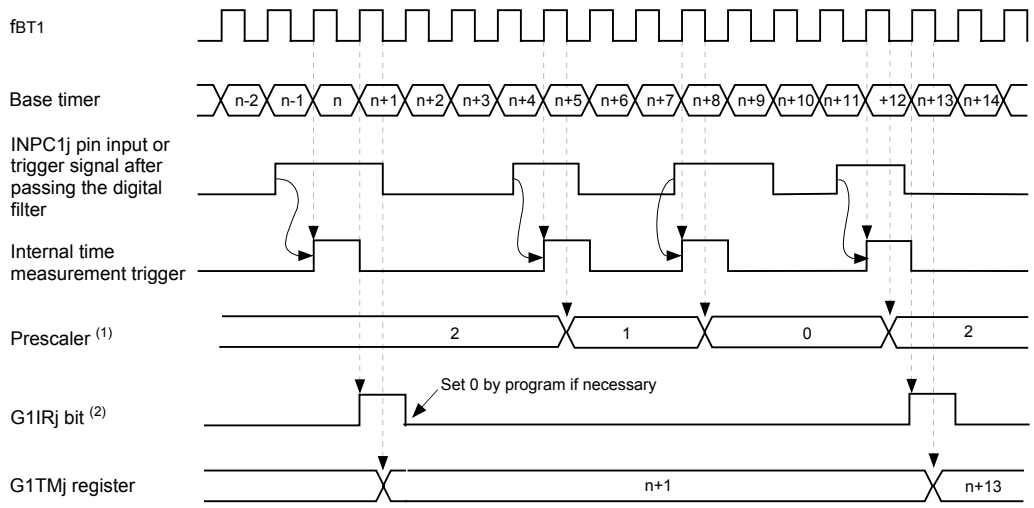


NOTE:
1. fBT1 when bits DF1 to DF0 are set to 10$_2$, and f1 or f2 when set to 11$_2$.

Figure 13.20 Time Measurement Function (2)

(a) With the prescaler function

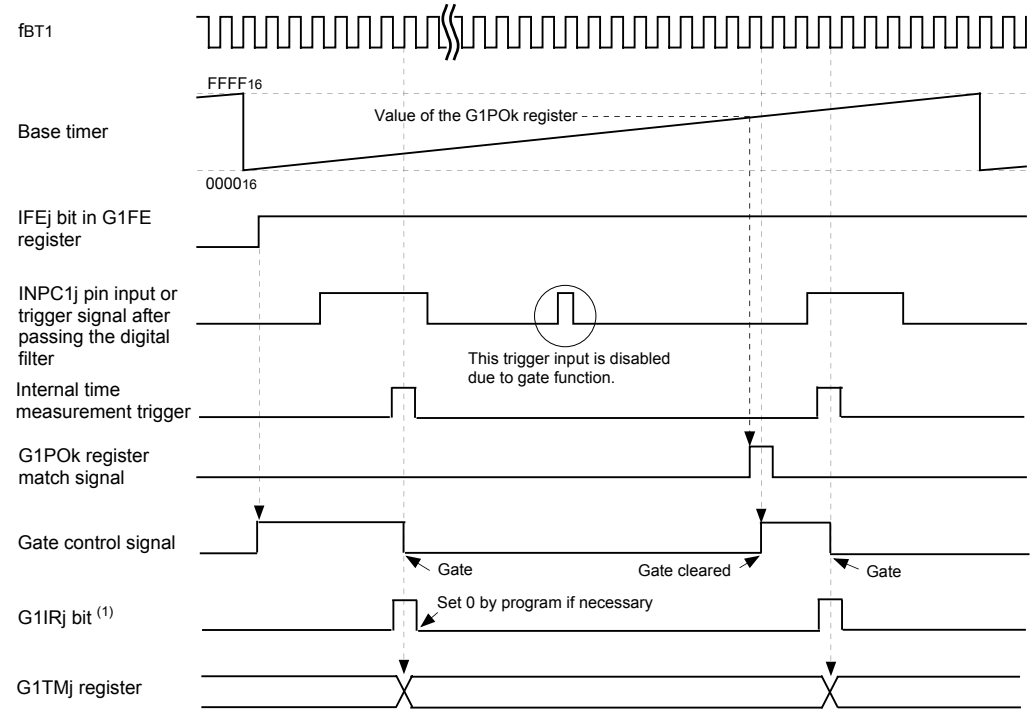
(When the G1TPRj register (j = 6, 7) is set to 02_{16}, the PR bit in the G1TMCRj (j = 6, 7) register is set to 1)



NOTES:
1. This applies to 2nd or later prescaler cycle after the PR bit in the G1TMCRj register is set to 1 (prescaler used).
2. Bits in the G1IR register.

(b) With the gate function

(The gate function is cleared by matching the base timer with the G1POk register(k=4,5), the GT bit in the G1TMCRj register is set to 1, the GOC bit is set to 1)



NOTE:
1. Bits in the G1IR register.

Figure 13.21 Prescaler Function and Gate Function

13.5 Waveform Generating Function

Waveforms are generated when the base timer value matches the G1POj (j=0 to 7) register value.

The waveform generating function has the following three modes :

- Single-phase waveform output mode
- Phase-delayed waveform output mode
- Set/Reset waveform output (SR waveform output) mode

Table 13.7 lists registers associated with the waveform generating function.

Table 13.7 Registers Related to the Waveform Generating Function Settings

Register	Bit	Function
G1POCRj	MOD1 to MOD0	Select output waveform mode
	IVL	Select default value
	RLD	Select G1POj register value reload timing
	INV	Select inverse output
G1POj	-	Select timing to output waveform inverted
G1FS	FSCj	Set to 0 (waveform generating function)
G1FE	IFEj	Set to 1 (enables function on channel j)

j = 0 to 7

Bit configurations and functions vary with channels used.

Registers associated with the waveform generating function must be set after setting registers associated with the base timer.

13.5.1 Single-Phase Waveform Output Mode

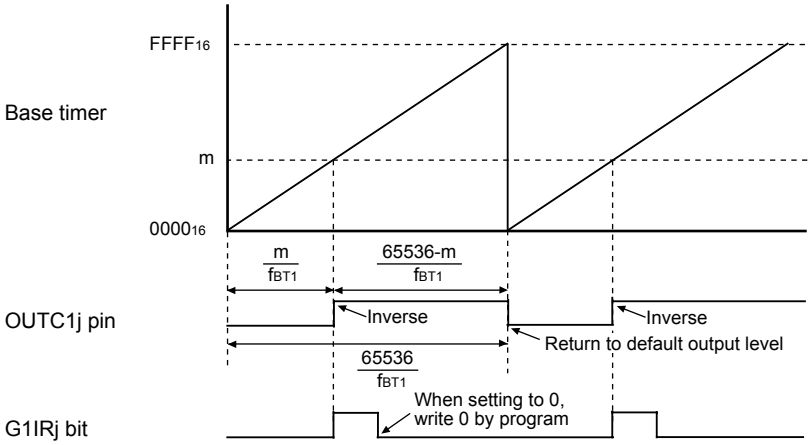
Output signal level of the OUTC1j pin becomes high ("H") when the INV bit in the G1POCRj (j=0 to 7) register is set to 0(output is not reversed) and the base timer value matches the G1POj (j=0 to 7) register value. The "H" signal switches to a low-level ("L") signal when the base timer reaches 0000_{16}. **Table 13.8** lists specifications of single-phase waveform mode. **Figure 13.22** lists an example of single-phase waveform mode operation.

Table 13.8 Single-phase Waveform Output Mode Specifications

Item	Specification
Output waveform	• Free-running operation (bits RST1, RST2, and RST4 of registers G1BCR1 and G1BCR0 are set to 0 (no reset)) Cycle : $\frac{65536}{f_{BT1}}$ Default output level width : $\frac{m}{f_{BT1}}$ Inverse level width : $\frac{65536-m}{f_{BT1}}$ • The base timer is cleared to 0000_{16} by matching the base timer with either following register (a) G1PO0 register (enabled by setting RST1 bit to 1, and RST4 and RST2 bits to 0), or (b) G1BTRR register (enabled by setting RST4 bit to 1, and RST2 and RST1 bits to 0) Cycle : $\frac{n+2}{f_{BT1}}$ Default output level width : $\frac{m}{f_{BT1}}$ Inverse level width $\frac{n+2-m}{f_{BT1}}$ m : setting value of the G1POj register (j=0 to 7), 0001_{16} to $FFFD_{16}$ n : setting value of the G1PO0 register or the G1BTRR register, 0001_{16} to $FFFD_{16}$
Waveform output start condition	The IFEj bit in the G1FE register is set to 1 (channel j function enabled)
Waveform output stop condition	The IFEj bit is set to 0 (channel j function disabled)
Interrupt request	The G1IRj bit in the G1IR register is set to 1 when the base timer value matches the G1POj register value (See **Figure 13.22**)
OUTC1j pin (1)	Pulse signal output pin
Selectable function	• Default value set function: Set starting waveform output level • Inverse output function: Waveform output signal is inversed and provided from the OUTC1j pin

NOTE:
1. Pins $OUTC1_0$ to $OUTC1_7$.

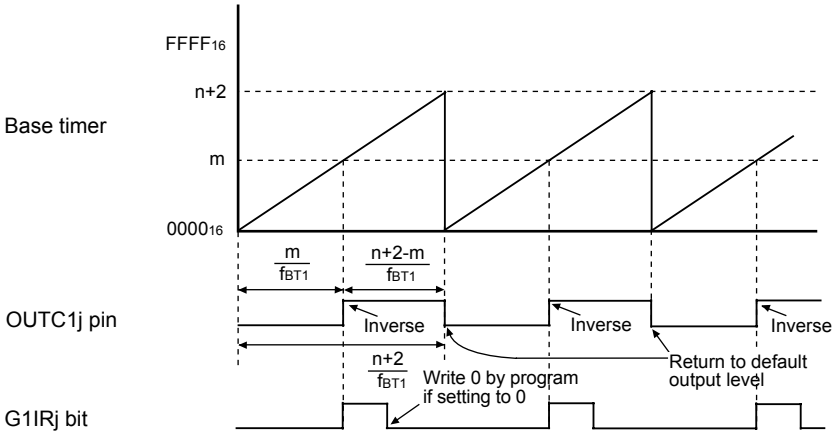
(1) Free-running operation
(The RST4, RST2, and RST1 bits in the G1BCR0 and G1BCR1 registers are set to 0)



j=0 to 7
m : Setting value of the G1POj register
G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.
-The IVL bit in the G1POCRj register is set to 0 ("L" output as a default value) and the INV bit is set to 0 (not inversed).
-Bits UD1 to UD0 are set to 00_2 (counter increment mode).

(2) The base timer is reset when the base timer matches either following register
(a) G1PO0 (enabled by setting bit RST1 to 1, and bits RST4 and RST2 to 0), or
(b) G1BTRR (enabled by setting bit RST4 to 1, and bits RST2 and RST1 to 0)



j=1 to 7
m : Setting value of the G1POj register
n: Setting value of either G1PO0 register or G1BTRR register
G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.
-The IVL bit in the G1POCRj register is set to 0 ("L" output as a default value) and the INV bit is set to 0 (not inversed).
-Bits UD1 to UD0 are set to 00_2 (counter increment mode).

Figure 13.22 Single-phase Waveform Output Mode

13.5.2 Phase-Delayed Waveform Output Mode

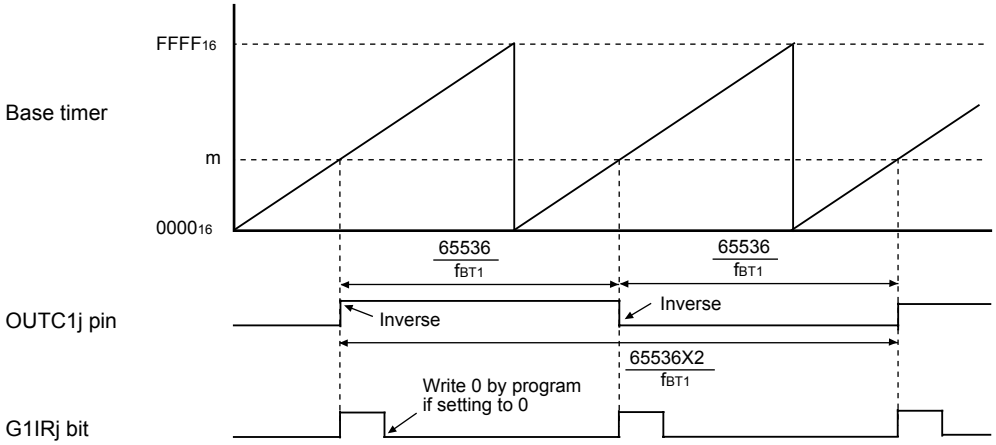
Output signal level of the OUTC1j pin is inversed every time the base timer value matches the G1POj register value (j=0 to 7). **Table 13.9** lists specifications of phase-delayed waveform mode. **Figure 13.23** shows an example of phase-delayed waveform mode operation.

Table 13.9 Phase-delayed Waveform Output Mode Specifications

Item	Specification
Output waveform	• Free-running operation (bits RST1, RST2, and RST4 in registers G1BCR1 and G1BCR0 are set to 0 (no reset)) Cycle : $\frac{65536 \times 2}{f_{BT1}}$ "H" and "L" width : $\frac{65536}{f_{BT1}}$ • The base timer is cleared to 0000_{16} by matching the base timer with either following register (a) G1PO0 register (enabled by setting RST1 bit to 1, and bits RST4 and RST2 to 0), or (b) G1BTRR register (enabled by setting RST4 bit to 1, and bits RST2 and RST1 to 0) Cycle : $\frac{2(n+2)}{f_{BT1}}$ "H" and "L" width : $\frac{n+2}{f_{BT1}}$ n : setting value of either G1PO0 register or G1BTRR register
Waveform output start condition	The IFEj bit in the G1FE register is set to 1 (channel j function enabled)
Waveform output stop condition	The IFEj bit is set to 0 (channel j function disabled)
Interrupt request	The G1IRj bit in the interrupt request register is set to 1 when the base timer value matches the G1POj register value. (See **Figure 13.23**)
OUTC1j pin (1)	Pulse signal output pin
Selectable function	• Default value set function: Set starting waveform output level • Inverse output function : Waveform output signal is inversed and provided from the OUTC1j pin

NOTE:
1. Pins $OUTC1_0$ to $OUTC1_7$.

(1) Free-running operation
(Bits RST4, RST2, and RST1 in the registers G1BCR0 and G1BCR1 are set to 0)

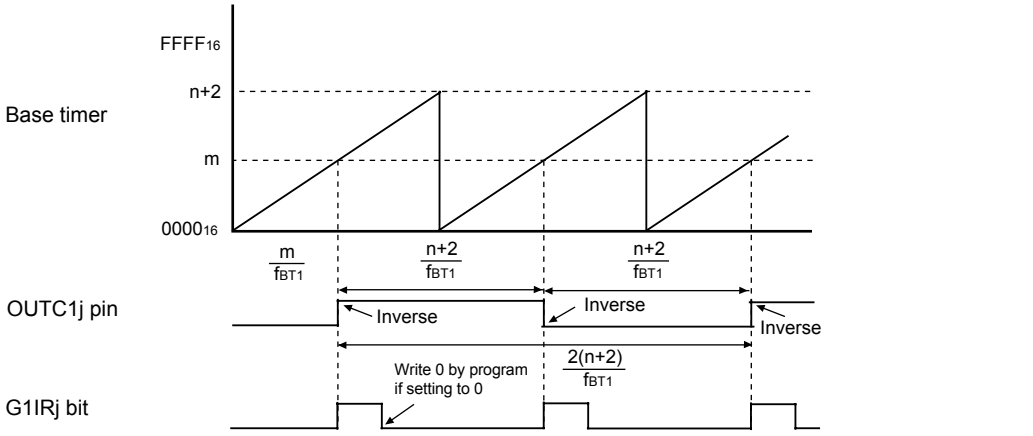


j=0 to 7
m : Setting value of the G1POj register
G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inversed).
- Bits UD1 to UD0 are set to 00_2 (counter increment mode).

(2) Base timer is reset when the base timer matches either following register
(a) G1PO0 (enabled by setting bit RST1 to 1, and bits RST4 and RST2 to 0), or
(b) G1BTRR (enabled by setting bit RST4 to 1, and bits RST2 and RST1 to 0)



j=1 to 7
m : Setting value of the G1POj register n: Setting value of either register G1PO0 or G1BTRR
G1IRj bit : Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inversed).
- Bits UD1 to UD0 are set to 00_2 (counter increment mode).

Figure 13.23 Phase-delayed Waveform Output Mode

13.5.3 Set/Reset Waveform Output (SR Waveform Output) Mode

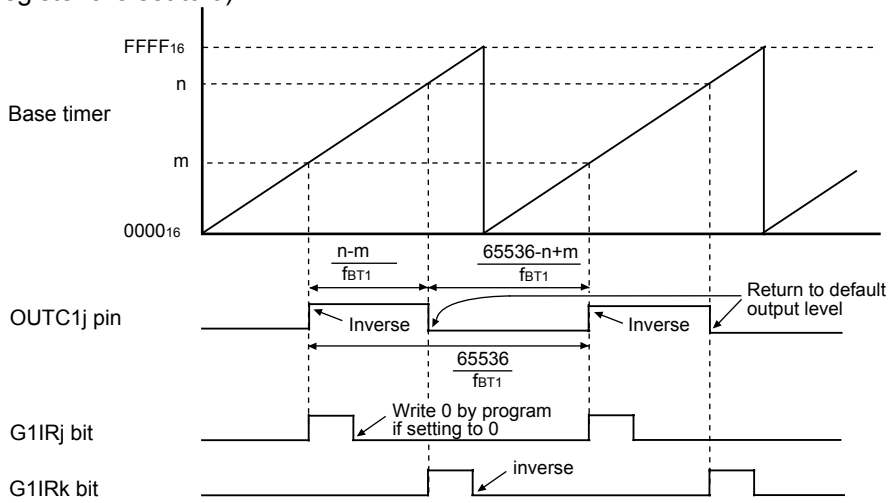
Output signal level of the OUTC1j pin becomes high ("H") when the INV bit in the G1POCRi (i=0 to 7) is set to 0 (output is not reversed) and the base timer value matches the G1POj register value (j=0, 2, 4, 6). The "H" signal switches to a low-level ("L") signal when the base timer value matches the G1POk (k=j+1) register value. **Table 13.10** lists specifications of SR waveform mode. **Figure 13.24** shows an example of the SR waveform mode operation.

Table 13.10 SR Waveform Output Mode Specifications

Item	Specification
Output waveform	• Free-running operation (the RST1, RTS2, and RST4 bits of the G1BCR1 and G1BCR0 registers are set to 0 (no reset)) Cycle : $\frac{65536}{f_{BT1}}$ Inverse level width(1) : $\frac{n-m}{f_{BT1}}$ • The base timer is cleared to 0000_{16} by matching the base timer with either following register (a) G1PO0 register (enabled by setting RST1 bit to 1, and bits RST4 and RST2 to 0)(2), or (b) G1BTRR register (enabled by setting RST4 bit to 1, and bits RST2 and RST1 to 0) Cycle : $\frac{p+2}{f_{BT1}}$ Inverse level width(1) : $\frac{n-m}{f_{BT1}}$ m : setting value of the G1POj register (j=0, 2, 4, 6) n : setting value of the G1POk register (k=j+1) p : setting value of the G1PO0 register or G1BTRR register value range of m, n, p: 0001_{16} to $FFFD_{16}$
Waveform output start condition	Bits IFEj and IFEk in the G1FE register are set to 1 (channel j function enabled)
Waveform output stop condition	Bits IFEj and IFEk are set to 0 (channel j function disabled)
Interrupt request	The G1IRj bit in the G1IR register is set to 1 when the base timer value matches the G1POj register value. The G1IRk bit in the interrupt request register is set to 1 when the base timer value matches the G1POk register value (See **Figure 13.24**)
OUTC1j pin (3)	Pulse signal output pin
Selectable function	• Default value set function : Set starting waveform output level • Inverse output function: Waveform output signal is inversed and provided from the OUTC1j pin

NOTES:
1. The odd channel's waveform generating register must have greater value than the even channel's.
2. When the G1PO0 register resets the base timer, the channel 0 and channel 1 SR waveform generating functions are not available.
3. Pins $OUTC1_0$, $OUTC1_2$, OUTC14, $OUTC1_6$.

(1) Free-running operation
(Bits RST2 and RST1 in the G1BCR0 register and the RST4 bit in the G1BCR1 register are set to 0)

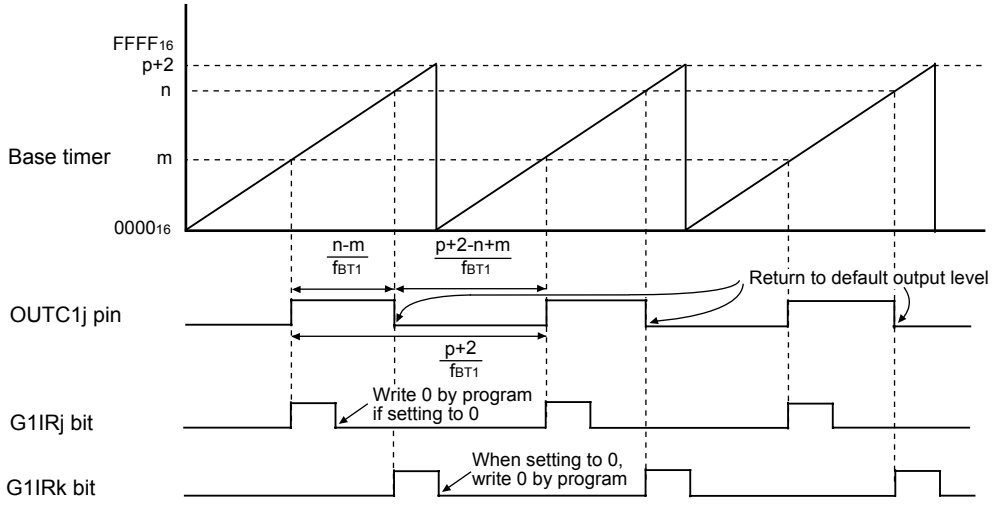


j=0, 2, 4, 6 k=j+1
m : Setting value of the G1POj register n: Setting value of the G1POk register
G1IRj, G1IRk bits: Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inversed).
- Bits UD1 and UD0 are set to 00_2 (counter increment mode).

(2) Base timer is reset when the base timer matches either following register
(a) G1PO0 (enabled by setting bit RST1 to 1, and bits RST4 and RST2 to 0), or
(b) G1BTRR (enabled by setting bit RST4 to 1, and bits RST2 and RST1 to 0)



j=2, 4, 6 k=j+1
m : Setting value of the G1POj register n: Setting value of the G1POk register
p: Setting value of either register G1PO0 or G1BTRR
G1IRj, G1IRk bits: Bits in the G1IR register

The above applies under the following conditions.

- The IVL bit in the G1POCRj register is set to 0 (L output as a default value). The INV bit is set to 0 (not inversed).
- Bits UD1 and UD0 are set to 00_2 (counter increment mode).

Figure 13.24 Set/Reset Waveform Output Mode

13.6 I/O Port Function Select

The value in the G1FE and G1FS registers decides which IC/OC pin to be an input or output pin.

In SR waveform generating mode, two channels, a set of even channel and odd channel, are used every output waveform, however, the waveform is output from an even channel only. In this case, the corresponding pin to the odd channel can be used as an I/O port.

Table 13.11 Pin setting for Time Measurement and Waveform Generating Functions

Pin	IFE	FSC	MOD1	MOD0	Port Direction	Port Data
$P2_7$/$INPC1_7$/ $OUTC1_7$	0	X	X	X	Determined by $PD2_7$	$P2_7$
	1	1	X	X	Determined by $PD2_7$, Input to $INPC1_7$ is always active	$P2_7$ or $INPC1_7$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_7$
	1	0	0	1	Determined by $PD2_7$, SR Waveform Output mode	$P2_7$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_7$
$P2_6$/$INPC1_6$/ $OUTC1_6$	0	X	X	X	Determined by $PD2_6$	$P2_6$
	1	1	X	X	Determined by $PD2_6$, Input to $INPC1_6$ is always active	$P2_6$ or $INPC1_6$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_6$
	1	0	0	1	SR Waveform Output	$OUTC1_6$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_6$
$P2_5$/$INPC1_5$/ $OUTC1_5$	0	X	X	X	Determined by $PD2_5$	$P2_5$
	1	1	X	X	Determined by $PD2_5$, Input to $INPC1_5$ is always active	$P2_5$ or $INPC1_5$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_5$
	1	0	0	1	Determined by $PD2_5$, SR Waveform Output mode	$P2_5$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_5$
$P2_4$/$INPC1_4$/ $OUTC1_4$	0	X	X	X	Determined by $PD2_4$	$P2_4$
	1	1	X	X	Determined by $PD2_4$, Input to $INPC1_4$ is always active	$P2_4$ or $INPC1_4$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_4$
	1	0	0	1	SR Waveform Output	$OUTC1_4$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_4$
$P2_3$/$INPC1_3$/ $OUTC1_3$	0	X	X	X	Determined by $PD2_3$	$P2_3$
	1	1	X	X	Determined by $PD2_3$, Input to $INPC1_3$ is always active	$P2_3$ or $INPC1_3$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_3$
	1	0	0	1	Determined by $PD2_3$, SR Waveform Output mode	$P2_3$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_3$
$P2_2$/$INPC1_2$/ $OUTC1_2$	0	X	X	X	Determined by $PD2_2$	$P2_2$
	1	1	X	X	Determined by $PD2_2$, Input to $INPC1_2$ is always active	$P2_2$ or $INPC1_2$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_2$
	1	0	0	1	SR Waveform Output	$OUTC1_2$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_2$
$P2_1$/$INPC1_1$/ $OUTC1_1$	0	X	X	X	Determined by $PD2_1$	$P2_1$
	1	1	X	X	Determined by $PD2_1$, Input to $INPC1_1$ is always active	$P2_1$ or $INPC1_1$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_1$
	1	0	0	1	Determined by $PD2_1$, SR Waveform Output mode	$P2_1$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_1$
$P2_0$/$INPC1_0$/ $OUTC1_0$	0	X	X	X	Determined by $PD2_0$	$P2_0$
	1	1	X	X	Determined by $PD2_0$, Input to $INPC1_0$ is always active	$P2_0$ or $INPC1_0$
	1	0	0	0	Single-phase Waveform Output	$OUTC1_0$
	1	0	0	1	SR Waveform Output	$OUTC1_0$
	1	0	1	0	Phase-delayed Waveform Output	$OUTC1_0$

IFE: IFEj (j=0 to 7) bits in the G1FE register.

FSC: FSCj (j=0 to 7) bits in the G1FS register.

MOD2 to MOD1: Bits in the G1POCRj (j=0 to 7) register.

13.6.1 INPC17 Alternate Input Pin Selection

The input capture pin for IC/OC channel 7 can be assigned to one of two package pins. The CH7INSEL bit in the G1BCR0 register selects IC/OC INPC17 from P27/OUTC17/INPC17 or P17/$\overline{\text{INT5}}$/INPC17/IDU.

13.6.2 Digital Debounce Function for Pin P17/$\overline{\text{INT5}}$/INPC17

The $\overline{\text{INT5}}$/INPC17 input from the P17/$\overline{\text{INT5}}$/INPC17/IDU pin has an effective digital debounce function against a noise rejection. Refer to **18.6 Digital Debounce function** for this detail.

14. Serial I/O

Note

The SI/O4 interrupt of peripheral function interrupt is not available in the 64-pin package.

Serial I/O is configured with five channels: UART0 to UART2, SI/O3 and SI/O4.

14.1 UARTi (i=0 to 2)

UARTi each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 14.1 shows the block diagram of UARTi. **Figures 14.2** and **14.3** shows the block diagram of the UARTi transmit/receive.

UARTi has the following modes:

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

Figures 14.4 to **14.9** show the UARTi-associated registers.

Refer to **Tables 14.2, 14.6, 14.11, 14.12, 14.16, 14.17**, and **14.19** to set the registers in individual mode.

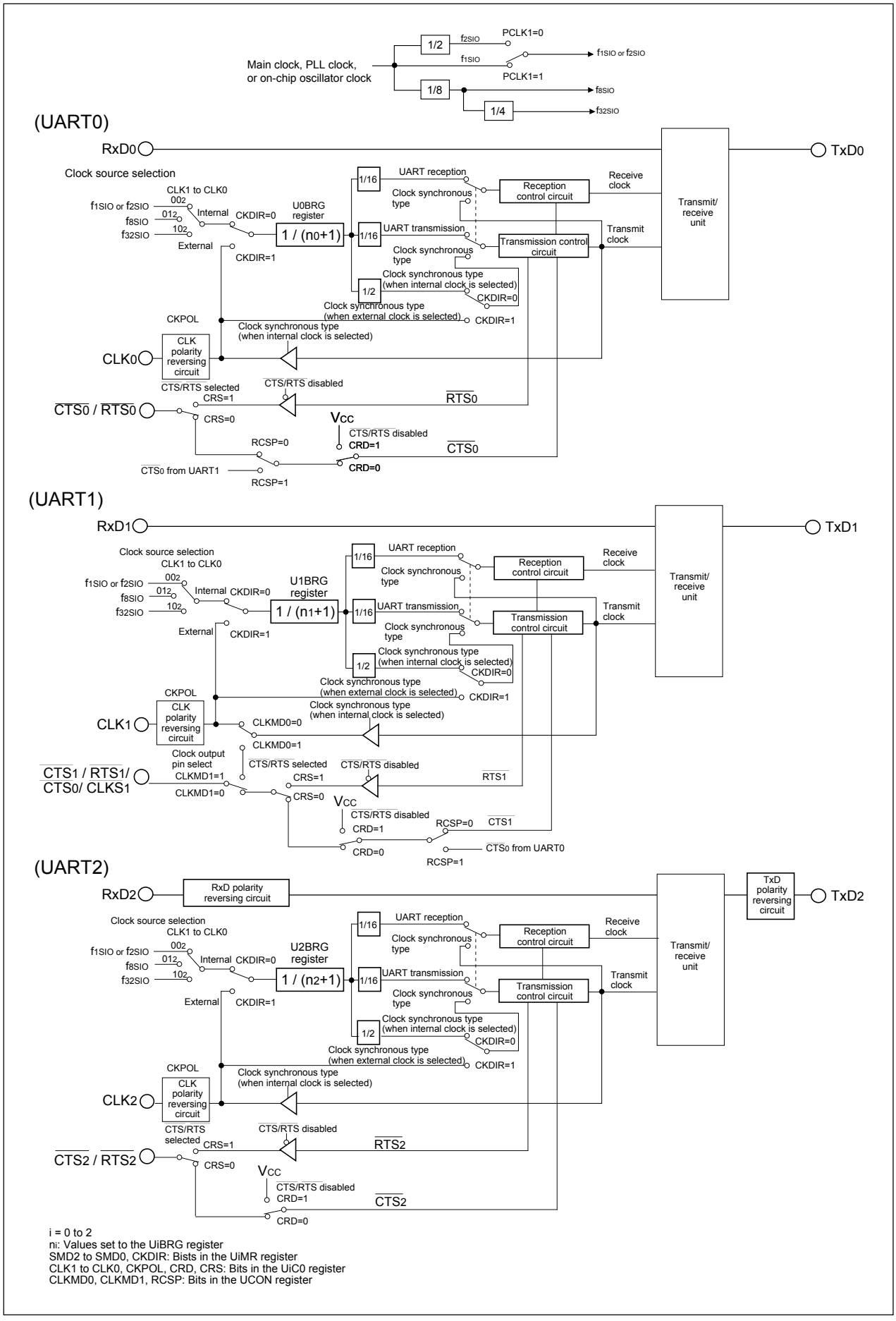


Figure 14.1 Block diagram of UARTi (i = 0 to 2)

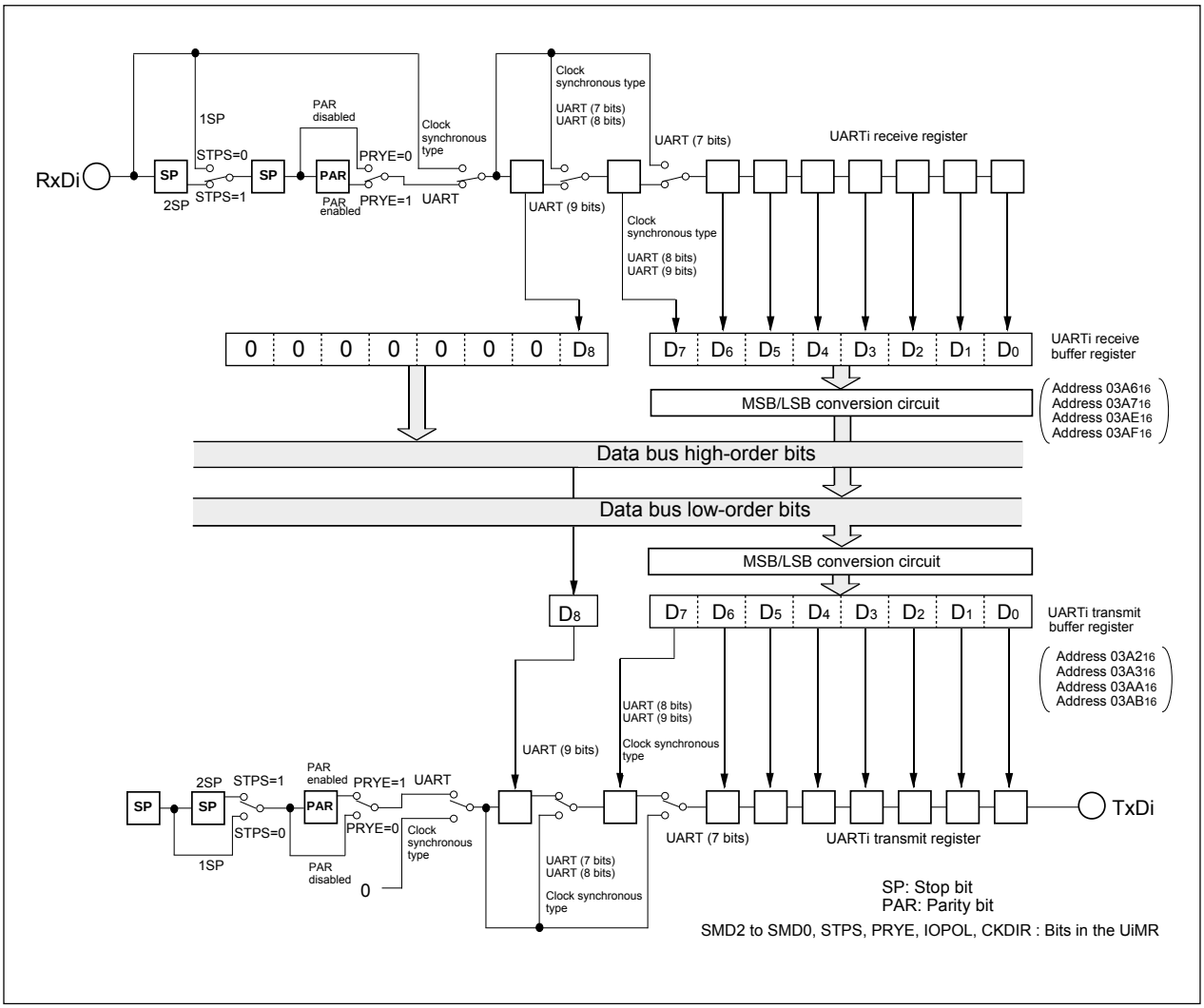


Figure 14.2 Block diagram of UARTi (i = 0, 1) transmit/receive unit

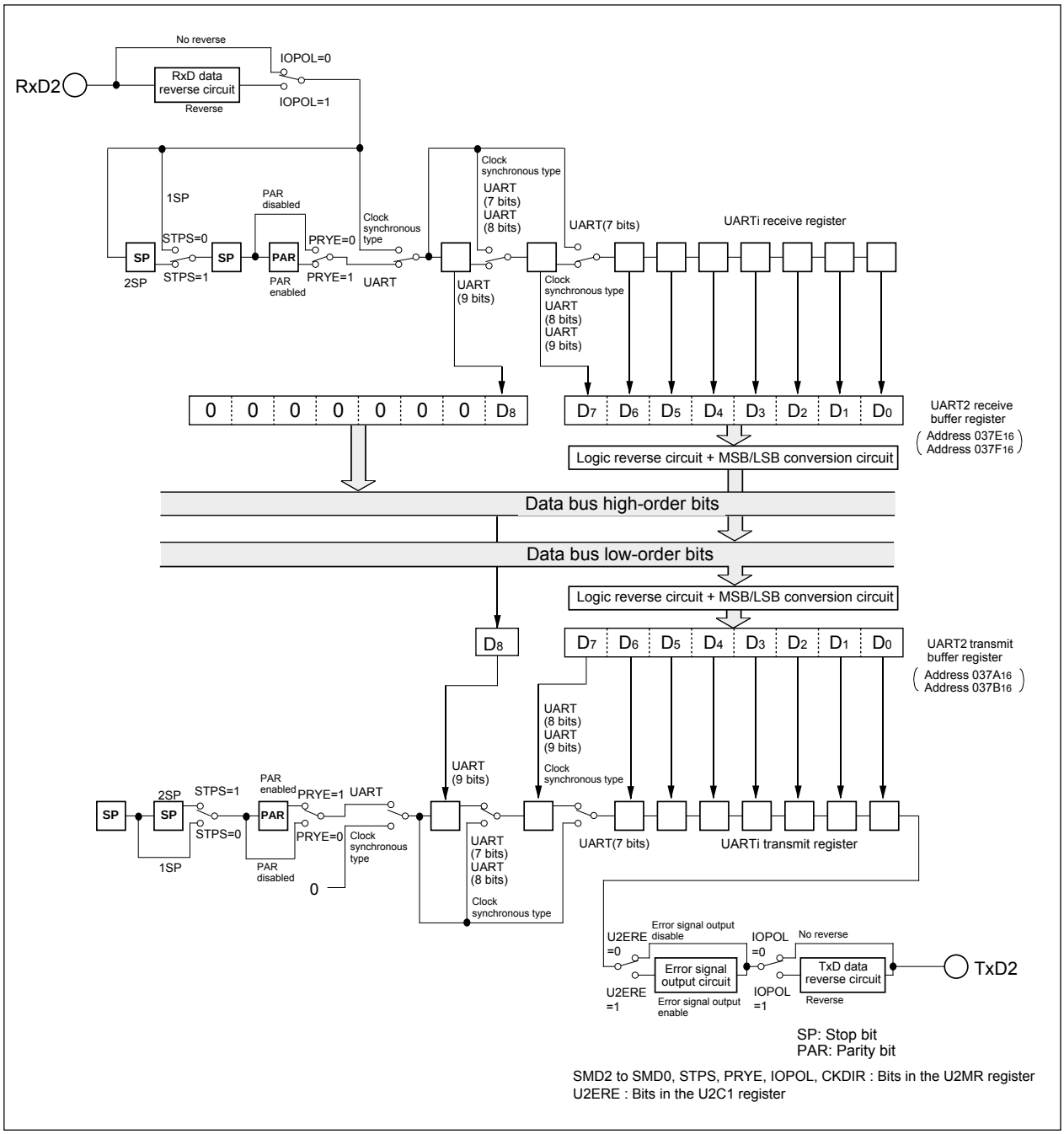
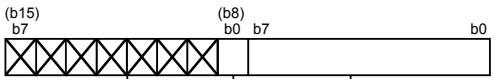


Figure 14.3 Block diagram of UART2 transmit/receive unit

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

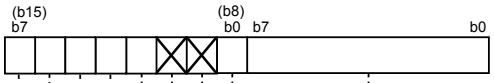


Symbol	Address	After Reset
U0TB	03A3$_{16}$-03A2$_{16}$	Undefined
U1TB	03AB$_{16}$-03AA$_{16}$	Undefined
U2TB	037B$_{16}$-037A$_{16}$	Undefined

Function	RW
Transmit data	WO
Nothing is assigned. If necessary, set to 0. When read, their contents are undefined	—

NOTES:
1. Use MOV instruction to write to this register.

UARTi Receive Buffer Register (i=0 to 2)

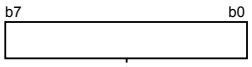


Symbol	Address	After Reset
U0RB	03A7$_{16}$-03A6$_{16}$	undefined
U1RB	03AF$_{16}$-03AE$_{16}$	undefined
U2RB	037F$_{16}$-037E$_{16}$	undefined

Bit Symbol	Bit Name	Function	RW
(b7-b0)	———	Receive data (D$_7$ to D$_0$)	RO
(b8)	———	Receive data (D$_8$)	RO
(b10-b9)	Nothing is assigned. If necessary, set to 0. When read, their contents are undefined		—
ABT	Arbitration lost detecting flag (2)	0 : Not detected 1 : Detected	RW
OER	Overrun error flag(1)	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag(1)	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag(1)	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag (1)	0 : No error 1 : Error found	RO

NOTES:
1. When the SMD2 to SMD0 bits in the UiMR register are set to 000$_2$ (serial I/O disabled) or the RE bit in the UiC1 register is set to 0 (reception disabled), all bits SUM, PER, FER and OER are set to 0 (no error). The SUM bit is set to 0 (no error) when all of the PER, FER and OER bits are set to 0 (no error). Also, bits PER and FER are set to 0 by reading the lower byte of the UiRB register.
2. The ABT bit is set to 0 by setting to 0 by program. (Writing 1 has no effect.) Nothing is assigned at the bit 11 in the U0RB and U1RB registers. If necessary, set to 0. When read, its content is 0.

UARTi Baud Rate Generation Register (i=0 to 2)(1, 2, 3)



Symbol	Address	After Reset
U0BRG	03A1$_{16}$	Undefined
U1BRG	03A9$_{16}$	Undefined
U2BRG	0379$_{16}$	Undefined

Function	Setting Range	RW
Assuming that set value = n, UiBRG divides the count source by n + 1	00$_{16}$ to FF$_{16}$	WO

NOTES:
1. Write to this register while serial I/O is neither transmitting nor receiving.
2. Use MOV instruction to write to this register.
   The transfer clock is shown below when the setting value in the UiBRG register is set as n.
   (1) When the CKDIR bit in the UiMR register to 0 (internal clock)
   - Clock synchronous serial I/O mode : fj/(2(n+1))
   - Clock asynchronous serial I/O (UART) mode : fj/(16(n+1))

   (2) When the CKDIR bit in the UiMR register to 1 (external clock)
   - Clock synchronous serial I/O mode : f$_{EXT}$
   - Clock asynchronous serial I/O (UART) mode : f$_{EXT}$/(16(n+1))

   fj : f1SIO, f2SIO, f8SIO, f32SIO
   f$_{EXT}$: Input from CLKi pin
3. Set the UiBRG register after setting bits CLK1 and CLK0 in the registers UiC0.

Figure 14.4 U0TB to U2TB, U0RB to U2RB, U0BRG to U2BRG Registers

UARTi Transmit/receive Mode Register (i=0, 1)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	After Reset
U0MR, U1MR	$03A0_{16}$, $03A8_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SMD0	Serial I/O mode select bit (2)	b2 b1 b0 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 1 0 0 : UART mode transfer data 7 bit long 1 0 1 : UART mode transfer data 8 bit long 1 1 0 : UART mode transfer data 9 bit long Do not set the value other than the above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (1)	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	Effective when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
$\overline{\text{(b7)}}$	Reserve bit	Set to 0	RW

NOTES:
1. Set the corresponding port direction bit for each CLKi pin to 0 (input mode).
2. To receive data, set the corresponding port direction bit for each RxDi pin to 0.

UART2 Transmit/receive Mode Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
U2MR	0378_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
SMD0	Serial I/O mode select bit (2)	b2 b1 b0 0 0 0 : Serial I/O disabled 0 0 1 : Clock synchronous serial I/O mode 0 1 0 : I^2C bus mode(3) 1 0 0 : UART mode transfer data 7 bit long 1 0 1 : UART mode transfer data 8 bit long 1 1 0 : UART mode transfer data 9 bits long Do not set the value other than the above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (1)	RW
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	RW
PRY	Odd/even parity select bit	Effective when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
IOPOL	TxD, RxD I/O polarity reverse bit	0 : No reverse 1 : Reverse	RW

NOTES:
1. Set the corresponding port direction bit for each CLK2 pin to 0 (input mode).
2. To receive data, set the corresponding port direction bit for each RxD2 pin to 0 (input mode).
3. Set the corresponding port direction bit for SCL_2 and SDA_2 pins to 0 (input mode).

Figure 14.5 U0MR to U2MR Registers

UARTi Transmit/receive Control Rregister 0 (i=0 to 2)

Symbol	Address	After Reset
U0C0 to U2C0	$03A4_{16}$, $03AC_{16}$, $037C_{16}$	00001000_2

Bit Symbol	Bit Name	Function	RW
CLK0	BRG count source select bit(7)	b1 b0 0 0 : $f1_{SIO}$ or $f2_{SIO}$ is selected 0 1 : $f8_{SIO}$ is selected 1 0 : $f32_{SIO}$ is selected 1 1 : Do not set	RW
CLK1			RW
CRS	$\overline{CTS}/\overline{RTS}$ function select bit (3)	Effective when CRD is set to 0 0 : $\overline{CTS}$ function is selected (1) 1 : RTS function is selected	RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : $\overline{CTS}/\overline{RTS}$ function disabled ($P6_0$, $P6_4$ and $P7_3$ can be used as I/O ports)(6)	RW
NCH	Data output select bit(5)	0 : TxD2/SDA2 and SCLi pins are CMOS output 1 : TxD2/SDA2 and SCLi pins are N-channel open-drain output(4)	RW
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	RW
UFORM	Transfer format select bit (2)	0 : LSB first 1 : MSB first	RW

NOTES:
1. Set the corresponding port direction bit for each $\overline{CTSi}$ pin to 0 (input mode).
2. Effective when bits SMD2 to SMD0 in the UMR register to 001_2 (clock synchronous serial I/O mode) or 010_2 (UART mode transfer data 8 bits long). Set the UFORM bit to 1 when bits SMD2 to SMD0 are set to 101_2 (I^2C bus mode) and 0 when they are set to 100_2.
3. $\overline{CTS1}/\overline{RTS1}$ can be used when the CLKMD1 bit in the UCON register is set to 0 (only CLK_1 output) and the RCSP bit in the UCON register is set to 0 ($\overline{CTS0}/\overline{RTS0}$ not separated).
4. SDA2 and SCL2 are effective when i = 2.
5. When bits SMD2 to SMD in the UiMR regiser are set to 000_2 (serial I/O disable), do not set NCH bit to 1 (TxDi/SDA2 and SCL2 pins are N-channel open-drain output).
6. When the U1MAP bit in PACR register is 1 ($P7_3$ to $P7_0$), $P7_0$ functions as $\overline{CTS}/\overline{RTS}$ pin in UART1.
7. When the CLK1 and CLK0 bit settings are changed, set the UiBRG register.

UART Transmit/receive Control Register 2

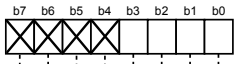
Symbol	Address	After Reset
UCON	$03B0_{16}$	$X0000000_2$

Bit Symbol	Bit Name	Function	RW
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U1IRS	UART1 transmit interrupt cause select	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	RW
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	RW
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
CLKMD0	UART1 CLK/CLKS select bit 0	Effective when the CLKMD1 bit is set to 1 0 : Clock output from CLK1 1 : Clock output from CLKS1	RW
CLKMD1	UART1 CLK/CLKS select bit 1 (1)	0 : Output from CLK1 only 1 : Transfer clock output from multiple pins function selected	RW
RCSP	Separate UART0 $\overline{CTS}/\overline{RTS}$ bit	0 : $\overline{CTS}/\overline{RTS}$ shared pin (2) 1 : $\overline{CTS}/\overline{RTS}$ separated ($P6_4$ pin functions as $\overline{CTS0}$ pin)	RW
$\overline{(b7)}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—

NOTES:
1. When using multiple transfer clock output pins, make sure the following conditions are met:set the CKDIR bit in the U1MR register to 0 (internal clock)
2. When the U1MAP bit in PACR register is set to 1 ($P7_3$ to $P7_0$), $P7_0$ pin functions as $\overline{CTS0}$ pin.

Figure 14.6 U0C0 to U2C0 and UCON Registers

UARTi Transmit/receive Control Register 1 (i=0, 1)



Symbol: U0C1, U1C1 | Address: 03A5$_{16}$,03AD$_{16}$ | After Reset: 00000010$_2$

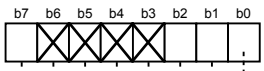
Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in UiTB register 1 : No data present in UiTB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in UiRB register 1 : Data present in UiRB register	RO
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

UART2 Transmit/receive Control Register 1

Symbol: U2C1 | Address: 037D$_{16}$ | After Reset: 00000010$_2$

Bit Symbol	Bit Name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in U2TB register 1 : No data present in U2TB register	RO
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag	0 : No data present in U2RB register 1 : Data present in U2RB register	RO
U2IRS	UART2 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	RW
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	RW
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	RW
U2ERE	Error signal output enable bit	0 : Output disabled 1 : Output enabled	RW

Pin Assignment Control Register (1)



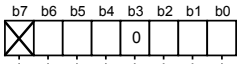
Symbpl: PACR | Address: 025D$_{16}$ | After Reset: 00$_{16}$

Bit Symbol	Bit Name	Function	RW
PACR0	Pin enabling bit	010 : 64 pin 011 : 80 pin All other values are reserved. Do not use.	RW
PACR1			RW
PACR2			RW
— (b6-b3)	Reserved bits	Nothing is assigned. If necessary, set to 0. When read, the content is 0	—
U1MAP	UART1 pin remapping bit	UART1 pins assigned to 0 : P6$_7$ to P6$_4$ 1 : P7$_3$ to P7$_0$	RW

NOTE:
1. Set the PACR register by the next instruction after setting the PRC2 bit in the PRCR register to 1(write enable).

Figure 14.7 U0C1 to U2C1 Register, and PACR Register

UART2 Special Mode Register

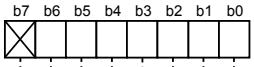


Symbol	Address	After Reset
U2SMR	037716	X0000000$_2$

Bit Symbol	Bit Name	Function	RW
IICM	I^{2}C mode select bit	0 : Other than I^{2}C mode 1 : I^{2}C mode	RW
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	RW
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected (busy)	RW(1)
—— (b3)	Reserved bit	Set to 0	RW
ABSCS	Bus collision detect sampling clock select bit	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	RW
ACSE	Auto clear function select bit of transmit enable bit	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	RW
SSS	Transmit start condition select bit	0 : Not synchronized to RxD2 1 : Synchronized to RxD2(2)	RW
—— (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		——

NOTES:
1: The BBS bit is set to 0 by writing 0 by program. (Writing 1 has no effect).
2: When a transfer begins, the SSS bit is set to 0 (Not synchronized to RxD2).

UART2 Special Mode Register 2

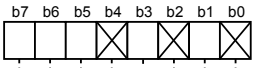


Symbol	Address	After Reset
U2SMR2	037616	X0000000$_2$

Bit Symbol	Bit Name	Function	RW
IICM2	I^{2}C bus mode select bit 2	Refer to **Table 14.13**	RW
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	RW
SWC	SCL2 wait output bit	0 : Disabled 1 : Enabled	RW
ALS	SDA2 output stop bit	0 : Disabled 1 : Enabled	RW
STAC	UART initialization bit	0 : Disabled 1 : Enabled	RW
SWC2	SCL2 wait output bit 2	0: Transfer clock 1: "L" output	RW
SDHI	SDA2 output disable bit	0: Enabled 1: Disabled (high impedance)	RW
—— (b7)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		——

Figure 14.8 U2SMR and U2SMR2 Registers

UART2 Special Mode Register 3

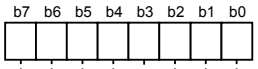


Symbol	Address	After Reset
U2SMR3	037516	$000X0X0X2$

Bit Symbol	Bit Name	Function	RW
— (b0)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
CKPH	Clock phase set bit	0 : Without clock delay 1 : With clock delay	RW
— (b2)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
NODC	Clock output select bit	0 : CLK_2 is CMOS output 1 : CLK_2 is N-channel open drain output	RW
— (b4)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
DL0	SDA2 digital delay setup bit (1, 2)	b7 b6 b5 0 0 0 : Without delay 0 0 1 : 1 to 2 cycle(s) of U2BRG count source 0 1 0 : 2 to 3 cycles of U2BRG count source 0 1 1 : 3 to 4 cycles of U2BRG count source 1 0 0 : 4 to 5 cycles of U2BRG count source 1 0 1 : 5 to 6 cycles of U2BRG count source 1 1 0 : 6 to 7 cycles of U2BRG count source 1 1 1 : 7 to 8 cycles of U2BRG count source	RW
DL1			RW
DL2			RW

NOTES:
1. Bits DL2 to DL0 are used to generate a delay in SDA output by digital means during I^2C bus mode. In other than I^2C bus mode, set these bits to 000_2 (no delay).
2. The amount of delay varies with the load on pins SCL2 and SDA2. Also, when using an external clock, the amount of delay increases by about 100 ns.

UART2 Special Mode Register 4



Symbol	Address	After Reset
U2SMR4	037416	0016

Bit Symbol	Bit Name	Function	RW
STAREQ	Start condition generate bit (1)	0 : Clear 1 : Start	RW
RSTAREQ	Restart condition generate bit (1)	0 : Clear 1 : Start	RW
STPREQ	Stop condition generate bit (1)	0 : Clear 1 : Start	RW
STSPSEL	SCL_2,SDA_2 output select bit	0 : Start and stop conditions not output 1 : Start and stop conditions output	RW
ACKD	ACK data bit	0 : ACK 1 : NACK	RW
ACKC	ACK data output enable bit	0 : Serial I/O data output 1 : ACK data output	RW
SCLHI	SCL_2 output stop enable bit	0 : Disabled 1 : Enabled	RW
SWC9	SCL_2 wait bit 3	0 : SCL_2 “L” hold disabled 1 : SCL_2 “L” hold enabled	RW

NOTE:
1. Set to 0 when each condition is generated.

Figure 14.9 U2SMR3 and U2SMR4 Registers

14.1.1 Clock Synchronous serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. **Table 14.1** lists the specifications of the clock synchronous serial I/O mode. **Table 14.2** lists the registers used in clock synchronous serial I/O mode and the register values set.

Table 14.1 Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• The CKDIR bit in the UiMR(i=0 to 2) register is set to 0 (internal clock) : fj/ (2(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • CKDIR bit is set to 1 (external clock) : Input from CLKi pin
Transmission, reception control	• Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}/\overline{RTS}$ function disable
Transmission start condition	• Before transmission can start, the following requirements must be met (1) – The TE bit in the UiC1 register is set to 1 (transmission enabled) – The TI bit in the UiC1 register is set to 0 (data present in UiTB register) – If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin is set to "L"
Reception start condition	• Before reception can start, the following requirements must be met (1) – The RE bit in the UiC1 register is set to 1 (reception enabled) – The TE bit in the UiC1 register is set to 1 (transmission enabled) – The TI bit in the UiC1 register is set to 0 (data present in the UiTB register)
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – The UiIRS bit (3) is set to 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit is set to 1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register • For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (2) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the 7th bit in the the next data
Select function	• CLK polarity selection Transfer data input/output can be chosen to occur synchronously with the rising or the falling edge of the transfer clock • LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Continuous receive mode selection Reception is enabled immediately by reading the UiRB register • Switching serial data logic (UART2) This function reverses the logic value of the transmit/receive data • Transfer clock output from multiple pins selection (UART1) The output pin can be selected in a program from two UART1 transfer clock pins that have been set • Separate $\overline{CTS}/\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins • UART1 pin remapping selection The UART1 pin can be selected from $P6_7$ to $P6_4$ or $P7_3$ to $P7_0$

NOTES:

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register is set to 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. If an overrun error occurs, bits 8 to 0 in the UiRB register (i = 0 to 2) are undefined. The IR bit in the SiRIC register remains unchanged.
3. The U0IRS and U1IRS bits respectively are the bits 0 and 1 in the UCON register; the U2IRS bit is bit 4 in the U2C1 register.

Table 14.2 Registers to Be Used and Settings in Clock Synchronous Serial I/O Mode

Register	Bit	Function
UiTB(3)	0 to 7	Set transmission data
UiRB(3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(3)	SMD2 to SMD0	Set to 001_2
	CKDIR	Select the internal clock or external clock
	IOPOL(i=2) (4)	Set to 0
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function
	NCH	Select TxDi pin output mode
	CKPOL	Select the transfer clock polarity
	UFORM	Select the LSB first or MSB first
UiC1	TE	Set this bit to 1 to enable transmission/reception
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS (1)	Select the source of UART2 transmit interrupt
	U2RRM (1)	Set this bit to 1 to use UART2 continuous receive mode
	U2LCH (3)	Set this bit to 1 to use UART2 inverted data logic
	U2ERE (3)	Set to 0
U2SMR	0 to 7	Set to 0
U2SMR2	0 to 7	Set to 0
U2SMR3	0 to 2	Set to 0
	NODC	Select clock output mode
	4 to 7	Set to 0
U2SMR4	0 to 7	Set to 0
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set this bit to 1 to use continuous receive mode
	CLKMD0	Select the transfer clock output pin when CLKMD1 is set to 1
	CLKMD1	Set this bit to 1 to output UART1 transfer clock from two pins
	RCSP	Set this bit to 1 to accept as input the UART0 $\overline{CTS_0}$ signal from the $P6_4$ pin
	7	Set to 0

NOTES:

1. Set bits 5 and 4 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
2. Not all register bits are described above. Set those bits to 0 when writing to the registers in clock synchronous serial I/O mode.
3. Set bits 7 and 6 in registers U0C1 and U1C1 to 0.
4. Set the bit 7 in registers U0MR and U1MR to 0.

i=0 to 2

Table 14.3 lists pin functions for the case where the multiple transfer clock output pin select function is deselected. **Table 14.4** lists the P64 pin functions during clock synchronous serial I/O mode.
Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 14.3 Pin Functions (When Not Select Multiple Transfer Clock Output Pin Function)(1)

Pin Name	Function	Method of Selection
TxDi (i = 0 to 2) (P63, P67, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P62, P66, P71)	Serial data input	Set the PD6_2 bit and PD6_6 bit in the PD6 register, and PD7_1 bit in the PD7 register to 0 (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72)	Transfer clock output	Set the CKDIR bit in the UiMR register to 0
	Transfer clock input	Set the CKDIR bit in the UiMR register to 1 Set the PD6_1 bit and PD6_5 bit in the PD6 register, and the PD7_2 bit in the PD7 register to 0
$\overline{CTS}i/\overline{RTS}i$ (P60, P64, P73)	$\overline{CTS}$ input	Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 0 Set the PD6_0 bit and PD6_4 bit in the PD6 register is set to 0, the PD7_3 bit in the PD7 register to 0
	$\overline{RTS}$ output	Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 1
	I/O port	Set the CRD bit in the UiC0 register to 1

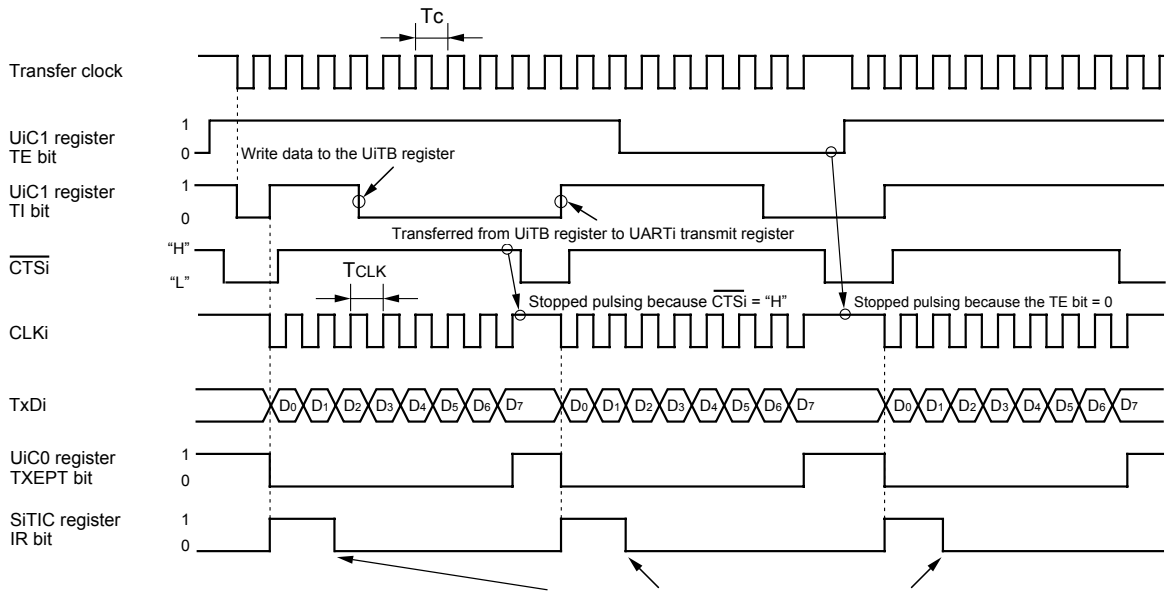
NOTE:
1: When the U1MAP bit in PACR register is 1 (P73 to P70), UART1 pin is assgined to P73 to P70.

Table 14.4 P64 Pin Functions(1)

Pin Function	Bit Set Value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P64	1	—	0	0	—	Input: 0, Output: 1
$\overline{CTS}_1$	0	0	0	0	—	0
$\overline{RTS}_1$	0	1	0	0	—	—
$\overline{CTS}_0$(2)	0	0	1	0	—	0
CLKS1	—	—	—	1(3)	1	—

NOTES:
1. When the U1MAP bit in PACR register is 1 (P73 to P70), this table lists the P70 functions.
2. In addition to this, set the CRD bit in the U0C0 register to 0 ($\overline{CT0}_0/\overline{RT0}_0$ enabled) and the CRS bit in the U0C0 register to 1 ($\overline{RTS}_0$ selected).
3. When the CLKMD1 bit is set to 1 and the CLKMD0 bit is set to 0, the following logic levels are output:
   • High if the CLKPOL bit in the U1C0 register is set to 0
   • Low if the CLKPOL bit in the U1C0 register is set to 1

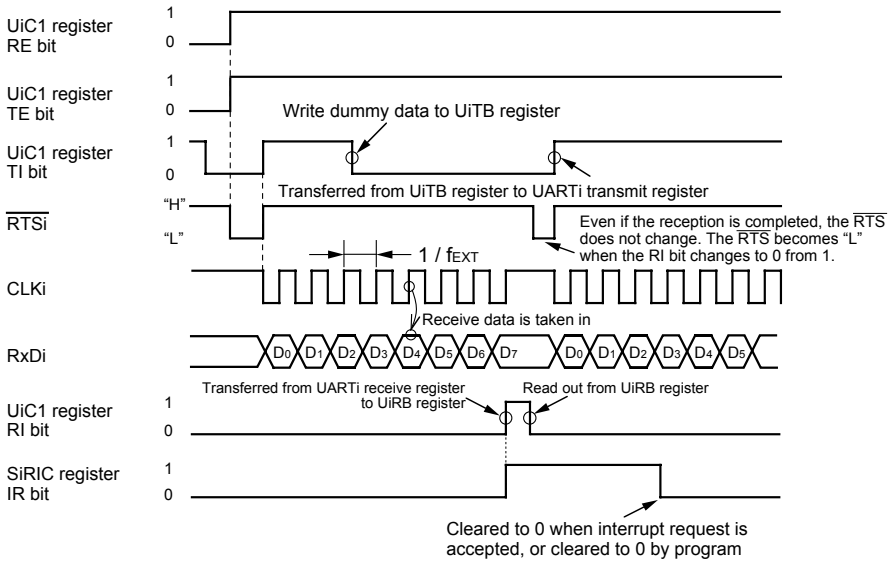
(1) Example of Transmit Timing (Internal clock is selected)



Tc = TCLK = 2(n + 1) / fj
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
n: value set to UiBRG register
i: 0 to 2

The above timing diagram applies to the case where the register bits are set as follows:
- The CKDIR bit in the UiMR register is set to 0 (internal clock)
- The CRD bit in the UiC0 register is set to 0 (CTS/RTS enabled); CRS bit is set to 0 (CTS selected)
- The CKPOL bit in the UiC0 register is set to 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)
- The UiIRS bit is set to 0 (an interrupt request occurs when the transmit buffer becomes empty): U0IRS bit is the bit 0 in the UCON register U1IRS bit is the bit 1 in the UCON register, and U2IRS bit is the bit 4 in the U2C1 register.

(2) Example of Receive Timing (External clock is selected)



The above timing diagram applies to the case where the register bits are set as follows:
- The CKDIR bit in the UiMR register is set to 1 (external clock)
- The CRD bit in the UiC0 register is set to 0 (CTS/RTS enabled); The CRS bit is set to 1 (RTS selected)
- UiC0 register CKPOL bit is set to 0 (transmit data output at the falling edge and receive data taken in at the rising edge of the transfer clock)

fEXT: frequency of external clock

Make sure the following conditions are met when input to the CLKi pin before receiving data is high:
- UiC0 register TE bit is set to 1 (transmit enabled)
- UiC0 register RE bit is set to 1 (Receive enabled)
- Write dummy data to the UiTB register

Figure 14.10 Typical transmit/receive timings in clock synchronous serial I/O mode

14.1.1.1 Counter Measure for Communication Error Occurs

If a communication error occurs while transmitting or receiving in clock synchronous serial I/O mode, follow the procedures below.

•Resetting the UiRB register (i=0 to 2)

(1) Set the RE bit in the UiC1 register to 0 (reception disabled)

(2) Set bits SMD2 to SMD0 in the UiMR register to 000_2 (Serial I/O disabled)

(3) Set bits SMD2 to SMD0 in the UiMR register to 001_2 (Clock synchronous serial I/O mode)

(4) Set the RE bit in the UiC1 register to 1 (reception enabled)

•Resetting the UiTB register (i=0 to 2)

(1) Set bits SMD2 to SMD0 in the UiMR register to 000_2 (Serial I/O disabled)

(2) Set bits SMD2 to SMD0 in the UiMR register to 001_2 (Clock synchronous serial I/O mode)

(3) 1 is written to TE bit in the UiC1 register (reception enabled), regardless to the TE bit.

14.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i=0 to 2) to select the transfer clock polarity. **Figure 14.11** shows the polarity of the transfer clock.

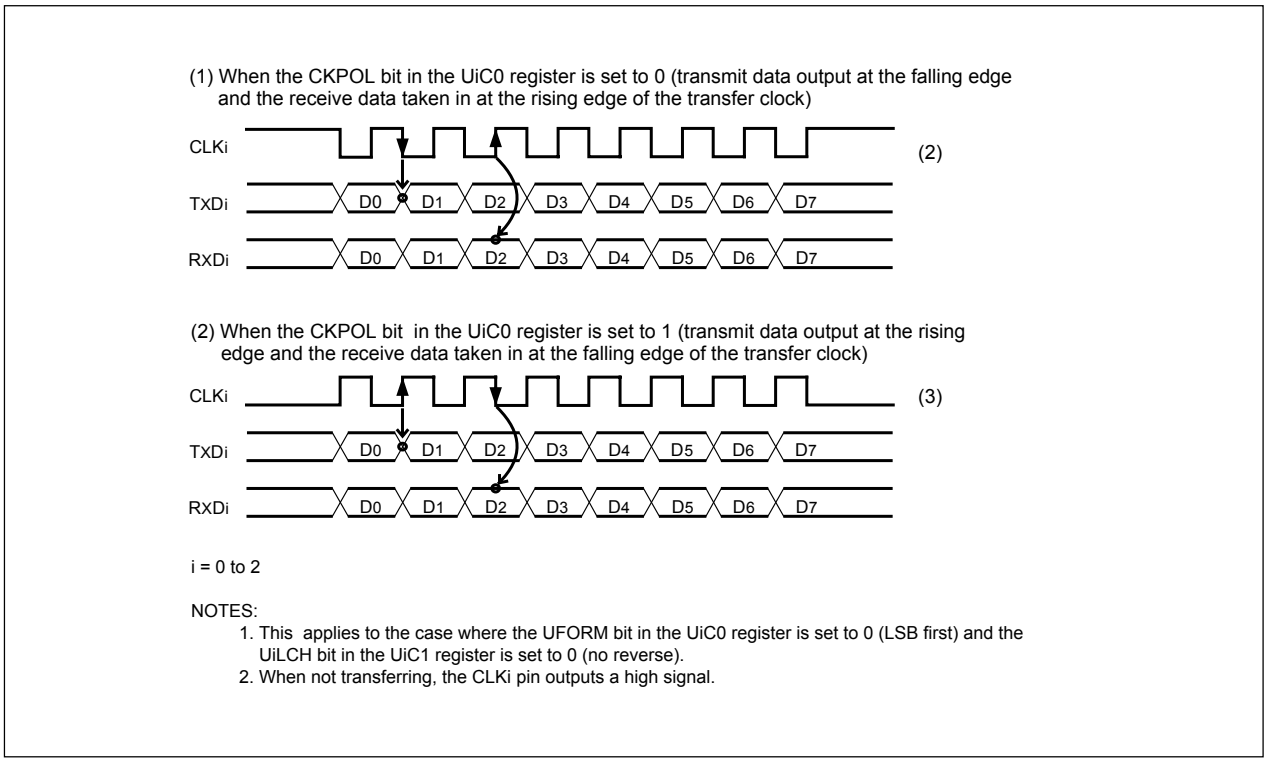


Figure 14.11 Polarity of transfer clock

14.1.1.3 LSB First/MSB First Select Function

Use the UFORM bit in the UiC0 register (i=0 to 2) to select the transfer format. **Figure 14.12** shows the transfer format.

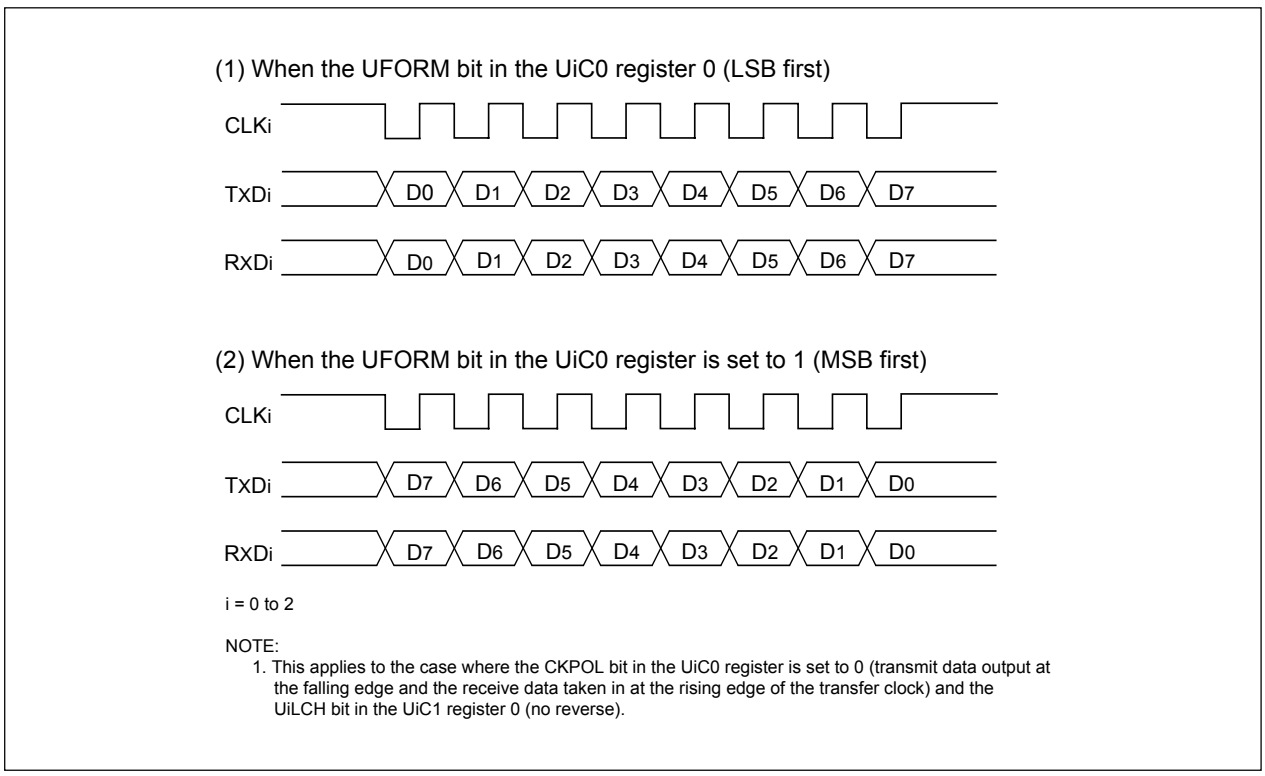


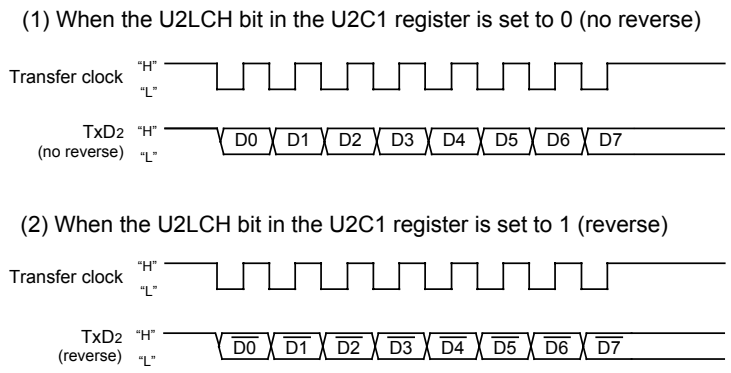
Figure 14.12 Transfer format

14.1.1.4 Continuous receive mode

When the UiRRM bit (i=0 to 2) is set to 1 (continuous receive mode), the TI bit in the UiC1 register is set to “0” (data present in the UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit is set to 1, do not write dummy data to the UiTB register in a program. The U0RRM and U1RRM bits are the bit 2 and bit 3 in the UCON register, respectively, and the U2RRM bit is the bit 5 in the U2C1 register.

14.1.1.5 Serial data logic switch function (UART2)

When the U2LCH bit in the U2C1 register is set to 1 (reverse), the data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. **Figure 14.13** shows serial data logic.

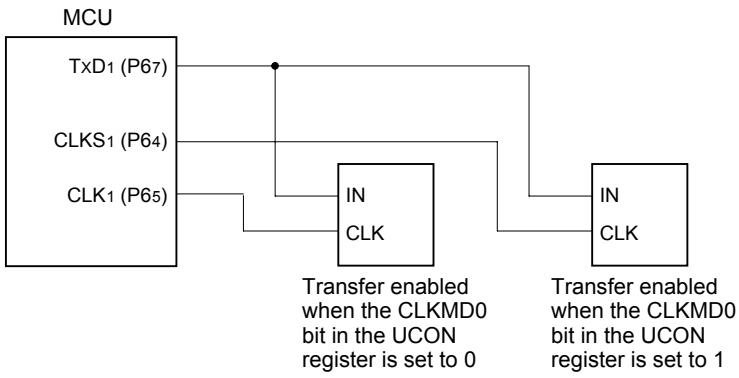


NOTE:
1. This applies to the case where the CKPOL bit in the U2C0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock) and the UFORM bit is set to 0 (LSB first).

Figure 14.13 Serial data logic switch timing

14.1.1.6 Transfer clock output from multiple pins function (UART1)

The CLKMD1 to CLKMD0 bits in the UCON register can choose one from two transfer clock output pins. (See **Figure 14.14**) This function is valid when the internal clock is selected for UART1.



NOTES:
1. This applies to the case where the CKDIR bit in the U1MRregister is set to 0 (internal clock) and the CLKMD1 bit in the UCON register is set to 1 (transfer clock output from multiple pins).
2. This applies to the case where U1MAP bit in PACR register is set to 0 (P67 to P64).

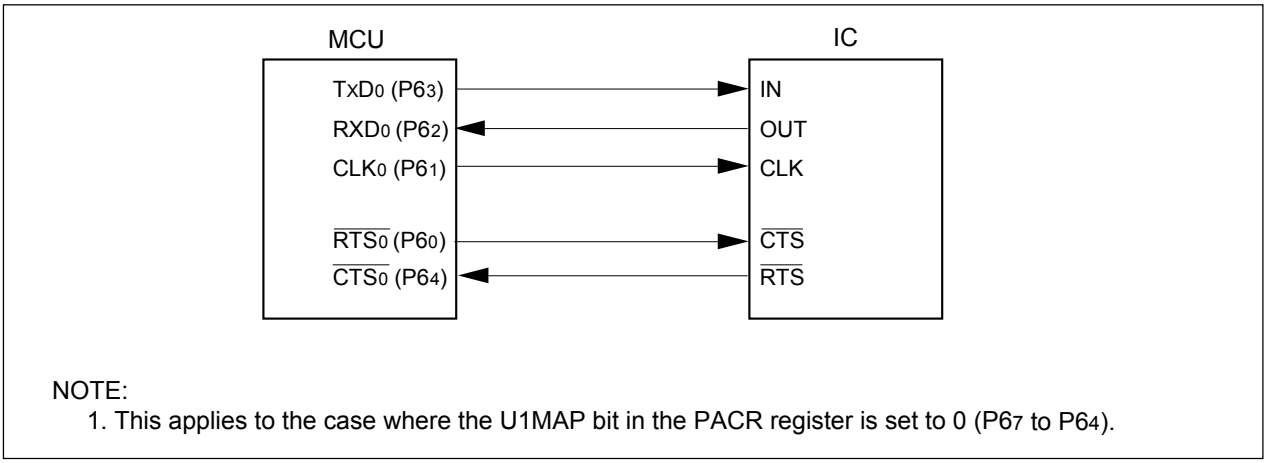
Figure 14.14 Transfer Clock Output From Multiple Pins

14.1.1.7 $\overline{CTS}/\overline{RTS}$ separate function (UART0)

This function separates $\overline{CTS}_0/\overline{RTS}_0$, outputs $\overline{RTS}_0$ from the $P6_0$ pin, and accepts as input the $\overline{CTS}_0$ from the $P6_4$ pin or $P7_0$ pin. To use this function, set the register bits as shown below.

- The CRD bit in the U0C0 register is set to 0 (enables UART0 $\overline{CTS}/\overline{RTS}$)
- The CRS bit in the U0C0 register is set to 1 (outputs UART0 $\overline{RTS}$)
- The CRD bit in the U1C0 register is set to 0 (enables UART1 $\overline{CTS}/\overline{RTS}$)
- The CRS bit in the U1C0 register is set to 0 (inputs UART1 $\overline{CTS}$)
- The RCSP bit in the UCON register is set to 1 (inputs $\overline{CTS}_0$ from the $P6_4$ pin or $P7_0$ pin)
- The CLKMD1 bit in the UCON register is set to 0 ($CLKS_1$ not used)

Note that when using the $\overline{CTS}/\overline{RTS}$ separate function, UART1 $\overline{CTS}/\overline{RTS}$ separate function cannot be used.



NOTE:
1. This applies to the case where the U1MAP bit in the PACR register is set to 0 ($P6_7$ to $P6_4$).

Figure 14.15 CTS/RTS separate function usage

14.1.2 Clock Asynchronous Serial I/O (UART) Mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. **Tables 14.5** lists the specifications of the UART mode.

Table 14.5 UART Mode Specifications

Item	Specification
Transfer data format	• Character bit (transfer data): Selectable from 7, 8 or 9 bits • Start bit: 1 bit • Parity bit: Selectable from odd, even, or none • Stop bit: Selectable from 1 or 2 bits
Transfer clock	• The CKDIR bit in the UiMR(i=0 to 2) register is set to 0 (internal clock) : fj/ (16(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of UiBRG register 00_{16} to FF_{16} • CKDIR bit is set to "1" (external clock) : f_{EXT}/16(n+1) f_{EXT}: Input from CLKi pin. n :Setting value of UiBRG register 00_{16} to FF_{16}
Transmission, reception control	• Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS}$/$\overline{RTS}$ function disable
Transmission start condition	• Before transmission can start, the following requirements must be met – The TE bit in the UiC1 register is set to 1 (transmission enabled) – The TI bit in the UiC1 register is set to 0 (data present in UiTB register) – If $\overline{CTS}$ function is selected, input on the $\overline{CTS}$i pin is set to "L"
Reception start condition	• Before reception can start, the following requirements must be met" – The RE bit in the UiC1 register is set to 1 (reception enabled) – Start bit detection
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – The UiIRS bit (2) is set to 0 (transmit buffer empty): when transferring data from the UiTB register to the UARTi transmit register (at start of transmission) – The UiIRS bit is set to1 (transfer completed): when the serial I/O finished sending data from the UARTi transmit register • For reception When transferring data from the UARTi receive register to the UiRB register (at completion of reception)
Error detection	• Overrun error (1) This error occurs if the serial I/O started receiving the next data before reading the UiRB register and received the bit one before the last stop bit in the the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error This error occurs when if parity is enabled, the number of 1 in parity and character bits does not match the number of 1 set • Error sum flag This flag is set to 1 when any of the overrun, framing, and parity errors is encountered
Select function	• LSB first, MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Serial data logic switch (UART2) This function reverses the logic of the transmit/receive data. The start and stop bits are not reversed. • TxD, RxD I/O polarity switch (UART2) This function reverses the polarities of hte TxD pin output and RxD pin input. The logic levels of all I/O data is reversed. • Separate $\overline{CTS}$/$\overline{RTS}$ pins (UART0) $\overline{CTS_0}$ and $\overline{RTS_0}$ are input/output from separate pins • UART1 pin remapping selection The UART1 pin can be selected from the $P6_7$ to $P6_4$ or $P7_3$ to $P7_0$

NOTES:
1. If an overrun error occurs, bits 8 to 0 in the UiRB (i=0 to 2) register are undefined. The IR bit in the SiRIC register remains unchanged.
2. Bits U0IRS and U1IRS respectively are the UCON register bits 0 and 1; the U2IRS bit is the U2C1 register bit 4.

Table 14.6 Registers to Be Used and Settings in UART Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data (1)
UiRB	0 to 8	Reception data can be read (1)
	OER,FER,PER,SUM	Error flag
UiBRG	0 to 7	Set a transfer rate
UiMR	SMD2 to SMD0	Set these bits to 100_2 when transfer data is 7 bits long Set these bits to 101_2 when transfer data is 8 bits long Set these bits to 110_2 when transfer data is 9 bits long
	CKDIR	Select the internal clock or external clock
	STPS	Select the stop bit
	PRY, PRYE	Select whether parity is included and whether odd or even
	IOPOL(i=2) (4)	Select the TxD/RxD input/output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select $\overline{CTS}$ or $\overline{RTS}$ to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the $\overline{CTS}$ or $\overline{RTS}$ function
	NCH	Select TxDi pin output mode
	CKPOL	Set to 0
	UFORM	LSB first or MSB first can be selected when transfer data is 8 bits long. Set this bit to 0 when transfer data is 7 or 9 bits long.
UiC1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS (2)	Select the source of UART2 transmit interrupt
	U2RRM (2)	Set to 0
	UiLCH (3)	Set this bit to 1 to use UART2 inverted data logic
	UiERE (3)	Set to 0
UiSMR	0 to 7	Set to 0
UiSMR2	0 to 7	Set to 0
UiSMR3	0 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to 0
	CLKMD0	Invalid because CLKMD1 is set to 0
	CLKMD1	Set to 0
	RCSP	Set this bit to 1 to accept as input the UART0 $\overline{CTS_0}$ signal from the $P6_4$ pin
	7	Set to 0

NOTES:

1. The bits used for transmit/receive data are as follows: Bit 0 to bit 6 when transfer data is 7 bits long; bits 7 to 0 when transfer data is 8 bits long; bit 0 to bit 8 when transfer data is 9 bits long.
2. Set bits 5 and 4 in registers U0C1 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM and U1RRM are included in the UCON register.
3. Set bits 7 and 6 in registers U0C1 and U1C1 to 0.
4. Set the bit 7 in registers U0MR and U1MR to 0.

i=0 to 2

Table 14.7 lists the functions of the input/output pins in UART mode. **Table 14.8** lists the P6$_4$ pin functions during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 14.7 I/O Pin Functions in UART mode(1)

Pin Name	Function	Method of Selection
TxDi (i = 0 to 2) (P6$_3$, P6$_7$, P7$_0$)	Serial data output	(Outputs "H" when performing reception only)
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	PD6_2 bit, PD6_6 bit in the PD6 register and the PD7_1 bit in the PD7 register (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Input/output port	Set the CKDIR bit in the UiMR register to 0
	Transfer clock input	Set the CKDIR bit in the UiMR register to 1 Set the PD6_1 bit and PD6_5 bit in the PD6 register to 0, PD7_2 bit in the PD7 register to 0
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ (P6$_0$, P6$_4$, P7$_3$)	$\overline{\text{CTS}}$ input	Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 0 Set the PD6_0 bit and PD6_4 bit in the PD6 register to 0, the PD7_3 bit in the PD7 register 0
	$\overline{\text{RTS}}$ output	Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 1
	Input/output port	Set the CRD bit in the UiC0 register 1

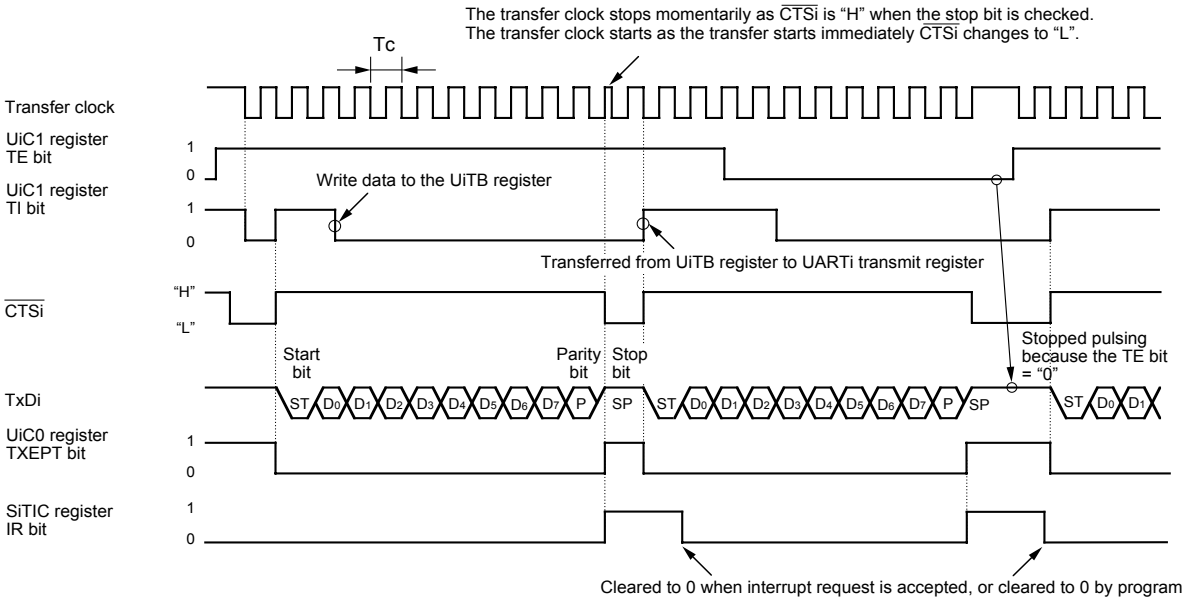
NOTE:
1. When the U1MAP bit in PACR register is set to 1 (P7$_3$ to P7$_0$), UART1 pin is assgined to P7$_3$ to P7$_0$.

Table 14.8 P6$_4$ Pin Functions in UART mode (1)

Pin Function	Bit Set Value				
	U1C0 register		UCON register		PD6 register
	CRD	CRS	RCSP	CLKMD1	PD6_4
P6$_4$	1	—	0	0	Input: 0, Output: 1
$\overline{\text{CTS}}_1$	0	0	0	0	0
$\overline{\text{RTS}}_1$	0	1	0	0	—
$\overline{\text{CTS}}_0$ (2)	0	0	1	0	0

NOTES:
1. When the U1MAP bit in PACR register is 1 (P7$_3$ to P7$_0$), this table lists the P7$_0$ functions.
2. In addition to this, set the CRD bit in the U0C0 register to 0 ($\overline{\text{CTS}}_0$/$\overline{\text{RTS}}_0$ enabled) and the CRS bit in the U0C0 register to 1 ($\overline{\text{RTS}}_0$ selected).

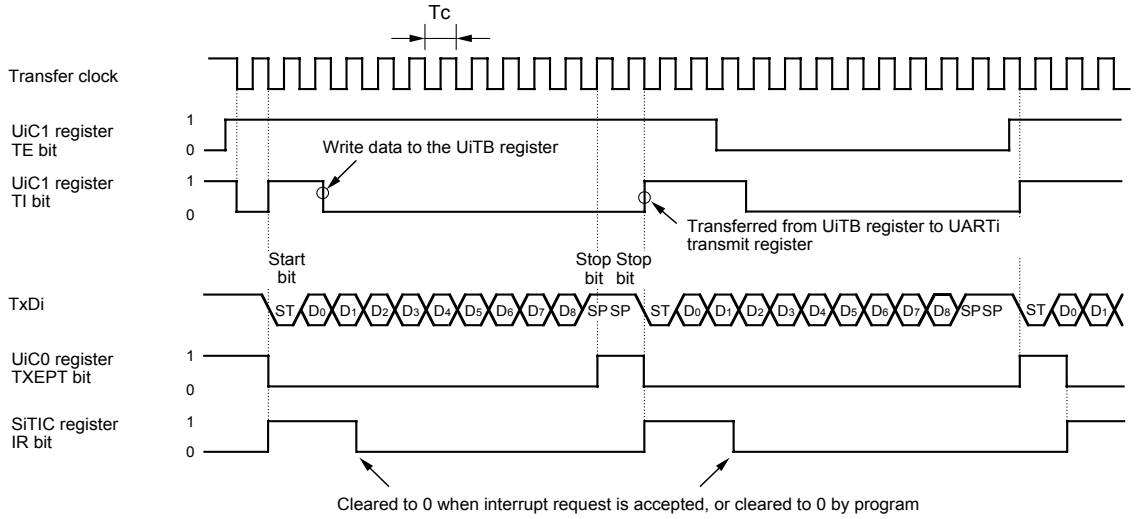
• Example of transmit timing when transfer data is 8-bit long (parity enabled, one stop bit)



The above timing diagram applies to the case where the register bits are set as follows:
- Set the PRYE bit in the UiMR register to 1 (parity enabled)
- Set the STPS bit in the UiMR register to 0 (1 stop bit)
- Set the CRD bit in the UiC0 register to 0 (CTS/RTS enabled), the CRS bit to 0 (CTS selected).
- Set the UiIRS bit to 1 (an interrupt request occurs when transmit completed):
  U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$Tc = 16 (n + 1) / fj$ or $16 (n + 1) / f_{EXT}$
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
f_{EXT}: frequency of UiBRG count source (external clock)
n: value set to UiBRG
i = 0 to 2

• Example of transmit timing when transfer data is 9-bit long (parity disabled, two stop bits)



The above timing diagram applies to the case where the register bits are set as follows:
- Set the PRYE bit in the UiMR register to 0 (parity disabled)
- Set the STPS bit in the UiMR register to 1 (2 stop bits)
- Set the CRD bit in the UiC0 register to 1 (CTS/RTS disabled)
- Set the UiIRS bit to 0 (an interrupt request occurs when transmit buffer becomes empty):
  U0IRS bit is the UCON register bit 0, U1IRS bit is the UCON register bit 1, and U2IRS bit is the U2C1 register bit 4

$Tc = 16 (n + 1) / fj$ or $16 (n + 1) / f_{EXT}$
fj: frequency of UiBRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
f_{EXT}: frequency of UiBRG count source (external clock)
n: value set to UiBRG
i = 0 to 2

Figure 14.16 Typical transmit timing in UART mode (UART0, UART1)

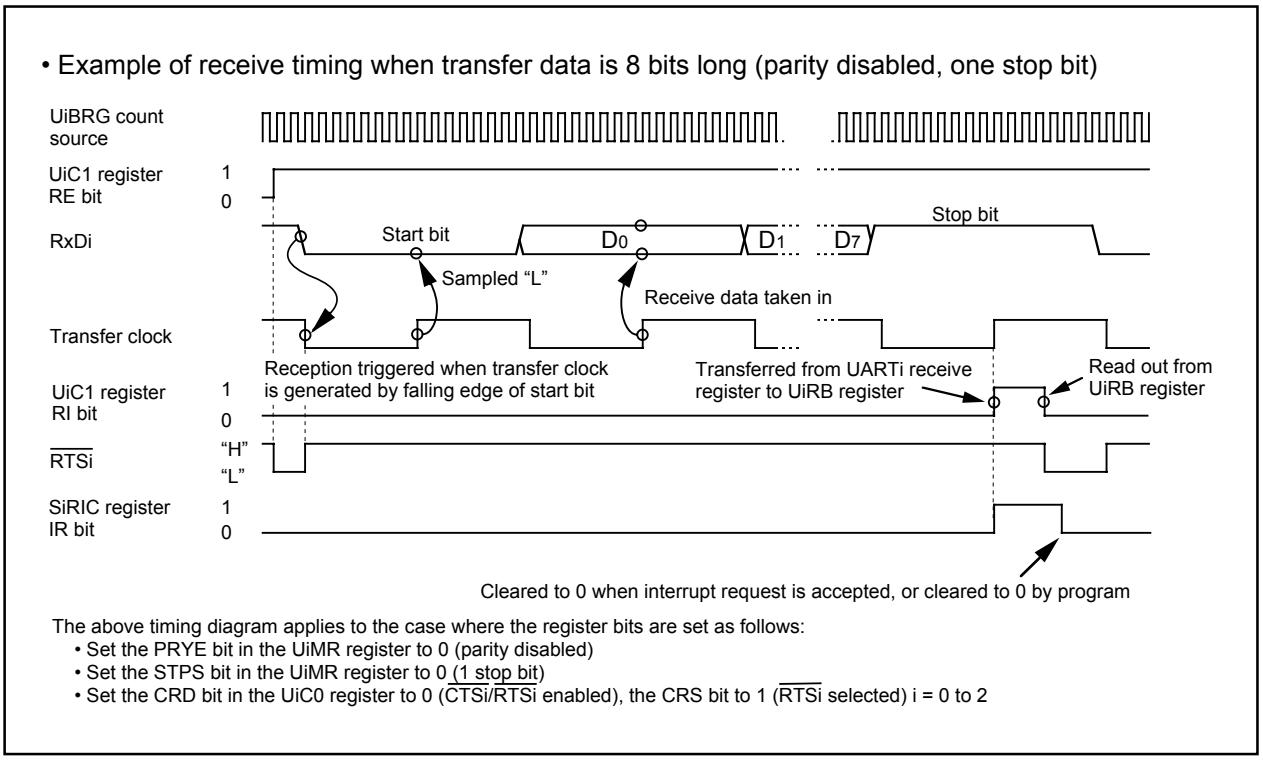


Figure 14.17 Receive Operation

14.1.2.1 Bit Rates

In UART mode, the frequency set by the UiBRG register (i=0 to 2) divided by 16 become the bit rates. **Table 14.9** lists example of bit rate and settings.

Table 14.9 Example of Bit Rates and Settings

Bit Rate (bps)	Count Source of BRG	Peripheral Function Clock : 16MHz		Peripheral Function Clock : 20MHz	
		Set Value of BRG : n	Actual Time (bps)	Set Value of BRG : n	Actual Time (bps)
1200	f8	103(67h)	1202	129(81h)	1202
2400	f8	51(33h)	2404	64(40h)	2404
4800	f8	25(19h)	4808	32(20h)	4735
9600	f1	103(67h)	9615	129(81h)	9615
14400	f1	68(44h)	14493	86(56h)	14368
19200	f1	51(33h)	19231	64(40h)	19231
28800	f1	34(22h)	28571	42(2Ah)	29070
31250	f1	31(1Fh)	31250	39(27h)	31250
38400	f1	25(19h)	38462	32(20h)	37879
51200	f1	19(13h)	50000	24(18h)	50000

14.1.2.2 Counter Measure for Communication Error

If a communication error occurs while transmitting or receiving in UART mode, follow the procedure below.

• Resetting the UiRB register (i=0 to 2)

(1) Set the RE bit in the UiC1 register to 0 (reception disabled)

(2) Set the RE bit in the UiC1 register to 1 (reception enabled)

• Resetting the UiTB register (i=0 to 2)

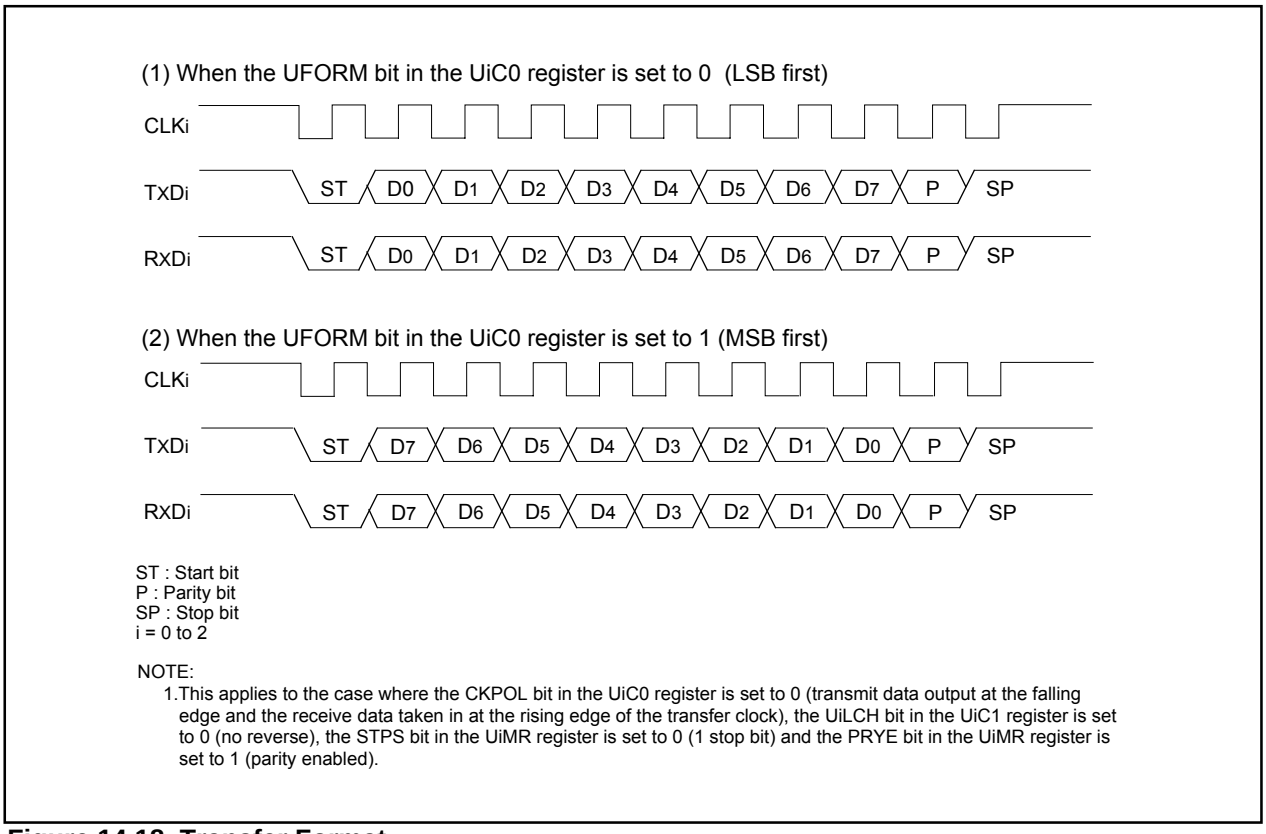
(1) Set bits SMD2 to SMD0 in UiMR register 000_2 (Serial I/O disabled)

(2) Set bits SMD2 to SMD0 in UiMR register 001_2, 101_2, 110_2

(3) 1 is written to TE bit in the UiC1 register (reception enabled), regardless of the TE bit

14.1.2.3 LSB First/MSB First Select Function

As shown in **Figure 14.18**, use the UFORM bit in the UiC0 register to select the transfer format. This function is valid when transfer data is 8 bits long.



ST : Start bit
P : Parity bit
SP : Stop bit
i = 0 to 2

NOTE:
1. This applies to the case where the CKPOL bit in the UiC0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the UiLCH bit in the UiC1 register is set to 0 (no reverse), the STPS bit in the UiMR register is set to 0 (1 stop bit) and the PRYE bit in the UiMR register is set to 1 (parity enabled).

Figure 14.18 Transfer Format

14.1.2.4 Serial Data Logic Switching Function (UART2)

The data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. **Figure 14.19** shows serial data logic.

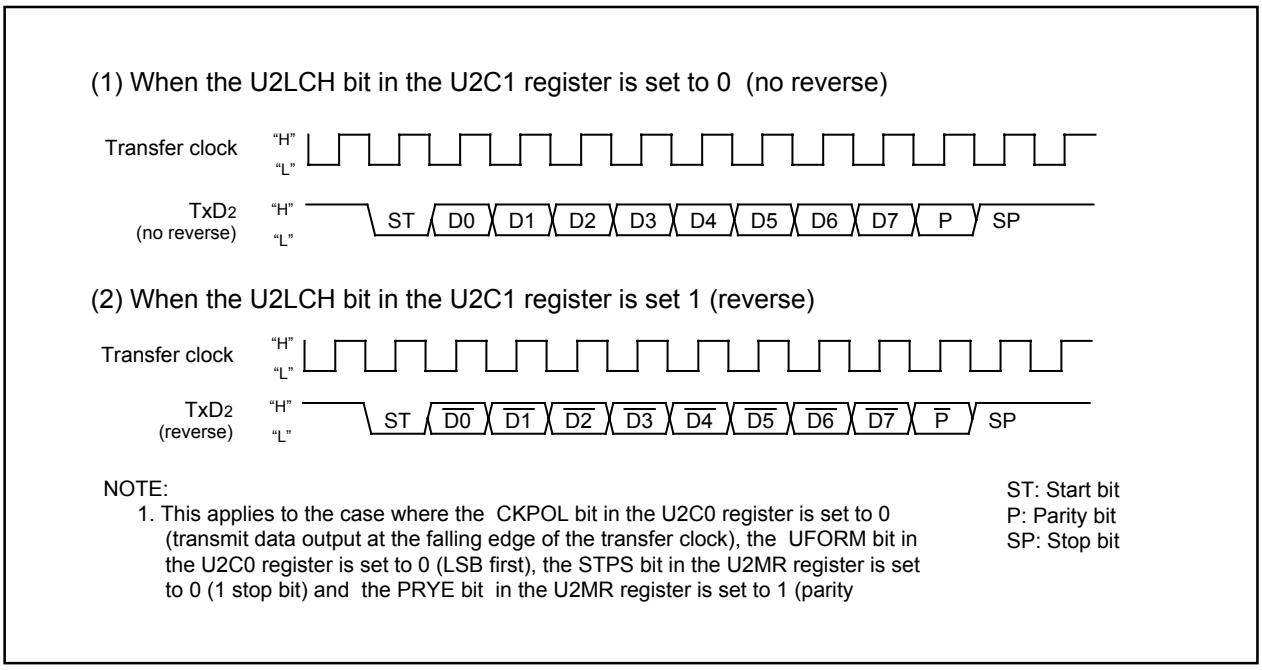


Figure 14.19 Serial Data Logic Switching

14.1.2.5 TxD and RxD I/O Polarity Inverse Function (UART2)

This function inverses the polarities of the TxD2 pin output and RxD2 pin input. The logic levels of all input/output data (including the start, stop and parity bits) are inversed. **Figure 14.20** shows the TxD pin output and RxD pin input polarity inverse.

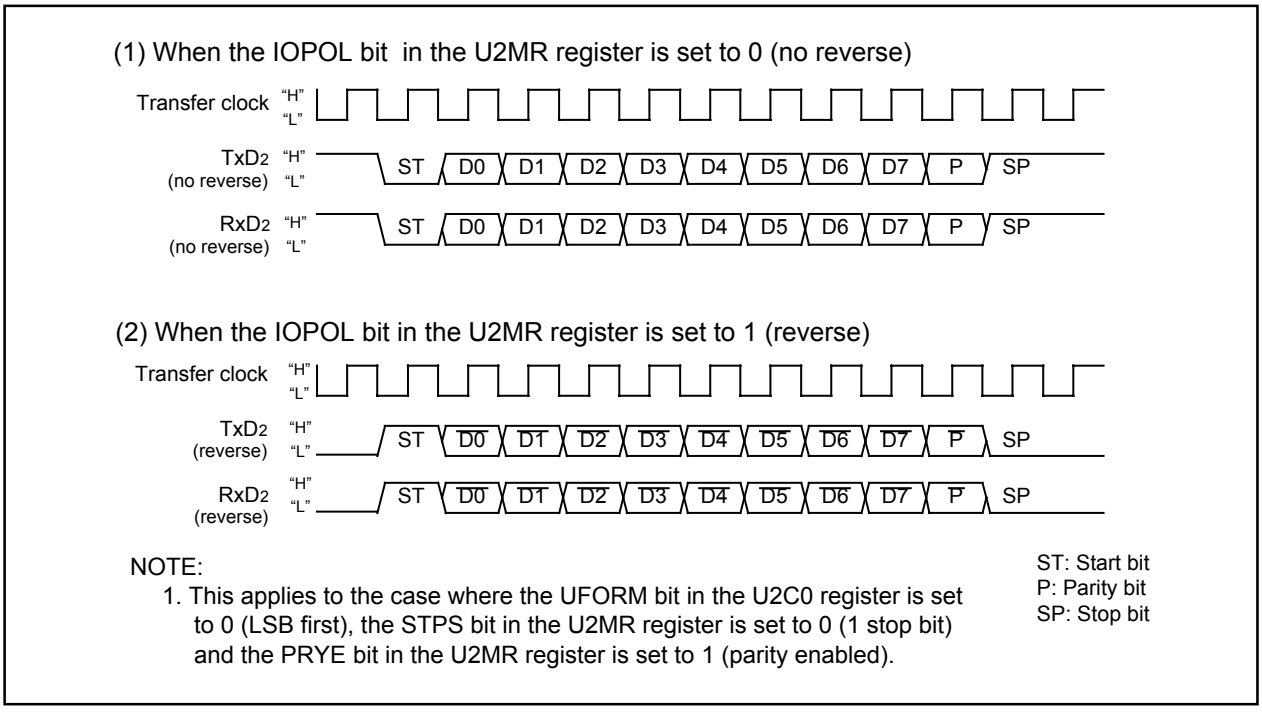


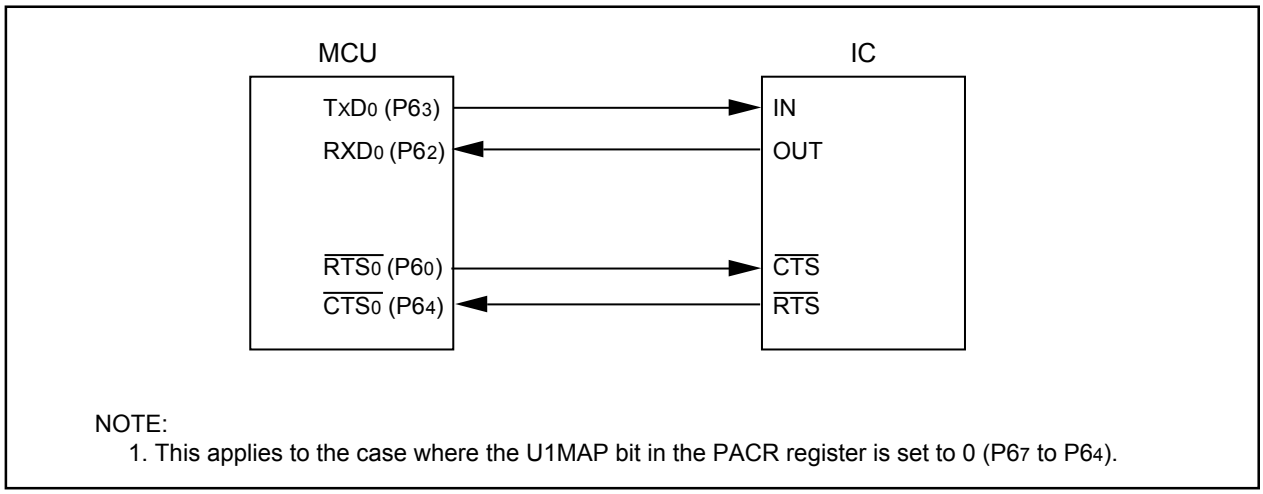
Figure 14.20 TxD and RxD I/O Polarity Inverse

14.1.2.6 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function (UART0)

This function separates $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$, outputs $\overline{\text{RTS}}_0$ from the $P6_0$ pin, and accepts as input the $\overline{\text{CTS}}_0$ from the $P6_4$ pin or $P7_0$ pin. To use this function, set the register bits as shown below.

- The CRD bit in the U0C0 register is set to 0 (enables UART0 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- The CRS bit in the U0C0 register is set to 1 (outputs UART0 $\overline{\text{RTS}}$)
- The CRD bit in the U1C0 register is set to 0 (enables UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$)
- The CRS bit in the U1C0 register is set to 0 (inputs UART1 $\overline{\text{CTS}}$)
- The RCSP bit in the UCON register is set to 1 (inputs $\overline{\text{CTS}}_0$ from the $P6_4$ pin or $P7_0$ pin)
- The CLKMD1 bit in the UCON register is set to 0 ($CLKS_1$ not used)

Note that when using the $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function, UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate function cannot be used.



NOTE:
1. This applies to the case where the U1MAP bit in the PACR register is set to 0 ($P6_7$ to $P6_4$).

Figure 14.21 $\overline{\text{CTS}}/\overline{\text{RTS}}$ Separate Function

14.1.3 Special Mode 1 (I^2C bus Mode) (UART2)

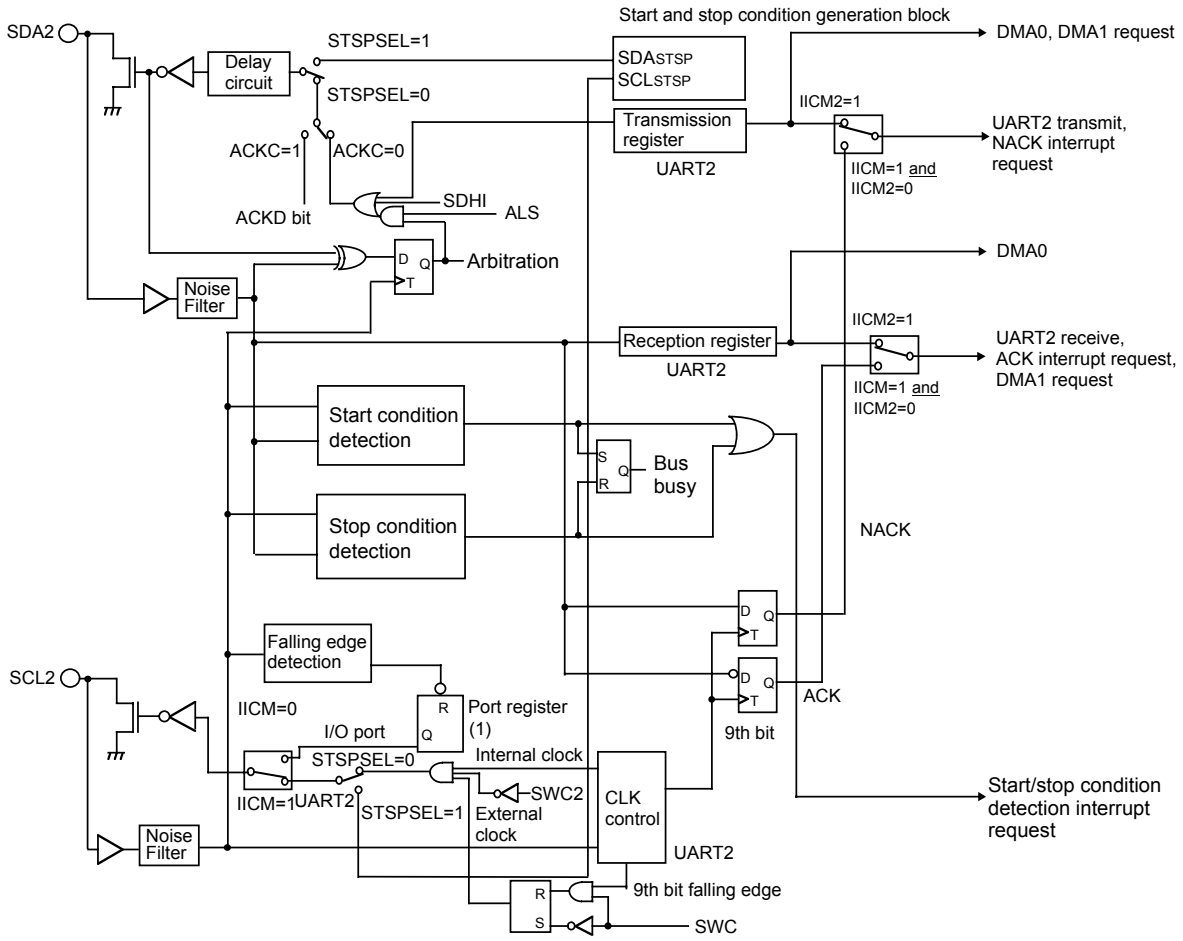
I^2C bus mode is provided for use as a simplified I^2C bus interface compatible mode. **Table 14.10** lists the specifications of the I^2C bus mode. **Tables 14.11** and **14.12** list the registers used in the I^2C bus mode and the register values set. **Table 14.13** lists the I^2C bus mode fuctions. **Figure 14.22** shows the block diagram for I^2C bus mode. **Figure 14.23** shows SCL_2 timing.

As shown in **Table 14.11**, the MCU is placed in I^2C bus mode by setting bits SMD2 to SMD0 to 010_2 and the IICM bit to 1. Because SDA_2 transmit output has a delay circuit attached, SDA output does not change state until SCL_2 goes low and remains stably low.

Table 14.10 I^2C bus Mode Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• During master the CKDIR bit in the U2MR register is set to 0 (internal clock) : fj/ (2(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value in the U2BRG register 00_{16} to FF_{16} • During slave CKDIR bit is set to 1 (external clock) : Input from SCL_2 pin
Transmission start condition	• Before transmission can start, the following requirements must be met (1) – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in U2TB register)
Reception start condition	• Before reception can start, the following requirements must be met (1) – The RE bit in the U2C1 register is set to 1 (reception enabled) – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in the UiTB register)
Interrupt request generation timing	When start / stop condition is detected, acknowledge undetected, and acknowledge detected
Error detection	• Overrun error (2) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 8th bit in the the next data
Select function	• Arbitration lost Timing at which the ABT bit in the U2RB register is updated can be selected • SDA digital delay No digital delay or a delay of 2 to 8 U2BRG count source clock cycles selectable • Clock phase setting With or without clock delay selectable

NOTES:
1. When an external clock is selected, the conditions must be met while the external clock is in the high state.
2. If an overrun error occurs, bits 8 to 0 in the UiRB register (i = 0 to 2) are undefined. The IR bit in the S2RIC register remains unchanged.



This diagram applies to the case where bits SMD2 to SMD0 in the U2MR register is set to 010_2 and the IICM bit in the U2SMR register is set to 1.

IICM: Bit in the U2SMR register
IICM2, SWC, ALS, SWC2, SDHI: Bits in the U2SMR2 register
STSPSEL, ACKD, ACKC: Bits in the U2SMR4 register

NOTE:
1. If the IICM bit is set to 1, the pin can be read even when the PD7_1 bit is set to 1 (output mode).

Figure 14.22 I^2C bus Mode Block Diagram

Table 14.11 Registers to Be Used and Settings in I^2C bus Mode (1) (Continued)

Register	Bit	Function	
		Master	Slave
U2TB	0 to 7	Set transmission data	Set transmission data
U2RB(1)	0 to 7	Reception data can be read	Reception data can be read
	8	ACK or NACK is set in this bit	ACK or NACK is set in this bit
	ABT	Arbitration lost detection flag	Invalid
	OER	Overrun error flag	Overrun error flag
U2BRG	0 to 7	Set a transfer rate	Invalid
U2MR(1)	SMD2 to SMD0	Set to 010_2	Set to 010_2
	CKDIR	Set to 0	Set to 1
	IOPOL	Set to 0	Set to 0
U2C0	CLK1, CLK0	Select the count source for the U2BRG register	Invalid
	CRS	Invalid because CRD = 1	Invalid because CRD = 1
	TXEPT	Transmit buffer empty flag	Transmit buffer empty flag
	CRD	Set to 1	Set to 1
	NCH	Set to 1	Set to 1
	CKPOL	Set to 0	Set to 0
	UFORM	Set to 1	Set to 1
U2C1	TE	Set this bit to 1 to enable transmission	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception	Set this bit to 1 to enable reception
	RI	Reception complete flag	Reception complete flag
	U2IRS	Invalid	Invalid
	U2RRM, U2LCH, U2ERE	Set to 0	Set to 0
U2SMR	IICM	Set to 1	Set to 1
	ABC	Select the timing at which arbitration-lost is detected	Invalid
	BBS	Bus busy flag	Bus busy flag
	3 to 7	Set to 0	Set to 0
U2SMR2	IICM2	Refer to **Table 14.13**	Refer to **Table 14.13**
	CSC	Set this bit to 1 to enable clock synchronization	Set to 0
	SWC	Set this bit to 1 to have SCL_2 output fixed to L at the falling edge of the 9th bit of clock	Set this bit to 1 to have SCL_2 output fixed to “L” at the falling edge of the 9^{th} bit of clock
	ALS	Set this bit to 1 to have SDA_2 output stopped when arbitration-lost is detected	Set to 0
	STAC	Set to 0	Set this bit to 1 to initialize UART2 at start condition detection
	SWC2	Set this bit to 1 to have SCL_2 output forcibly pulled low	Set this bit to 1 to have SCL_2 output forcibly pulled low
	SDHI	Set this bit to 1 to disable SDA_2 output	Set this bit to 1 to disable SDA_2 output
	7	Set to 0	Set to 0
U2SMR3	0, 2, 4 and NODC	Set to 0	Set to 0
	CKPH	Refer to **Table 14.13**	Refer to **Table 14.13**
	DL2 to DL0	Set the amount of SDA_2 digital delay	Set the amount of SDA_2 digital delay

NOTE:

1. Not all bits in the register are described above. Set those bits to 0 when writing to the registers in I^2C bus mode.

Table 14.12 Registers to Be Used and Settings in I^2C bus Mode (2) (Continued)

Register	Bit	Function	
		Master	Slave
U2SMR4	STAREQ	Set this bit to 1 to generate start condition	Set to 0
	RSTAREQ	Set this bit to 1 to generate restart condition	Set to 0
	STPREQ	Set this bit to 1 to generate stop condition	Set to 0
	STSPSEL	Set this bit to 1 to output each condition	Set to 0
	ACKD	Select ACK or NACK	Select ACK or NACK
	ACKC	Set this bit to 1 to output ACK data	Set this bit to 1 to output ACK data
	SCLHI	Set this bit to 1 to have SCL_2 output stopped when stop condition is detected	Set to 0
	SWC9	Set to 0	Set this bit to 1 to set the SCL_2 to "L" hold at the falling edge of the 9th bit of clock

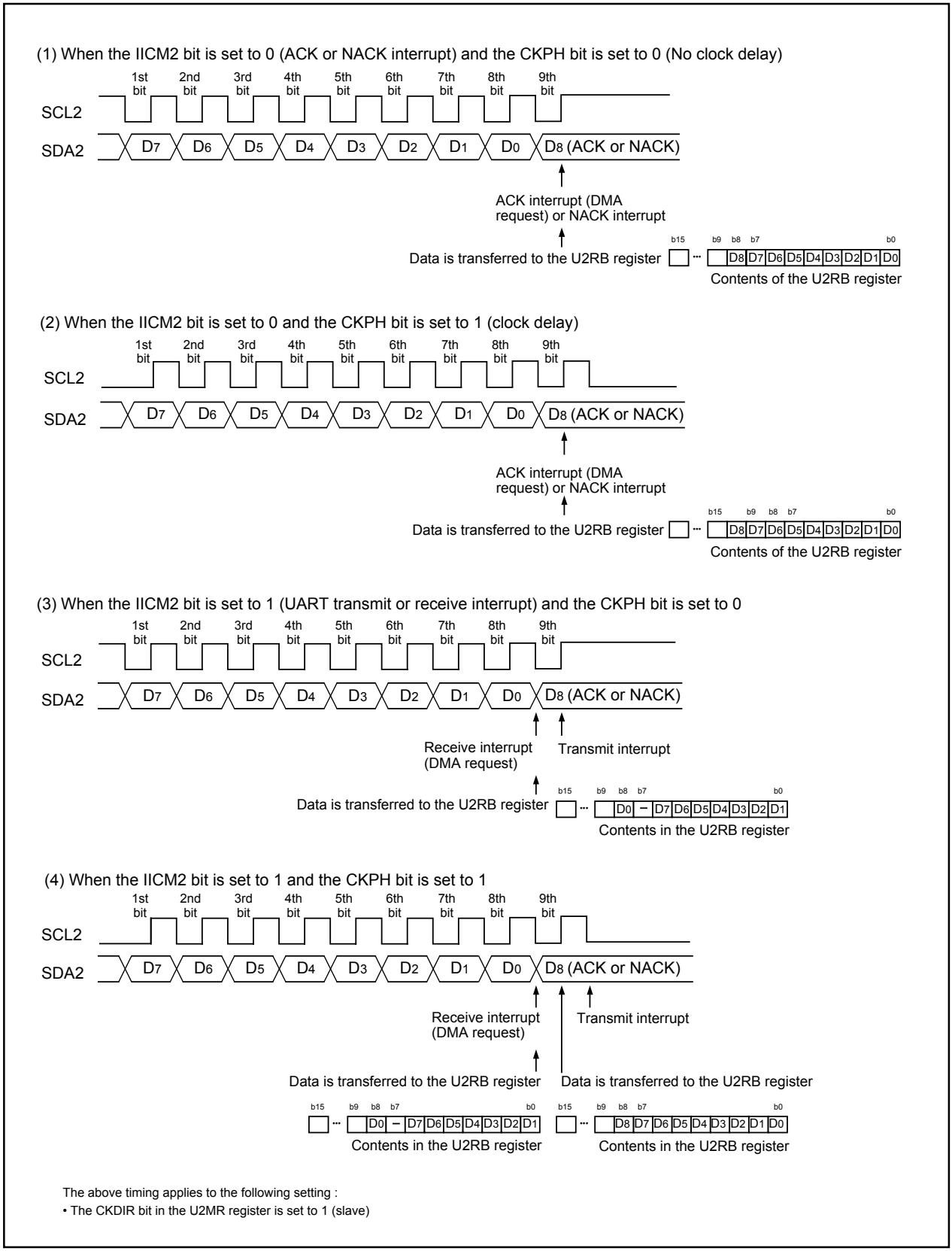
NOTE:

1: Not all bits in the register are described above. Set those bits to 0 when writing to the registers in I^2C bus mode.

Table 14.13 I^2C bus Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0 = 001₂, IICM = 0)	I^2C bus mode (SMD2 to SMD0 = 010₂, 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 number 10 (1) (See **Fig.14.23**)	–	Start condition detection or stop condition detection (Refer to **Table 14.14**)			
Interrupt source for number 15 (1) (See **Fig.14.23**)	UART2 transmit operation - transmit operation is started or completed (selected by U2IRS	No acknowledgment detection (NACK) - at the rising edge of 9th bit of SCL2		UART2 transmit operation - at the rising edge of 9th bit of SCL2	UART2 transmit operation - at the next falling edge after the 9th bit of SCL2
Interrupt source for number 16 (1) (See **Fig.14.23**)	UART2 receive timing - when 8th bit is received, CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) - at the rising edge of 9th bit of SCL2		UART2 receive operation - at the falling edge of 9th bit of SCL2	
Data transfer timing from the UART receive shift register to the U2RB register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	At the rising edge of 9th bit of SCL2		Falling edge of 9th bit of SCL2	Falling edge and rising edge of 9th bit of SCL2
UART2 transmit output delay	No delay	Delay			
Function of P70	TxD2 output	SDA2 input and output			
Function of P71	RxD2 input	SCL2 input and output			
Function of P72	Select CLK2 input or output	- (Not used in I^2C bus mode)			
Noise filter width	15ns	200n			
Reading RxD2, SCL2 pin levels	Can be read if the corresponding port direction bit is set to 0	Can be read regardless of the corresponding port direction bit			
Default value of TxD2, SDA2 output	CKPOL = 0 (H) CKPOL = 1 (L)	Value set in the port register before entering I^2C bus mode(1)			
SCL2 default and end values	–	Acknowledgment detection (ACK)		UART2 receive operation - at the falling edge of 9th bit of SCL2	
DMA1 source (See **Fig.14.23**)	UART2 receive operation	H	L	H	L
Storing receive data	1st to 8th bits are stored into bits 7 to 0 in the U2RB register	1st to 8th bits are stored into bits 7 to 0 in the U2RB register		1st to 7th bits are stored into the bit 6 to bit 0 in the U2RB register, with 8th bit stored in the bit 8 in the U2RB register	1st to 8th bits are stored into bits 7 to 0 in the U2RB register(3)
Reading receive data	The U2RB register status is read				Bit 6 to bit 0 in the U2RB register are read as bit 7 to bit 1. Bit 8 in the U2RB register is read as bit 0 (4)

NOTES:
1. If the interrupt source is changed, the IR bit in the interrupt control register for the changed interrupt may be set to 1 (interrupt requested). (Refer to “Interrupts” in Precautions.) If any of the following bits are changed, the interrupt source, the interrupt timing, etc. will be changed also. Therefore, set the IR bit to 0 (interrupt not requested) after those bits are changed.
   Bits SMD2 to SMD0 in the U2MR register, the IICM bit in the U2SMR register, the IICM2 bit in the U2SMR2 register, the CKPH bit in the U2SMR3 register
2. Set the default value of the SDA2 output when bits SMD2 to SMD0 in the U2MR register are set to 000₂ (serial I/O disabled).
3. Second data transfer to the U2RB register (at the rising edge of the ninth bit of SCL2)
4. First data transfer to the U2RB register (at falling edge of the ninth bit of SCL2)



The above timing applies to the following setting :
• The CKDIR bit in the U2MR register is set to 1 (slave)

Figure 14.23 Transfer to U2RB Register and Interrupt Timing

14.1.3.1 Detection of Start and Stop Condition

Whether a start or a stop condition has been detected is determined.

A start condition-detected interrupt request is generated when the SDA2 pin changes state from high to low while the SCL2 pin is in the high state. A stop condition-detected interrupt request is generated when the SDA2 pin changes state from low to high while the SCL2 pin is in the high state.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the BBS bit in the U2SMR register to determine which interrupt source is requesting the interrupt.

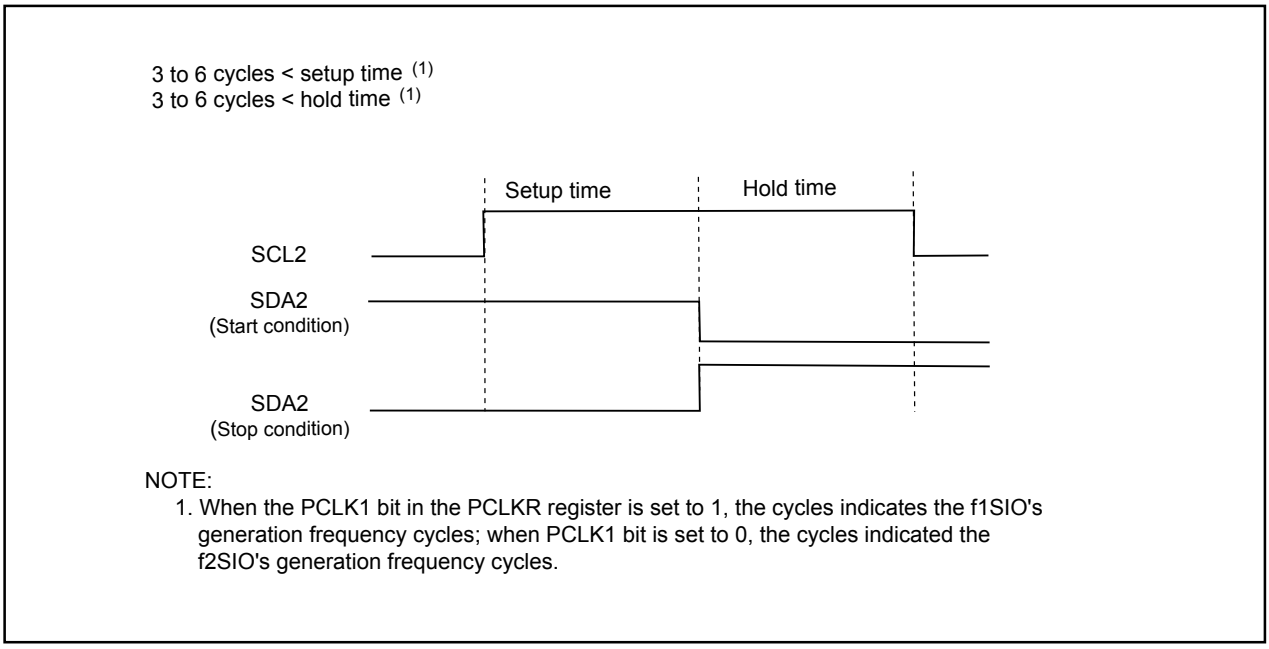


Figure 14.24 Detection of Start and Stop Condition

14.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the U2SMR4 register to 1 (start).

A restart condition is generated by setting the RSTAREQ bit in the U2SMR4 register to 1 (start).

A stop condition is generated by setting the STPREQ bit in the U2SMR4 register to 1 (start).

The output procedure is described below.

(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to 1 (start).

(2) Set the STSPSEL bit in the U2SMR4 register to 1 (output).

Make sure that no interrupts or DMA transfers will occur between (1) and (2).

The function of the STSPSEL bit is shown in **Table 14.14** and **Figure 14.25**.

Table 14.14 STSPSEL Bit Functions

Function	STSPSEL = 0	STSPSEL = 1
Output of SCL2 and SDA2 pins	Output transfer clock and data/ Program with a port determines how the start condition or stop condition is output	The STAREQ, RSTAREQ and STPREQ bit determine how the start condition or stop condition is output
Start/stop condition interrupt request generation timing	Start/stop condition are detected	Start/stop condition generation are completed

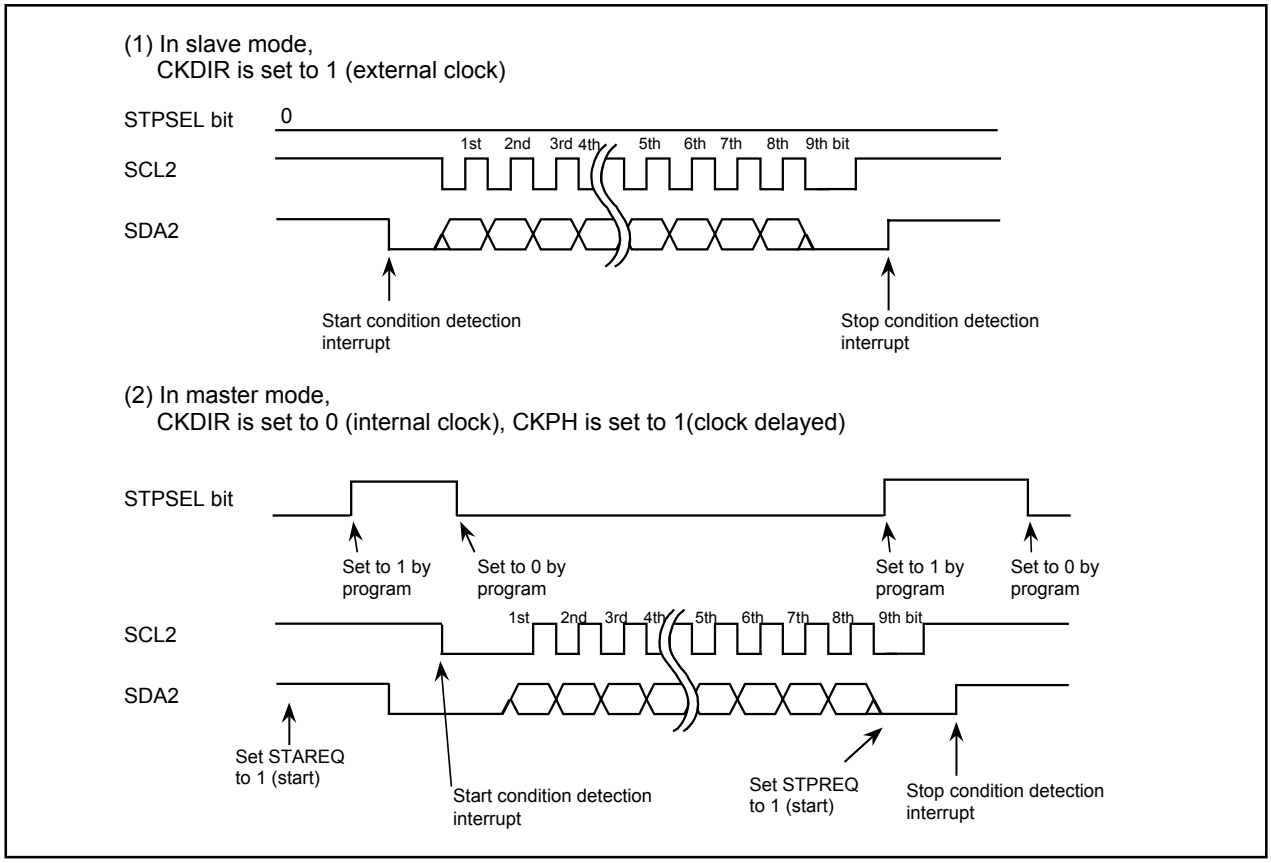


Figure 14.25 STSPSEL Bit Functions

14.1.3.3 Arbitration

Unmatching of the transmit data and SDA2 pin input data is checked synchronously with the rising edge of SCL2. Use the ABC bit in the U2SMR register to select the timing at which the ABT bit in the U2RB register is updated. If the ABC bit is set to 0 (updated bitwise), the ABT bit is set to 1 at the same time unmatching is detected during check, and is cleared to 0 when not detected. In cases when the ABC bit is set to 1, if unmatching is detected even once during check, the ABT bit is set to 1 (unmatching detected) at the falling edge of the clock pulse of 9th bit. If the ABT bit needs to be updated bytewise, clear the ABT bit to 0 (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the ALS bit in the U2SMR2 register to 1 (SDA2 output stop enabled) causes arbitration-lost to occur, in which case the SDA2 pin is placed in the high-impedance state at the same time the ABT bit is set to 1 (unmatching detected).

14.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in **Figure 14.25**.

The CSC bit in the U2SMR2 register is used to synchronize the internally generated clock (internal SCL2) and an external clock supplied to the SCL_2 pin. In cases when the CSC bit is set to 1 (clock synchronization enabled), if a falling edge on the SCL_2 pin is detected while the internal SCL_2 is high, the internal SCL_2 goes low, at which time the U2BRG register value is reloaded with and starts counting in the low-level interval. If the internal SCL_2 changes state from low to high while the SCL_2 pin is low, counting stops, and when the SCL_2 pin goes high, counting restarts.

In this way, the UART2 transfer clock is comprised of the logical product of the internal SCL_2 and SCL_2 pin signal. The transfer clock works from a half period before the falling edge of the internal SCL_2 1st bit to the rising edge of the 9^{th} bit. To use this function, select an internal clock for the transfer clock.

The SWC bit in the U2SMR2 register allows to select whether the SCL_2 pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the SCLHI bit in the U2SMR4 register is set to 1 (enabled), SCL_2 output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the U2SMR2 register is set to 1 (0 output) makes it possible to forcibly output a low-level signal from the SCL_2 pin even while sending or receiving data. Clearing the SWC2 bit to 0 (transfer clock) allows the transfer clock to be output from or supplied to the SCL_2 pin, instead of outputting a low-level signal.

If the SWC9 bit in the U2SMR4 register is set to 1 (SCL_2 hold low enabled) when the CKPH bit in the U2SMR3 register is set to 1, the SCL_2 pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit to 0 (SCL_2 hold low disabled) frees the SCL_2 pin from low-level output.

14.1.3.5 SDA Output

The data written to the bit 7 to bit 0 (D_7 to D_0) in the U2TB register is sequentially output beginning with D_7. The ninth bit (D_8) is ACK or NACK.

The initial value of SDA_2 transmit output can only be set when IICM is set to 1 (I^2C bus mode) and bits SMD2 to SMD0 in the U2MR register is set to 000_2 (serial I/O disabled).

Bits DL2 to DL0 in the U2SMR3 register allow to add no delays or a delay of 2 to 8 U2BRG count source clock cycles to SDA_2 output.

Setting the SDHI bit in the U2SMR2 register to 1 (SDA_2 output disabled) forcibly places the SDA_2 pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UART2 transfer clock. This is because the ABT bit may inadvertently be set to 1 (detected).

14.1.3.6 SDA Input

When the IICM2 bit is set to 0, the 1st to 8th bits (D_7 to D_0) in the received data are stored in bits 7 to 0 in the U2RB register. The 9th bit (D_8) is ACK or NACK.

When the IICM2 bit is set to 1, the 1st to 7th bits (D_7 to D_1) in the received data are stored in the bit 6 to bit 0 in the U2RB register and the 8th bit (D_0) is stored in the bit 8 in the U2RB register. Even when the IICM2 bit is set to 1, providing the CKPH bit is set to 1, the same data as when the IICM2 bit is set to 0 can be read out by reading the U2RB register after the rising edge of the corresponding clock pulse of 9th bit.

14.1.3.7 ACK and NACK

If the STSPSEL bit in the U2SMR4 register is set to 0 (start and stop conditions not generated) and the ACKC bit in the U2SMR4 register is set to 1 (ACK data output), the value of the ACKD bit in the U2SMR4 register is output from the SDA_2 pin.

If the IICM2 bit is set to 0, a NACK interrupt request is generated if the SDA_2 pin remains high at the rising edge of the 9th bit of transmit clock pulse. An ACK interrupt request is generated if the SDA_2 pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACK2 is selected for the cause of DMA1 request, a DMA transfer can be activated by detection of an acknowledge.

14.1.3.8 Initialization of Transmission/Reception

If a start condition is detected while the STAC bit is set to 1 (UART2 initialization enabled), the serial I/O operates as described below.

- The transmit shift register is initialized, and the content of the U2TB register is transferred to the transmit shift register. In this way, the serial I/O starts sending data synchronously with the next clock pulse applied. However, the UART2 output value does not change state and remains the same as when a start condition was detected until the first bit in the data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial I/O starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to 1 (SCL_2 wait output enabled). Consequently, the SCL_2 pin is pulled low at the falling edge of the ninth clock pulse.

Note that when UART2 transmission/reception is started using this function, the TI does not change state. Note also that when using this function, the selected transfer clock should be an external clock.

14.1.4 Special Mode 2 (UART2)

Multiple slaves can be serially communicated from one master. Transfer clock polarity and phase are selectable. **Table 14.15** lists the specifications of Special Mode 2. **Table 14.16** lists the registers used in Special Mode 2 and the register values set. **Figure 14.26** shows communication control example for Special Mode 2.

Table 14.15 Special Mode 2 Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• Master mode the CKDIR bit in the U2MR register is set to 0 (internal clock) : fj/ (2(n+1)) fj = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value in the U2BRG register 00_{16} to FF_{16} • Slave mode CKDIR bit is set to 1 (external clock selected) : Input from CLK2 pin
Transmit/receive control	Controlled by input/output ports
Transmission start condition	• Before transmission can start, the following requirements must be met (1) – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in U2TB register)
Reception start condition	• Before reception can start, the following requirements must be met (1) – The RE bit in the U2C1 register is set to 1 (reception enabled) – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in the U2TB register)
Interrupt request generation timing	• For transmission, one of the following conditions can be selected – The U2IRS bit in the U2C1 register is set to 0 (transmit buffer empty): when transferring data from the U2TB register to the UART2 transmit register (at start of transmission) – The U2IRS bit is set to 1 (transfer completed): when the serial I/O finished sending data from the UART2 transmit register • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error detection	• Overrun error (2) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the 7th bit in the the next data
Select function	• Clock phase setting Selectable from four combinations of transfer clock polarities and phases

NOTES:

1. When an external clock is selected, the conditions must be met while if the CKPOL bit in the U2C0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the U2C0 register is set to 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.
2. If an overrun error occurs, bits 8 to 0 in the U2RB register are undefined. The IR bit in the S2RIC register remains unchanged.

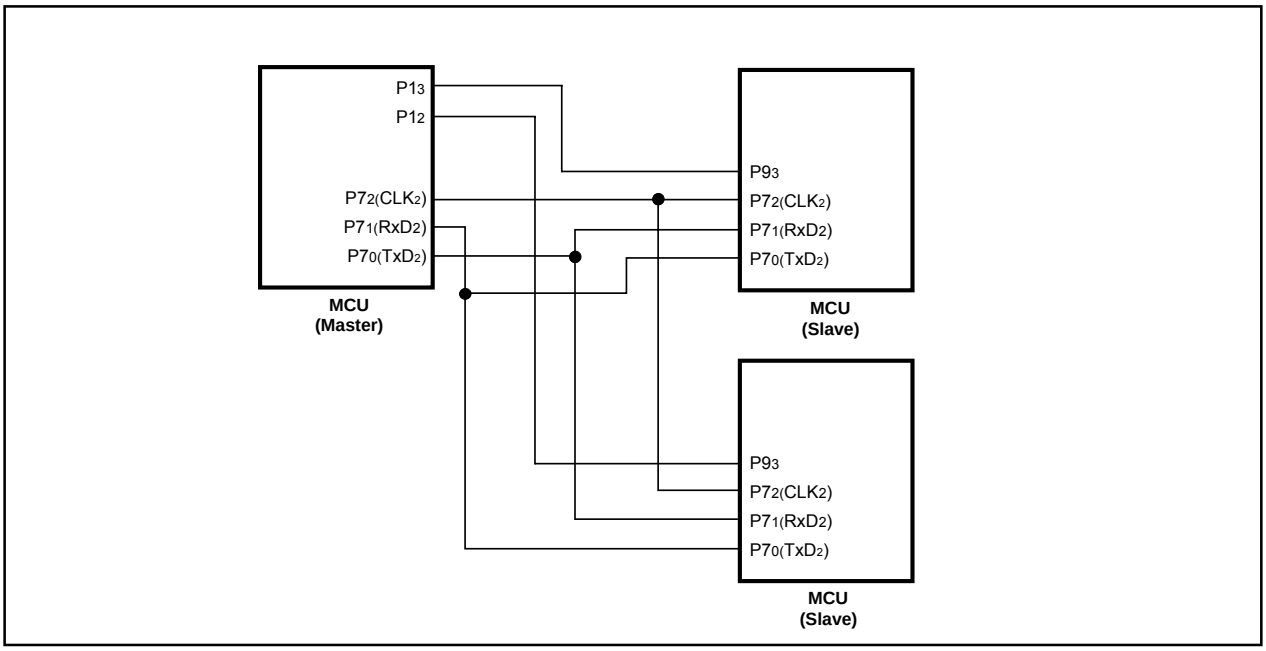


Figure 14.26 Serial Bus Communication Control Example (UART2)

Table 14.16 Registers to Be Used and Settings in Special Mode 2

Register	Bit	Function
U2TB(1)	0 to 7	Set transmission data
U2RB(1)	0 to 7	Reception data can be read
	OER	Overrun error flag
U2BRG	0 to 7	Set a transfer rate
U2MR(1)	SMD2 to SMD0	Set to 001$_2$
	CKDIR	Set this bit to 0 for master mode or 1 for slave mode
	IOPOL	Set to 0
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRD is set to 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TxD2 pin output format
	CKPOL	Clock phases can be set in combination with the CKPH bit in the U2SMR3 register
	UFORM	Select the LSB first or MSB first
U2C1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS	Select UART2 transmit interrupt cause
	U2RRM, U2LCH, U2ERE	Set to 0
U2SMR	0 to 7	Set to 0
U2SMR2	0 to 7	Set to 0
U2SMR3	CKPH	Clock phases can be set in combination with the CKPOL bit in the U2C0 register
	NODC	Set to 0
	0, 2, 4 to 7	Set to 0
U2SMR4	0 to 7	Set to 0

NOTE:
1. Not all bits in the registers are described above. Set those bits to 0 when writing to the registers in Special Mode 2.

14.1.4.1 Clock Phase Setting Function

One of four combinations of transfer clock phases and polarities can be selected using the CKPH bit in the U2SMR3 register and the CKPOL bit in the U2C0 register.

Make sure the transfer clock polarity and phase are the same for the master and slave to communicate.

14.1.4.1.1 Master (Internal Clock)

Figure 14.27 shows the transmission and reception timing in master (internal clock).

14.1.4.1.2 Slave (External Clock)

Figure 14.28 shows the transmission and reception timing (CKPH=0) in slave (external clock) while **Figure 14.29** shows the transmission and reception timing (CKPH=1) in slave (external clock).

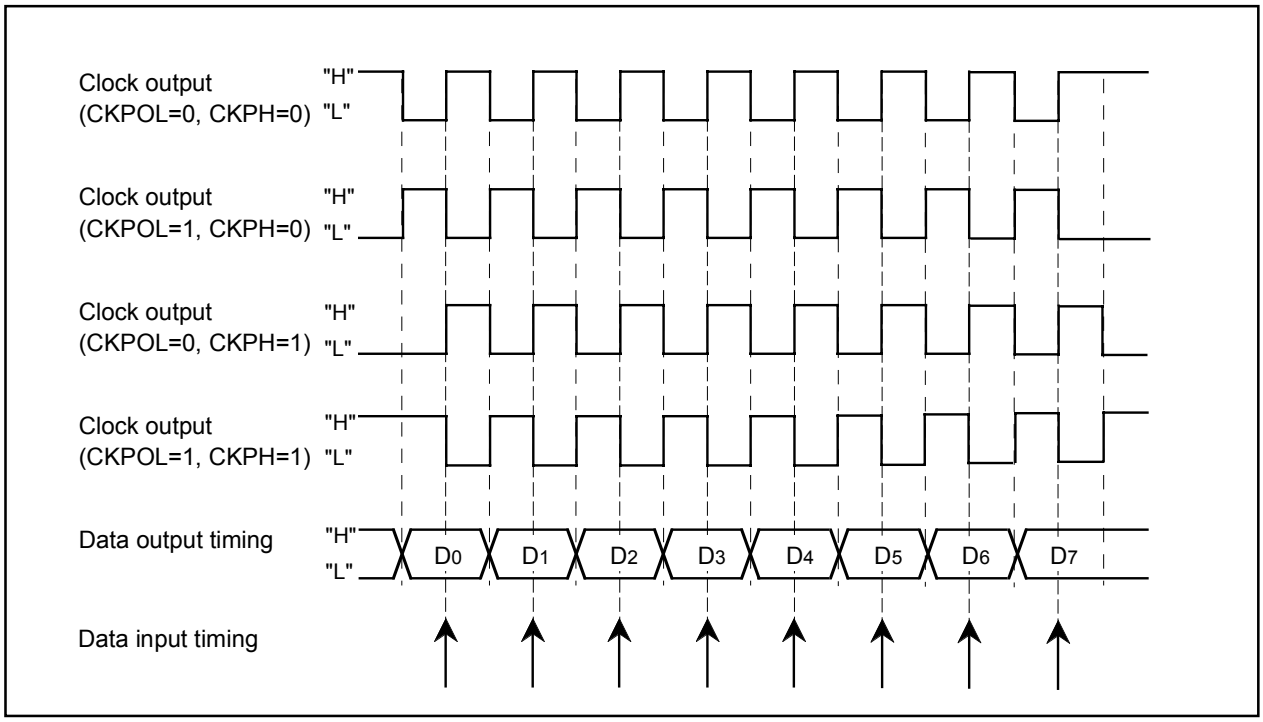


Figure 14.27 Transmission and Reception Timing in Master Mode (Internal Clock)

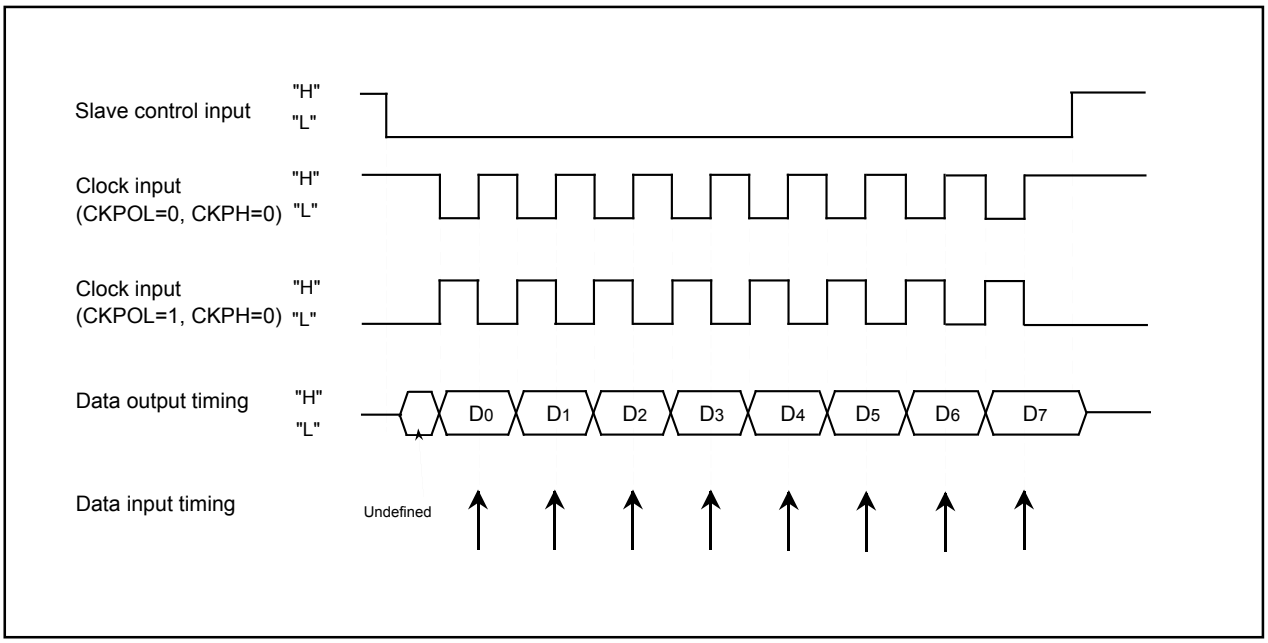


Figure 14.28 Transmission and Reception Timing (CKPH=0) in Slave Mode (External Clock)

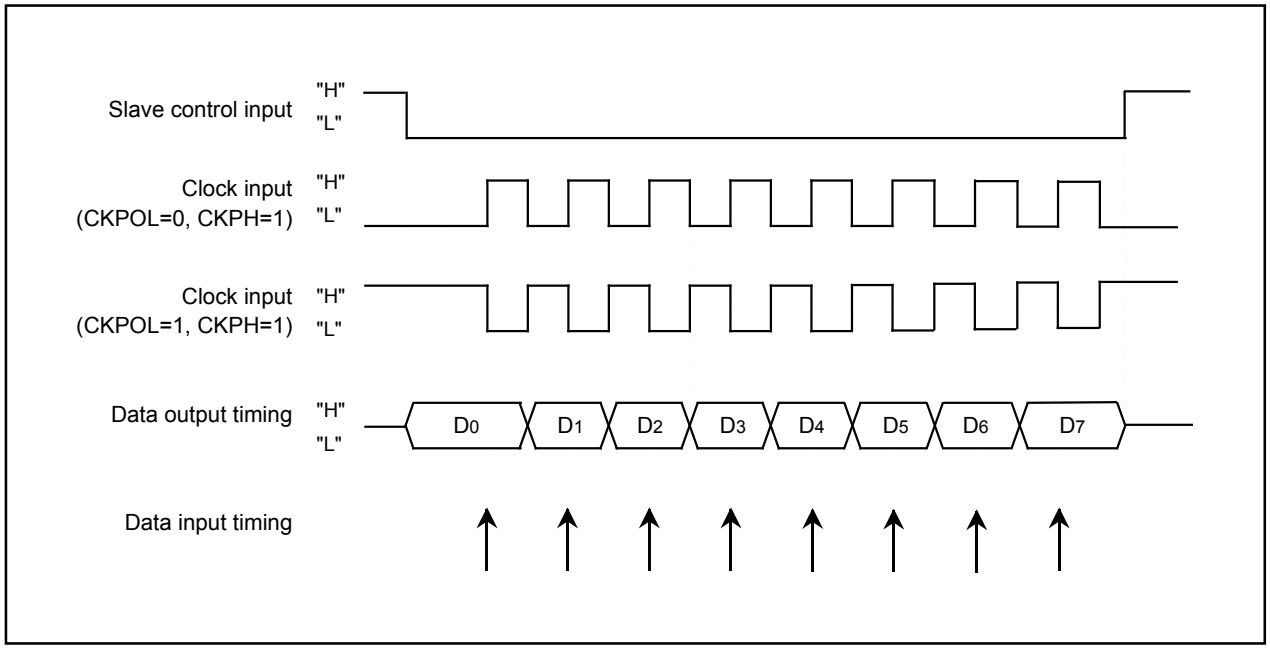


Figure 14.29 Transmission and Reception Timing (CKPH=1) in Slave Mode (External Clock)

14.1.5 Special Mode 3 (IEBus mode)(UART2)

In this mode, one bit in the IEBus is approximated with one byte of UART mode waveform.

Table 14.17 lists the registers used in IEBus mode and the register values set. **Figure 14.30** shows the functions of bus collision detect function related bits.

If the TxD2 pin output level and RxD2 pin input level do not match, a UART2 bus collision detect interrupt request is generated.

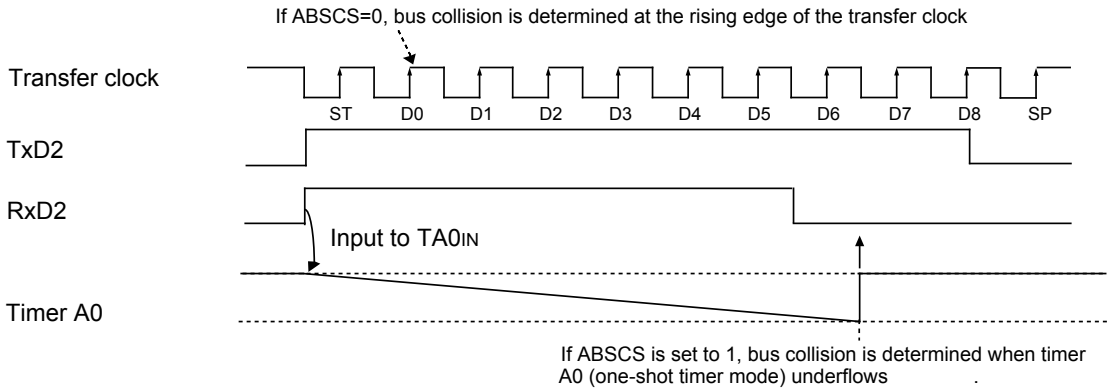
Table 14.17 Registers to Be Used and Settings in IEBus Mode

Register	Bit	Function
U2TB	0 to 8	Set transmission data
U2RB(1)	0 to 8	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set a transfer rate
U2MR	SMD2 to SMD0	Set to 110_2
	CKDIR	Select the internal clock or external clock
	STPS	Set to 0
	PRY	Invalid because PRYE is set to 0
	PRYE	Set to 0
	IOPOL	Select the TxD/RxD input/output polarity
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRDis set to 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TxD2 pin output mode
	CKPOL	Set to 0
	UFORM	Set to 0
U2C1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS	Select the source of UART2 transmit interrupt
	U2RRM, U2LCH, U2ERE	Set to 0
U2SMR	0 to 3, 7	Set to 0
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to 1 to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
U2SMR2	0 to 7	Set to 0
U2SMR3	0 to 7	Set to 0
U2SMR4	0 to 7	Set to 0

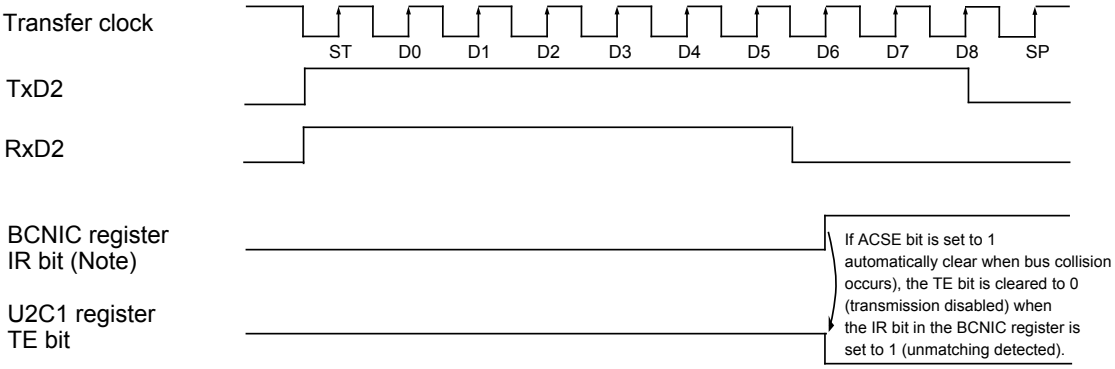
NOTE:

1. Not all register bits are described above. Set those bits to 0 when writing to the registers in IEBus mode.

(1) The ABSCS bit in the U2SMR register (bus collision detect sampling clock select)

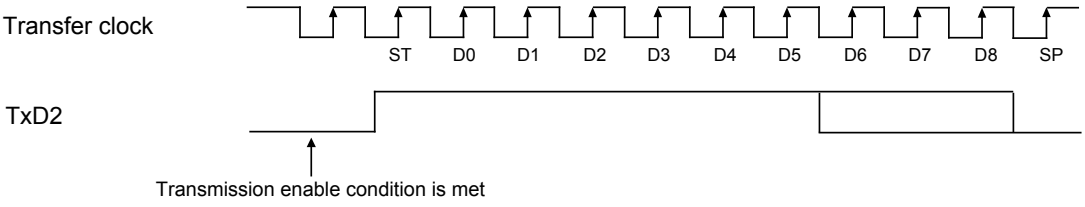


(2) The ACSE bit in the U2SMR register (auto clear of transmit enable bit)

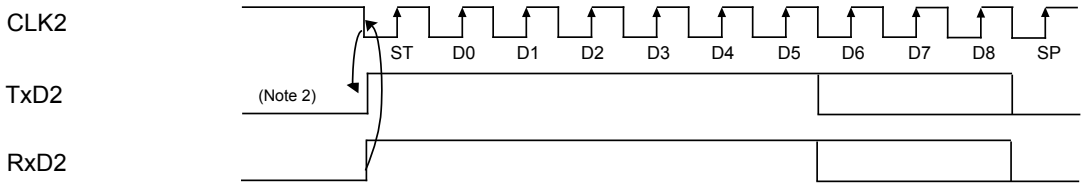


(3) The SSS bit in the U2SMR register (Transmit start condition select)

If SSS bit is set to 0, the serial I/O starts sending data one transfer clock cycle after the transmission enable condition is met.



If SSS bit = 1, the serial I/O starts sending data at the rising edge (Note 1) of RxD2



NOTES:
1: The falling edge of RxD2 when the IOPOL is set to 0; the rising edge of RxD2 when the IOPOL is set to 1.
2: The transmit condition must be met before the falling edge (Note 1) of RxD.

This diagram applies to the case where the IOPOL is set to 1 (reversed)

Figure 14.30 Bus Collision Detect Function-Related Bits

14.1.6 Special Mode 4 (SIM Mode) (UART2)

Based on UART mode, this is an SIM interface compatible mode. Direct and inverse formats can be implemented, and this mode allows output of a low from the TxD2 pin when a parity error is detected.

Tables 14.18 lists the specifications of SIM mode. **Table 14.19** lists the registers used in the SIM mode and the register values set.

Table 14.18 SIM Mode Specifications

Item	Specification
Transfer data format	• Direct format • Inverse format
Transfer clock	• The CKDIR bit in the U2MR register is set to 0 (internal clock) : fi/ (16(n+1)) fi = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO}. n: Setting value of U2BRG register 00_{16} to FF_{16} • The CKDIR bit is set to 1 (external clock) : f_{EXT}/16(n+1) f_{EXT}: Input from CLK_2 pin. n: Setting value of U2BRG register 00_{16} to FF_{16}
Transmission start condition	• Before transmission can start, the following requirements must be met – The TE bit in the U2C1 register is set to 1 (transmission enabled) – The TI bit in the U2C1 register is set to 0 (data present in U2TB register)
Reception start condition	• Before reception can start, the following requirements must be met – The RE bit in the U2C1 register is set to 1 (reception enabled) – Start bit detection
Interrupt request generation timing (2) bit =1)	• For transmission When the serial I/O finished sending data from the U2TB transfer register (U2IRS • For reception When transferring data from the UART2 receive register to the U2RB register (at completion of reception)
Error detection	• Overrun error (1) This error occurs if the serial I/O started receiving the next data before reading the U2RB register and received the bit one before the last stop bit in the the next data • Framing error This error occurs when the number of stop bits set is not detected • Parity error During reception, if a parity error is detected, parity error signal is output from the TxD_2 pin. During transmission, a parity error is detected by the level of input to the RxD_2 pin when a transmission interrupt occurs • Error sum flag This flag is set to 1 when any of the overrun, framing, and parity errors is encountered

NOTES:

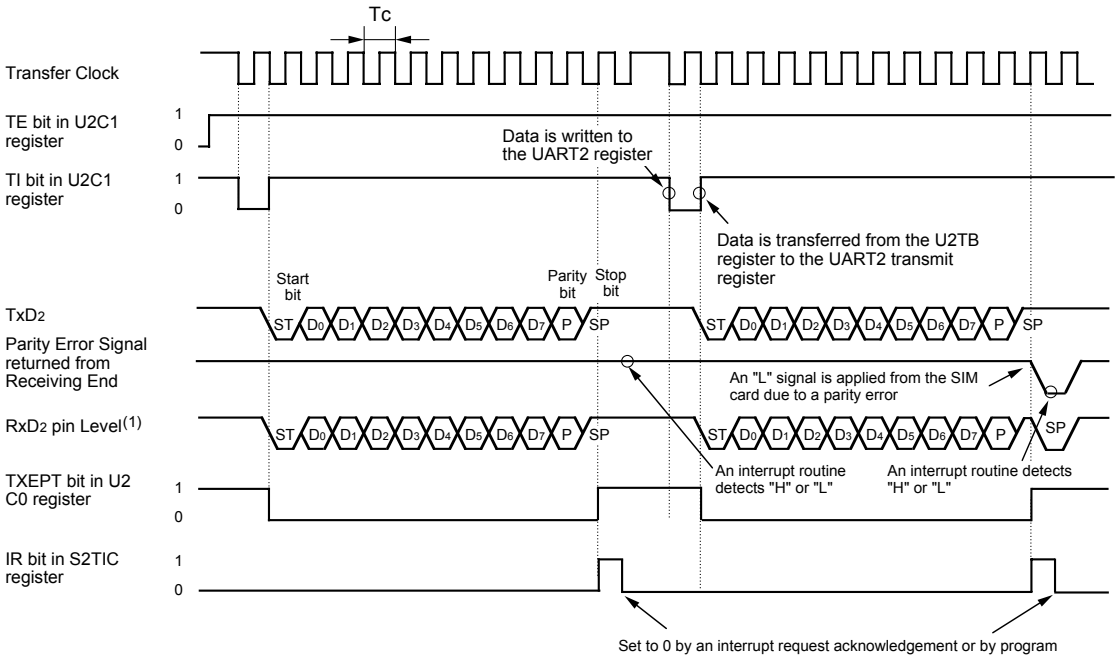
1. If an overrun error occurs, bits 8 to 0 in the U2RB register are undefined. The IR bit in the S2RIC register remains unchanged.
2. A transmit interrupt request is generated by setting the U2IRS bit in the U2C1 register to 1 (transmission complete) and U2ERE bit to 1 (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to 0 (no interrupt request) after setting these bits.

Table 14.19 Registers to Be Used and Settings in SIM Mode

Register	Bit	Function
U2TB(1)	0 to 7	Set transmission data
U2RB(1)	0 to 7	Reception data can be read
	OER,FER,PER,SUM	Error flag
U2BRG	0 to 7	Set a transfer rate
U2MR	SMD2 to SMD0	Set to 101_2
	CKDIR	Select the internal clock or external clock
	STPS	Set to 0
	PRY	Set this bit to 1 for direct format or 0 for inverse format
	PRYE	Set to 1
	IOPOL	Set to 0
U2C0	CLK1, CLK0	Select the count source for the U2BRG register
	CRS	Invalid because CRDis set to 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Set to 0
	CKPOL	Set to 0
	UFORM	Set this bit to 0 for direct format or 1 for inverse format
U2C1	TE	Set this bit to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set this bit to 1 to enable reception
	RI	Reception complete flag
	U2IRS	Set to 1
	U2RRM	Set to 0
	U2LCH	Set this bit to 0 for direct format or 1 for inverse format
	U2ERE	Set to 1
U2SMR(1)	0 to 3	Set to 0
U2SMR2	0 to 7	Set to 0
U2SMR3	0 to 7	Set to 0
U2SMR4	0 to 7	Set to 0

NOTE:
1. Not all register bits are described above. Set those bits to 0 when writing to the registers in SIM mode.

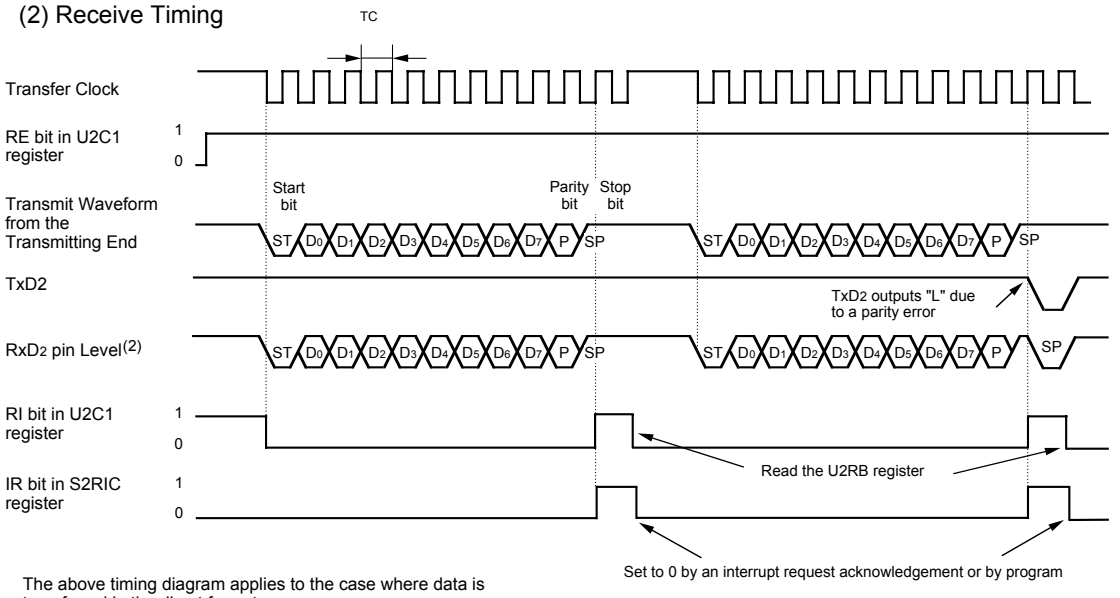
(1) Transmit Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of U2BRG count source (external clock)
n : value set to U2BRG

(2) Receive Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of U2BRG count source (external clock)
n : value set to U2BRG

NOTES:
1. Because TxD2 and RxD2 are connected, this is composite waveform consisting of the TxD2 output and the parity error signal sent back from receiver.
2. Because TxD2 and RxD2 are connected, this is composite waveform consisting of the transmitter's transmit waveform and the parity error signal received.

Figure 14.31 Transmit and Receive Timing in SIM Mode

Figure 14.32 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

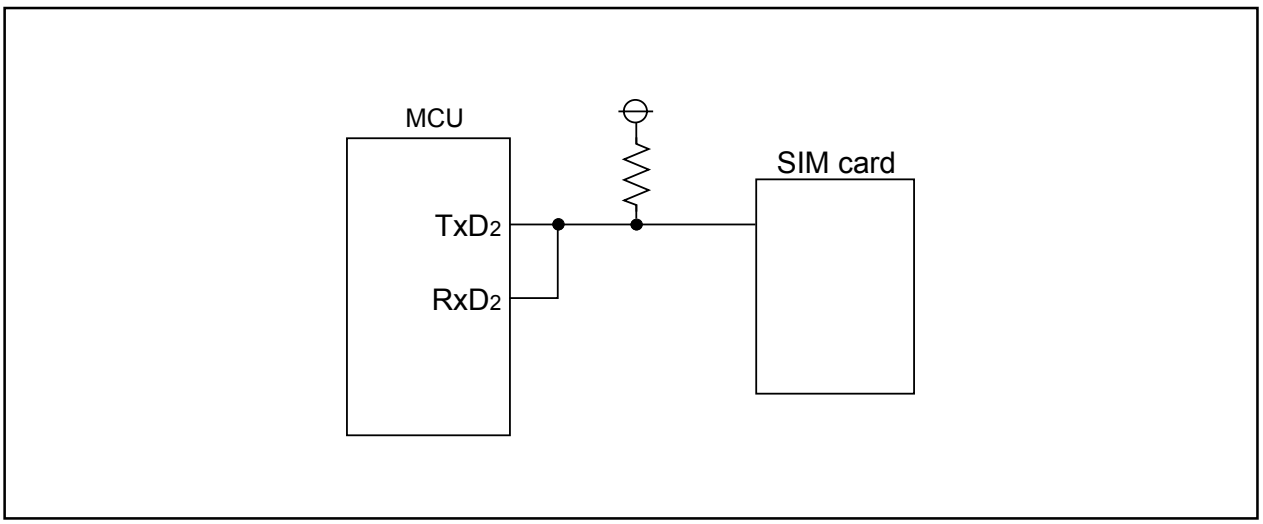


Figure 14.32 SIM Interface Connection

14.1.6.1 Parity Error Signal Output

The parity error signal is enabled by setting the U2ERE bit in theU2C1 register to 1.

• When receiving

The parity error signal is output when a parity error is detected while receiving data. This is achieved by pulling the TxD2 output low with the timing shown in **Figure 14.33**. If the R2RB register is read while outputting a parity error signal, the PER bit is cleared to 0 and at the same time the TxD2 output is returned high.

• When transmitting

A transmission-finished interrupt request is generated at the falling edge of the transfer clock pulse that immediately follows the stop bit. Therefore, whether a parity signal has been returned can be determined by reading the port that shares the RxD2 pin in a transmission-finished interrupt service routine.

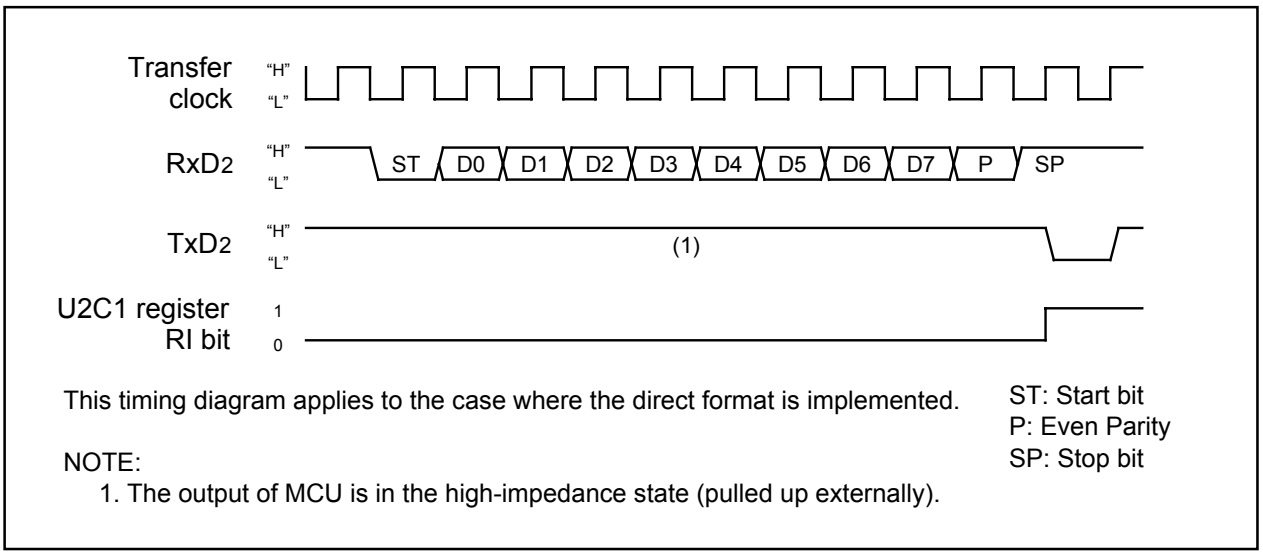


Figure 14.33 Parity Error Signal Output Timing

14.1.6.2 Format

• Direct Format

Set the PRY bit in the U2MR register to 1, the UFORM bit in U2C0 register to 0 and the U2LCH bit in U2C1 register to 0.

• Inverse Format

Set the PRY bit to 0, UFORM bit to 1 and U2LCH bit to 1.

Figure 14.34 shows the SIM interface format.

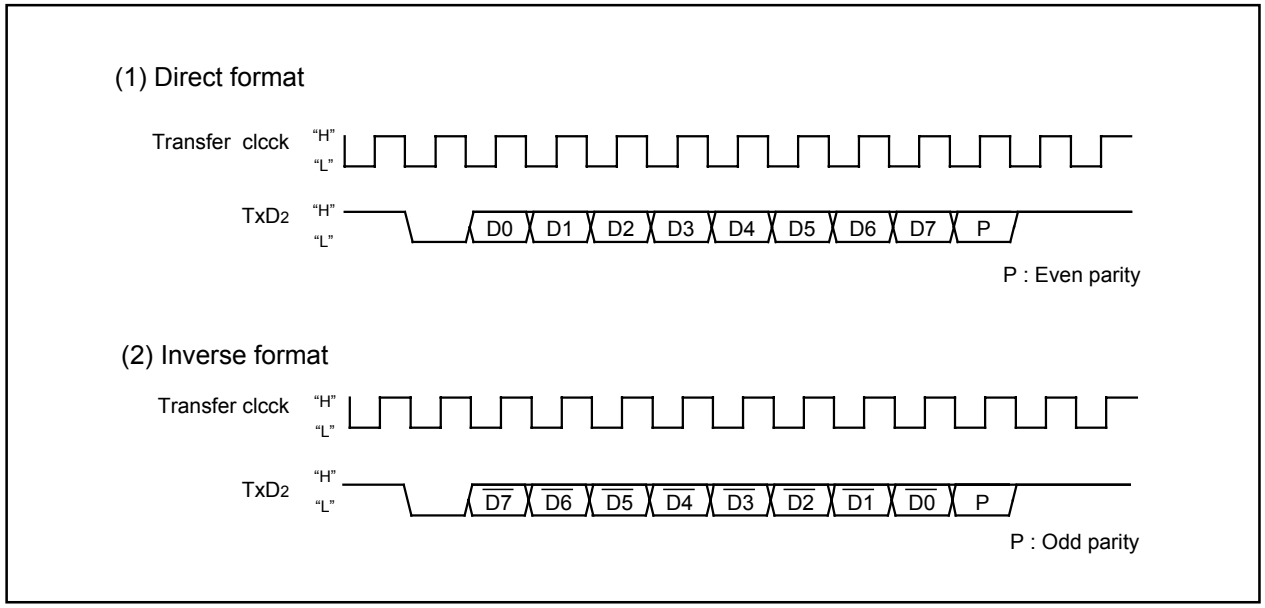


Figure 14.34 SIM Interface Format

14.2 SI/O3 and SI/O4

Note

The SI/O4 interrupt of peripheral function interrupt is not available in the 64-pin package.

SI/O3 and SI/O4 are exclusive clock-synchronous serial I/Os.

Figure 14.35 shows the block diagram of SI/O3 and SI/O4, and **Figure 14.36** shows the SI/O3 and SI/O4-related registers.

Table 14.20 shows the specifications of SI/O3 and SI/O4.

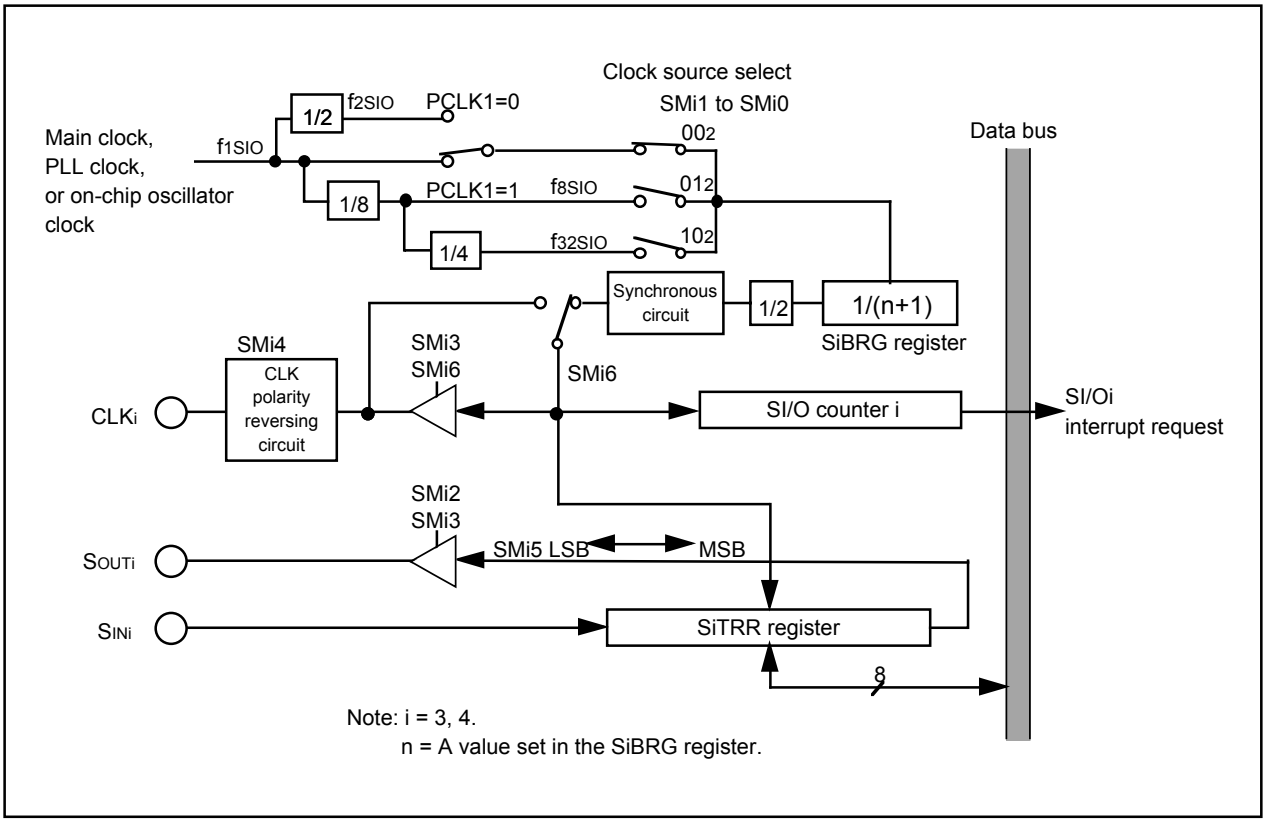


Figure 14.35 SI/O3 and SI/O4 Block Diagram

SI/Oi Control Register (i = 3, 4) (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
S3C	036216	01000000_2
S4C	036616	01000000_2

Bit Symbol	Bit Name	Function	
SMi0	Internal synchronous clock select bit (5)	b1 b0 0 0 : Selecting f_1 or f_2 0 1 : Selecting f_8 1 0 : Selecting f_{32} 1 1 : Do not set	RW
SMi1			RW
SMi2	SOUTi output disable bit (4)	0 : SOUTi output 1 : SOUTi output disable(high impedance)	RW
SMi3	S I/Oi port select bit	0 : Input/output port 1 : SOUTi output, CLKi function	RW
SMi4	CLK polarity selct bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	RW
SMi5	Transfer direction select bit	0 : LSB first 1 : MSB first	RW
SMi6	Synchronous clock select bit	0 : External clock (2) 1 : Internal clock (3)	RW
SMi7	SOUTi initial value set bit	Effective when the SMi3 is set to 0 0 : "L" output 1 : "H" output	RW

NOTES:
1. Set the S4C register by the next instruction after setting the PRC2 bit in the PRCR register to 1 (write enable).
2. Set the SMi3 bit to 1 and the corresponding port direction bit to 0 (input mode).
3. Set the SMi3 bit to 1 (SOUTi output, CLKi function) .
4. When the SMi2 bit is set to 1, the corresponding pin goes to high-impedance regardless of the function in use.
5. When the SMi1 and SMi0 bit settings are changed, set the SiBRG register .

SI/Oi Bit Rate Generation Register (i = 3, 4) (1, 2, 3)

b7	b0

Symbol	Address	After Reset
S3BRG	036316	Undefined
S4BRG	036716	Undefined

Description	Setting Range	RW
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	WO

NOTES:
1. Write to this register while serial I/O is neither transmitting or receiving.
2. Use MOV instruction to write to this register.
3. Set the SiBRG register after setting bits SMi1 and SMi0 in the SiC register.

SI/Oi Transmit/Receive Register (i = 3, 4) (1, 2)

b7	b0

Symbol	Address	After Reset
S3TRR	036016	Undefined
S4TRR	036416	Undefined

Description	RW
Transmission/reception starts by writing transmit data to this register. After transmission/reception completion, reception data can be read by reading this register.	RW

NOTES:
1. Write to this register while serial I/O is neither transmitting or receiving.
2. To receive data, set the corresponding port direction bit for SINi to 0 (input mode).

Figure 14.36 S3C and S4C Registers, S3BRG and S4BRG Registers, and S3TRR and S4TRR Registers

Table 14.20 SI/O3 and SI/O4 Specifications

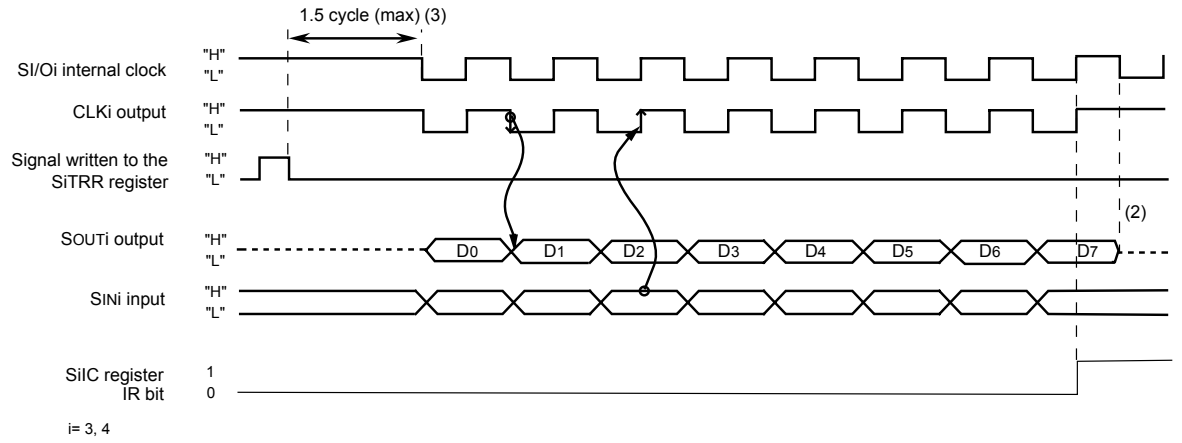
Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• The SMi6 bit in the SiC (i=3, 4) register is set to 1 (internal clock) : fj/ (2(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO. n=Setting value of SiBRG register 0016 to FF16. • SMi6 bit is set to 0 (external clock) : Input from CLKi pin (1)
Transmission/reception start condition	• Before transmission/reception can start, the following requirements must be met Write transmit data to the SiTRR register (2, 3)
Interrupt request generation timing	• When the SMi4 bit in the SiC register is set to 0 The rising edge of the last transfer clock pulse (4) • When SMi4 is set to 1 The falling edge of the last transfer clock pulse (4)
CLKi pin fucntion	I/O port, transfer clock input, transfer clock output
SOUTi pin function	I/O port, transmit data output, high-impedance
SINi pin function	I/O port, receive data input
Select function	• LSB first or MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Function for setting an SOUTi initial value set function When the SMi6 bit in the SiC register is set to 0 (external clock), the SOUTi pin output level while not tranmitting can be selected. • CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected.

NOTE:

1. To set the SMi6 bit in the SiC register to 0 (external clock), follow the procedure described below.
   - If the SMi4 bit in the SiC register is set to 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SMi7 bit in the SiC register.
   - If the SMi4 bit is set to 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMi7 bit.
   - Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock 2. Unlike UART0 to UART2, SI/Oi (i = 3 to 4) is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.
3. When the SMi6 bit in the SiC register is set to 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.
4. When the SMi6 bit in the SiC register is set to 1 (internal clock), the transfer clock stops in the high state if the SMi4 bit is set to 0, or stops in the low state if the SMi4 bit is set to 1.

14.2.1 SI/Oi Operation Timing

Figure 14.37 shows the SI/Oi operation timing

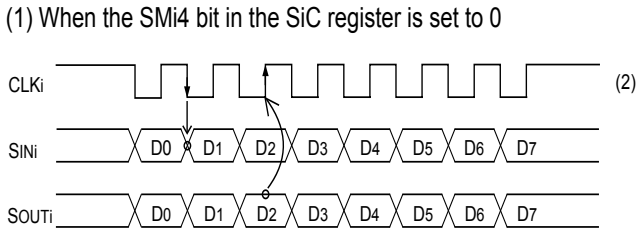


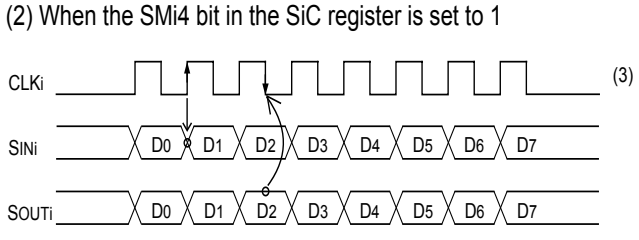
NOTES:
1. This diagram applies to the case where the SiC register bits are set as follows:
   SMi2 = 0 (SOUTi output), SMi3 = 1 (SOUTi output, CLKi function), SMi4 = 0 (transmit data output at the falling edge and receive data input at the rising edge of the transfer clock), SMi5 = 0 (LSB first) and SMi6 = 1 (internal clock)
2. When the SMi6 bit is set to 0 (internal clock), the SOUTi pin is placed in the high-impedance state after the transfer is completed.
3. If the SMi6 bit is set to 0 (internal clock), the serial I/O starts sending or receiving data a maximum of 1.5 transfer clock cycles after writing to the SiTRR register.

Figure 14.37 SI/Oi Operation Timing

14.2.2 CLK Polarity Selection

The the SMi4 bit in the SiC register allows selection of the polarity of the transfer clock. **Figure 14.38** shows the polarity of the transfer clock.





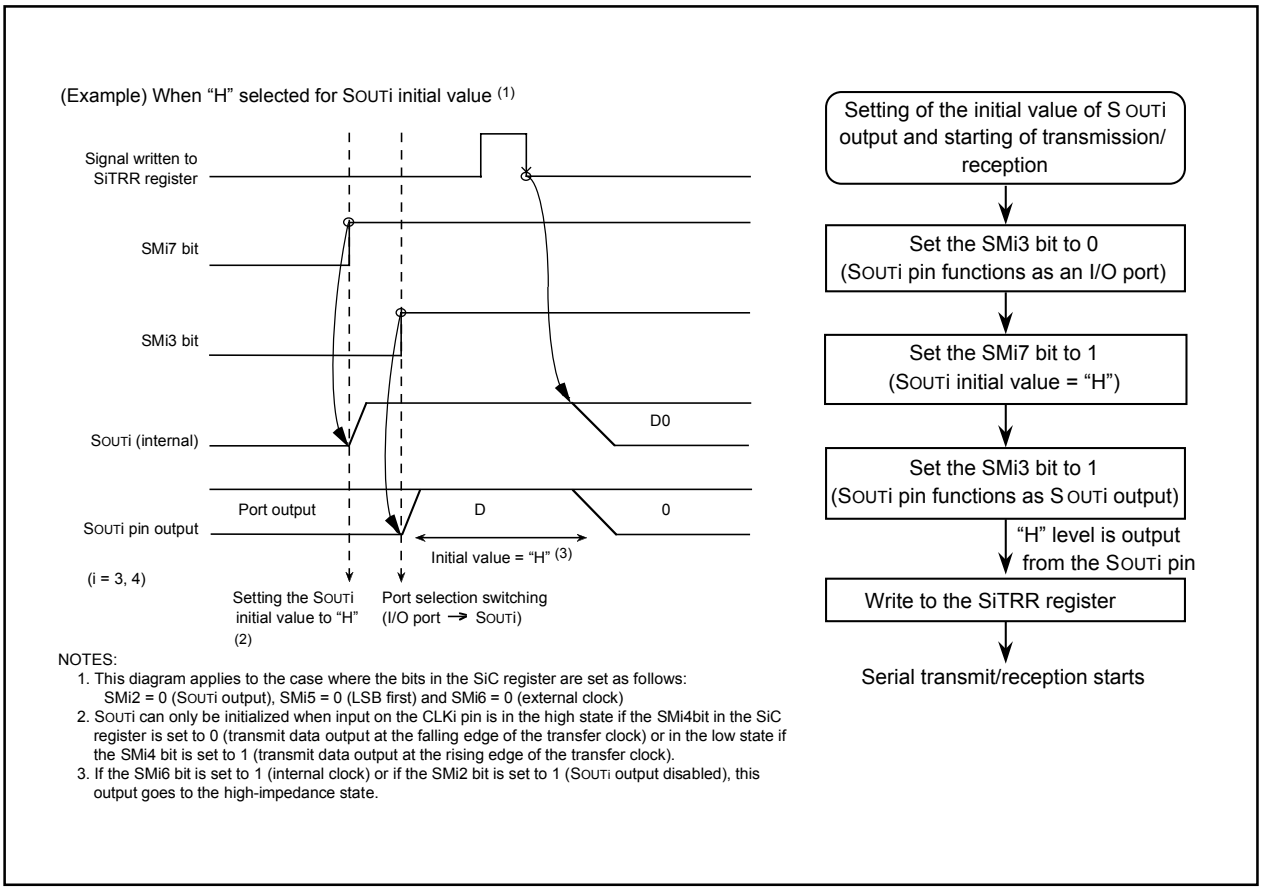
i=3 and 4

NOTES:
1. This diagram applies to the case where the SiC register bits are set as follows:
   SMi5 = 0 (LSB first) and SMi6 = 1 (internal clock)
2. When the SMi6 bit is set to 1 (internal clock), a high level is output from the CLKi pin if not transferring data.
3 When the SMi6 bit is set to 1 (internal clock), a low level is output from the CLKi pin if not transferring data.

Figure 14.38 Polarity of Transfer Clock

14.2.3 Functions for Setting an SOUTi Initial Value

If the SMi6 bit in SiC register is set to 0 (external clock), the SOUTi pin output level can be fixed high or low when not transferring data. However, when transmitting data consecutively, the last bit (bit 0) value of the last transmitted data is retained between the sccessive data transmissions. **Figure 14.39** shows the timing chart for setting an SOUTi initial value and how to set it.



NOTES:
1. This diagram applies to the case where the bits in the SiC register are set as follows:
   SMi2 = 0 (SOUTi output), SMi5 = 0 (LSB first) and SMi6 = 0 (external clock)
2. SOUTi can only be initialized when input on the CLKi pin is in the high state if the SMi4bit in the SiC register is set to 0 (transmit data output at the falling edge of the transfer clock) or in the low state if the SMi4 bit is set to 1 (transmit data output at the rising edge of the transfer clock).
3. If the SMi6 bit is set to 1 (internal clock) or if the SMi2 bit is set to 1 (SOUTi output disabled), this output goes to the high-impedance state.

Figure 14.39 SOUTi Initial Value Setting

15. A/D Converter

Note

Ports P04 to P07(AN04 to AN07), P10 to P13(AN20 to AN23) and P95 to P97(AN25 to AN27) are not available in 64-pin package. Do not use port P04 to P07(AN04 to AN07), P10 to P13(AN20 to AN23) and P95 to P97(AN25 to AN27) as analog input pins in 64-pin ver.

The MCU contains one A/D converter circuit based on 10-bit successive approximation method configured with a capacitive-coupling amplifier. The analog inputs share the pins with P100 to P107 (AN0 to AN7), P00 to P07 (AN00 to AN07), and P10 to P13, P93, P95 to P97 (AN20 to AN27), and P90 to P92 (AN30 to AN32). Similarly, $\overline{\text{ADTRG}}$ input shares the pin with P15. Therefore, when using these inputs, make sure the corresponding port direction bits are set to 0 (input mode).

When not using the A/D converter, set the VCUT bit to 0 (Vref unconnected), so that no current will flow from the Vref pin into the resistor ladder, helping to reduce the power consumption of the chip.

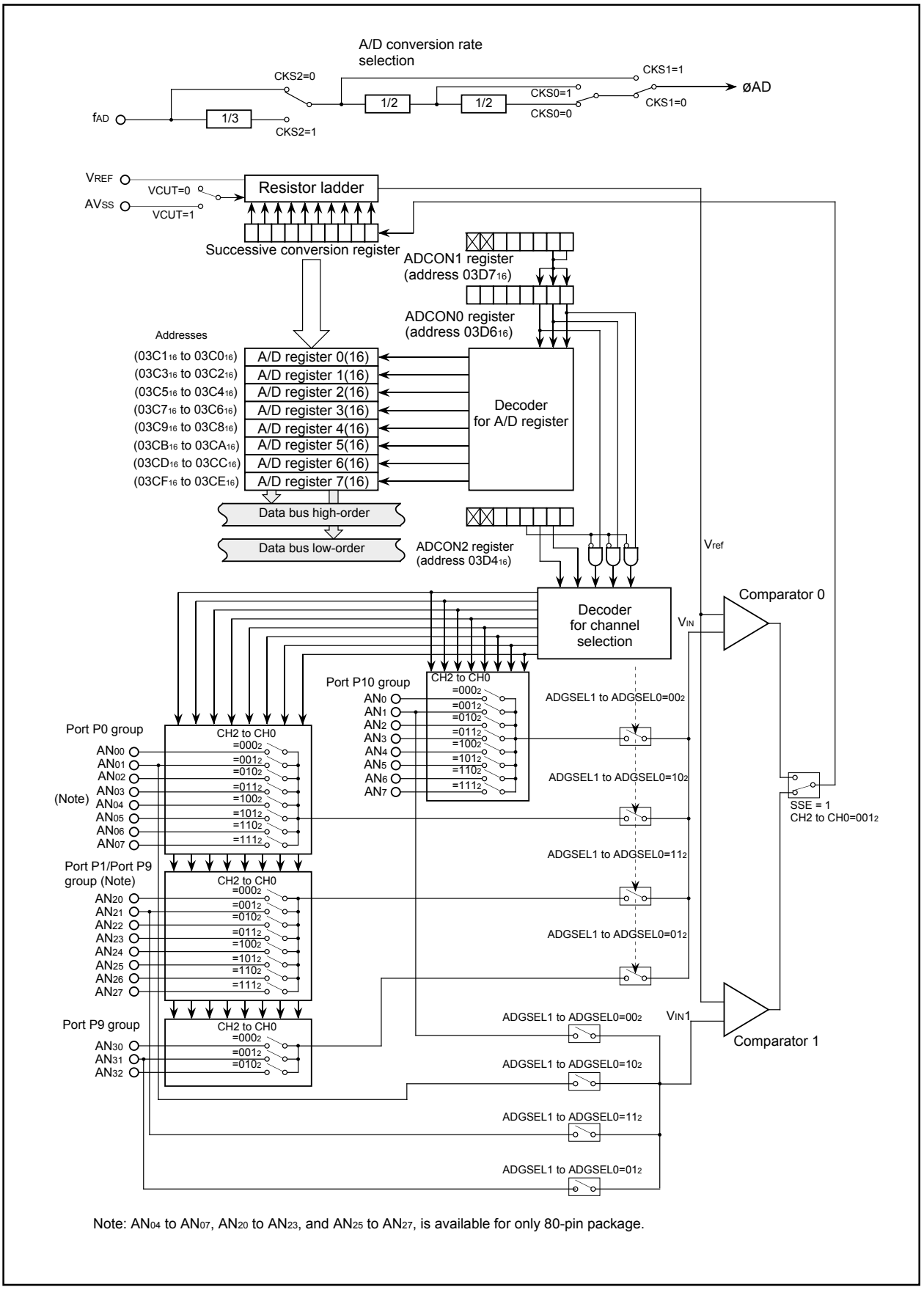
The A/D conversion result is stored in the ADi register bits for ANi, AN0i, AN2i (i = 0 to 7), and AN3i pins (i = 0 to 2). **Table 15.1** shows the A/D converter performance. **Figure 15.1** shows the A/D converter block diagram and **Figures 15.2** to **15.5** show the A/D converter associated with registers.

Table 15.1 A/D Converter Performance

Item	Performance
A/D Conversion Method	Successive approximation (capacitive coupling amplifier)
Analog Input Voltage (1)	0V to AVCC (VCC)
Operating Clock ϕAD (2)	fAD/divided-by-2 or fAD/divided-by-3 or fAD/divided-by-4 or fAD/divided-by-6 or fAD/divided-by-12 or fAD
Resolution	8-bit or 10-bit (selectable)
Integral Nonlinearity Error	When AVCC = Vref = 5V • With 8-bit resolution: ±2LSB • With 10-bit resolution: ±3LSB When AVCC = Vref = 3.3V • With 8-bit resolution: ±2LSB • With 10-bit resolution: ±5LSB
Operating Modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, repeat sweep mode 1, simultaneous sample sweep mode and delayed trigger mode 0,1
Analog Input Pins	8 pins (AN0 to AN7) + 8 pins (AN00 to AN07) + 8 pins (AN20 to AN27) + 3 pins (AN30 to AN32) (80pin-ver.) 8 pins (AN0 to AN7) + 4 pins (AN00 to AN03) + 1 pin (AN24) + 3 pins (AN30 to AN32) (64pin-ver.)
Conversion Speed 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. Not dependent on use of sample and hold function.
2. Set the ϕAD frequency to 10 MHz or less.
   Without sample-and-hold function, set the ϕAD frequency to 250 kHz or more.
   With the sample and hold function, set the ϕAD frequency to 1MHz or more.



Note: AN04 to AN07, AN20 to AN23, and AN25 to AN27, is available for only 80-pin package.

Figure 15.1 A/D Converter Block Diagram

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	Function varies with each operation mode	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 0 0: One-shot mode or Delayed trigger mode 0,1 0 1: Repeat mode 1 0: Single sweep mode or Simultaneous sample sweep mode 1 1: Repeat sweep mode 0 or Repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

Symbol	Address	After Reset
ADCON1	$03D7_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit	Function varies with each operation mode	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0 : Other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (2)	0 : Vref not connected 1 : Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

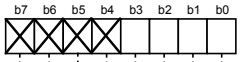
Symbol	Address	After Reset
ADCON2	$03D4_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Function varies with each operation mode	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.2 ADCON0 to ADCON2 Registers

A/D Trigger Control Register (1, 2)



Symbol	Address	After Reset
ADTRGCON	$03D2_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SSE	A/D Operation Mode Select Bit 2	0 : Other than simultaneous sample sweep mode or delayed trigger mode 0,1 1 : Simultaneous sample sweep mode or delayed trigger mode 0,1	RW
DTE	A/D Operation Mode Select Bit 3	0 : Other than delayed trigger mode 0,1 1 : Delayed trigger mode 0,1	RW
HPTRG0	AN0 Trigger Select Bit	Function varies with each operation mode	RW
HPTRG1	AN1 Trigger Select Bit	Function varies with each operation mode	RW
—— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, its content is 0		—

NOTES:
1. If the ADTRGCON register is rewritten during A/D conversion, the conversion result will be undefined.
2. Set 00_{16} in this register in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1.

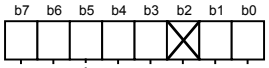
Figure 15.3 ADTRGCON Register

Table 15.2 A/D Conversion Frequency Select

CKS2	CKS1	CKS0	ØAD
0	0	0	fAD divided by 4
0	0	1	fAD divided by 2
0	1	0	fAD
0	1	1	
1	0	0	fAD divided by 12
1	0	1	fAD divided by 6
1	1	0	fAD divided by 3
1	1	1	

NOTE:
1. ØAD frequency must be under 10 MHz. Combination of the CKS0 bit in the ADCON0 register, the CKS1 bit in the ADCON1 register, and the CKS2 bit in the ADCON2 register selects ØAD.

A/D Conversion Status Register 0 (1)



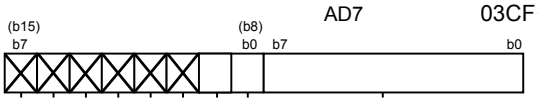
Symbol	Address	After reset
ADSTAT0	03D3₁₆	00₁₆

Bit Symbol	Bit Name	Function	RW
ADERR0	AN1 trigger status flag	0: AN1 trigger did not occur during AN0 conversion 1: AN1 trigger occured during AN0 conversion	RW
ADERR1	Conversion termination flag	0: Conversion not terminated 1: Conversion terminated by Timer B0 underflow	RW
(b2)	Nothing is assigned. If necessary, set to 0. When read, its content is 0		—
ADTCSF	Delayed trigger sweep status flag	0: Sweep not in progress 1: Sweep in progress	RO
ADSTT0	AN0 conversion status flag	0: AN0 conversion not in progress 1: AN0 conversion in progress	RO
ADSTT1	AN1 conversion status flag	0: AN1 conversion not in progress 1: AN1 conversion in progress	RO
ADSTRT0	AN0 conversion completion status flag	0: AN0 conversion not completed 1: AN0 conversion completed	RW
ADSTRT1	AN1 conversion completion status flag	0: AN1 conversion not completed 1: AN1 conversion completed	RW

NOTE:
1. ADSTAT0 register is valid only when the DTE bit in the ADTRGCON register is set to 1.

A/D Register i (i=0 to 7)

Symbol	Address	After Reset
AD0	03C1₁₆ to 03C0₁₆	Undefined
AD1	03C3₁₆ to 03C2₁₆	Undefined
AD2	03C5₁₆ to 03C4₁₆	Undefined
AD3	03C7₁₆ to 03C6₁₆	Undefined
AD4	03C9₁₆ to 03C8₁₆	Undefined
AD5	03CB₁₆ to 03CA₁₆	Undefined
AD6	03CD₁₆ to 03CC₁₆	Undefined
AD7	03CF₁₆ to 03CE₁₆	Undefined



Function		RW
When the BITS bit in the ADCON1 register is 1 (10-bit mode)	When the BITS bit in the ADCON1 register is 0 (8-bit mode)	RW
Eight low-order bits of A/D conversion result	A/D conversion result	RO
Two high-order bits of A/D conversion result	When read, its content is undefined	RO
Nothing is assigned. If necessary, set to 0. When read, its content is 0		—

Figure 15.4 ADSTAT0 Register and AD0 to AD7 Registers

Timer B2 Special Mode Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	0	0					

Symbol	Address	After Reset
TB2SC	039E16	X0000000₂

Bit Symbol	Bit Name	Function	RW
PWCOM	Timer B2 reload timing switch bit	0: Timer B2 underflow 1: Timer A output at odd-numbered (2)	RW
IVPCR1	Three-phase output port $\overline{SD}$ control bit 1 (3, 4, 7)	0: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) disabled 1: Three-phase output forcible cutoff by $\overline{SD}$ pin input (high impedance) enabled	RW
TB0EN	Timer B0 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode(5)	RW
TB1EN	Timer B1 operation mode select bit	0: Other than A/D trigger mode 1: A/D trigger mode(5)	RW
TB2SEL	Trigger select bit	0: TB2 interrupt 1: Underflow of TB2 interrupt generation frequency setting counter [ICTB2](6)	RW
$\overline{\text{(b6-b5)}}$	Reserved bit	Set to 0	RW
— (b7)	Nothing is assigned. If necessary, set to 0. When read, its content is 0		—

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enabled).
2. If the INV11 bit is 0 (three-phase mode 0) or the INV06 bit is 1 (triangular wave modulation mode), set this bit to 0 (timer B2 underflow).
3. When setting the IVPCR1 bit to 1 (three-phase output forcible cutoff by $\overline{SD}$ pin input enabled), set the PD8$_5$ bit to 0 (= input mode).
4. Related pins are U(P8$_0$), $\overline{U}$(P8$_1$), V(P7$_2$), $\overline{V}$(P7$_3$), W(P7$_4$), $\overline{W}$(P7$_5$). After forcible cutoff, input "H" to the P8$_5$/$\overline{NMI}$/$\overline{SD}$ pin. Set the IVPCR1 bit to 0, and this forcible cutoff will be reset. If “L” is input to the P8$_5$/$\overline{NMI}$/$\overline{SD}$ pin, a three-phase motor control timer output will be disabled (INV03=0). At this time, when the IVPCR1 bit is 0, the target pins changes to programmable I/O port. When the IVPCR1 bit is 1, the target pins changes to high-impedance state regardless of which functions of those pins are used.
5. When using in delay trigger mode, set both bits TB0EN and TB1EN to 1.
6. When setting the TB2SEL bit to 1 (underflow of TB2 interrupt generation frequency setting counter[ICTB2]), set the INV02 bit to 1 (three-phase motor control timer function).
7. Refer to **18.6 Digital Debounce function** for $\overline{SD}$ input.

Figure 15.5 TB2SC Register

15.1 Operating Modes

15.1.1 One-Shot Mode

In one-shot mode, analog voltage applied to a selected pin is once converted to a digital code. **Table 15.3** shows the one-shot mode specifications. **Figure 15.6** shows the operation example in one-shot mode. **Figure 15.7** shows registers ADCON0 to ADCON2 in one-shot mode.

Table 15.3 One-shot Mode Specifications

Item	Specification
Function	Bits CH2 to CH0 in the ADCON0 register and registers ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to a selected pin is once converted to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) • When the TRG bit in the ADCON0 register is 1 (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to 1 (A/D conversion started)
A/D Conversion Stop Condition	• A/D conversion completed (If a software trigger is selected, the ADST bit is set to 0 (A/D conversion halted)). • Set the ADST bit to 0
Interrupt Request Generation Timing	A/D conversion completed
Analog Input Pin	Select one pin from AN_0 to AN_7, $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, $AN3_0$ to $AN3_2$
Readout of A/D Conversion Result	Readout one of registers AD0 to AD7 that corresponds to the selected pin

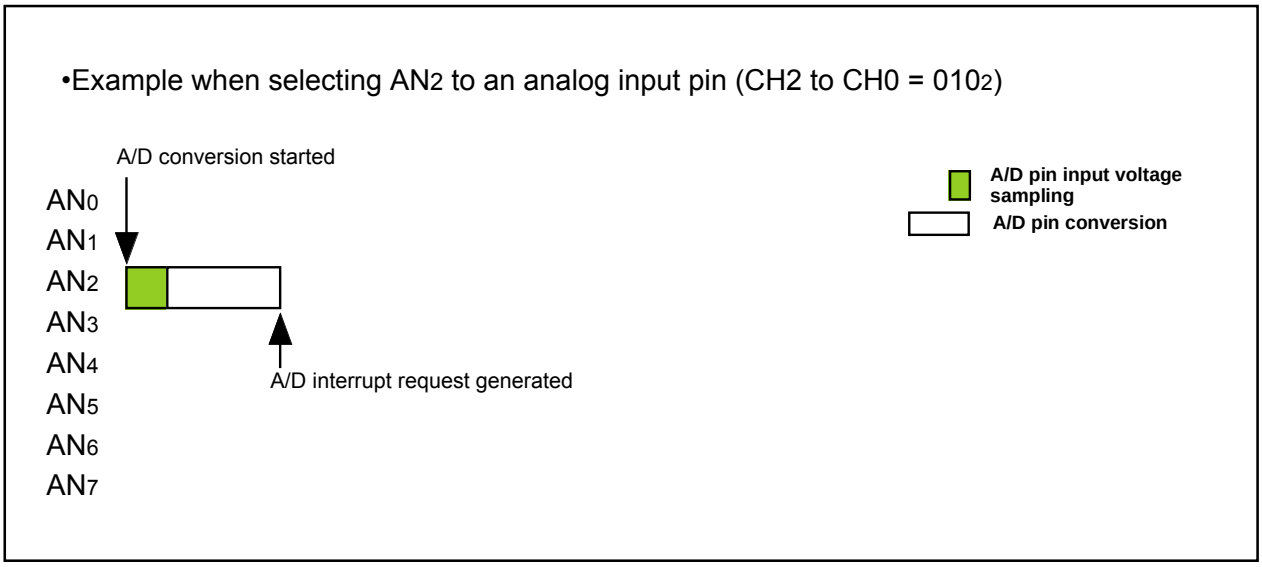


Figure 15.6 Operation Example in One-Shot Mode

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
			0	0				ADCON0	03D6$_{16}$	00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit (2, 3)	b2 b1 b0 0 0 0: Select AN0 0 0 1: Select AN1 0 1 0: Select AN2 0 1 1: Select AN3 1 0 0: Select AN4 1 0 1: Select AN5 1 1 0: Select AN6 1 1 1: Select AN7	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0 (3)	b4 b3 0 0: One-shot mode or delayed trigger mode 0,1	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger ($\overline{\text{ADTRG}}$ trigger)	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. After rewriting bits MD1 and MD0, set bits CH2 to CH0 over again using an another instruction.

A/D Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	1			0			ADCON1	03D7$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit	Invalid in one-shot mode	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (2)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	0		0				ADCON2	03D4$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Set to 0 in one-shot mode	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.7 ADCON0 to ADCON2 Registers in One-Shot Mode

15.1.2 Repeat mode

In repeat mode, analog voltage applied to a selected pin is repeatedly converted to a digital code. **Table 15.4** shows the repeat mode specifications. **Figure 15.8** shows the operation example in repeat mode. **Figure 15.9** shows the ADCON0 to ADCON2 registers in repeat mode.

Table 15.4 Repeat Mode Specifications

Item	Specification
Function	Bits CH2 to CH0 in the ADCON0 register and the ADGSEL1 to ADGSEL0 bits in the ADCON2 register select pins. Analog voltage applied to a selected pin is repeatedly converted to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) • When the TRG bit in the ADCON0 register is 1 (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from “H” to “L” after setting the ADST bit to 1 (A/D conversion started)
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pin	Select one pin from AN_0 to AN_7, $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, and $AN3_0$ to $AN3_2$
Readout of A/D Conversion Result	Readout one of the AD0 to AD7 registers that corresponds to the selected pin

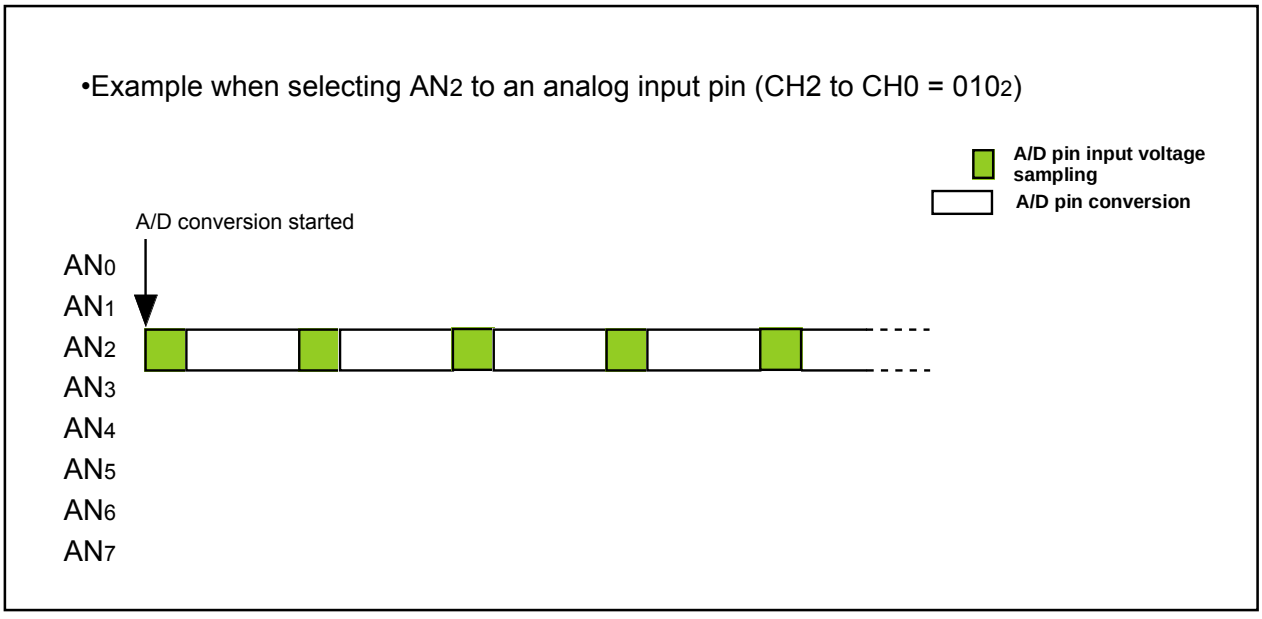


Figure 15.8 Operation Example in Repeat Mode

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	1			

Symbol	Address	After Reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit (2, 3)	b2 b1 b0 0 0 0: Select AN_0 0 0 1: Select AN_1 0 1 0: Select AN_2 0 1 1: Select AN_3 1 0 0: Select AN_4 1 0 1: Select AN_5 1 1 0: Select AN_6 1 1 1: Select AN_7	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0 (3)	b4 b3 0 1: Repeat mode	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger ($\overline{ADTRG}$ trigger)	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN_{00} to AN_{07}, AN_{20} to AN_{27}, and AN_{30} to AN_{32} can be used in the same way as AN_0 to AN_7. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. After rewriting bits MD1 and MD0, set bits CH2 to CH0 over again using an another instruction.

A/D Control Register 1 (1)

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

Symbol	Address	After Reset
ADCON1	$03D7_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit	Invalid in repeat mode	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect bit (2)	1: Vref connected	RW
$\overline{(b7\text{-}b6)}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

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

Symbol	Address	After Reset
ADCON2	$03D4_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{(b3)}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Set to 0 in one-shot mode	RW
$\overline{(b7\text{-}b6)}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.9 ADCON0 to ADCON2 Registers in Repeat Mode

15.1.3 Single Sweep Mode

In single sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. **Table 15.5** shows the single sweep mode specifications. **Figure 15.10** shows the operation example in single sweep mode. **Figure 15.11** shows the ADCON0 to ADCON2 registers in single sweep mode.

Table 15.5 Single Sweep Mode Specifications

Item	Specification
Function	Bits SCAN1 to SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) • When the TRG bit in the ADCON0 register is 1 (hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started)
A/D Conversion Stop Condition	• A/D conversion completed(When selecting a software trigger, the ADST bit is set to 0 (A/D conversion halted)). • Set the ADST bit to 0
Interrupt Request Generation Timing	A/D conversion completed
Analog Input Pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), AN_0 to AN_7 (8 pins) (1)
Readout of A/D Conversion Result	Readout one of registers AD0 to AD7 that corresponds to the selected pin

NOTE:

1. $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, and $AN3_0$ to $AN3_2$ can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

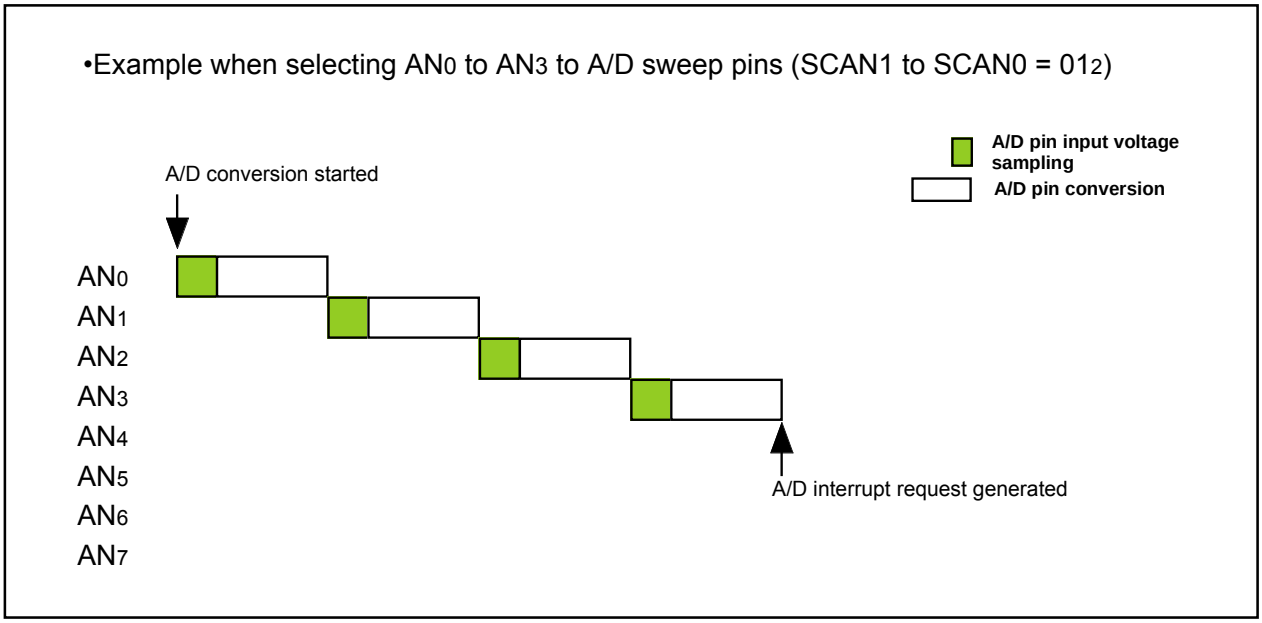


Figure 15.10 Operation Example in Single Sweep Mode

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
			1	0				ADCON0	03D6$_{16}$	00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	Invalid in single sweep mode	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 1 0: Single sweep mode or Simultaneous sample sweep mode	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger ($\overline{\text{ADTRG}}$ trigger)	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	1			0			ADCON1	03D7$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit (2)	When single sweep mode is selected, b1 b0 0 0: AN$_0$ to AN$_1$ (2 pins) 0 1: AN$_0$ to AN$_3$ (4 pins) 1 0: AN$_0$ to AN$_5$ (6 pins) 1 1: AN$_0$ to AN$_7$ (8 pins)	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (3)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN$_0$ to AN$_7$. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 μs or more before starting A/D conversion.

A/D Control Register 2(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	0		0				ADCON2	03D4$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Set to 0 in single sweep mode	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.11 ADCON0 to ADCON2 Registers in Single Sweep Mode

15.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltages applied to the selected pins are repeatedly converted to a digital code. **Table 15.6** shows the repeat sweep mode 0 specifications. **Figure 15.12** shows the operation example in repeat sweep mode 0. **Figure 15.13** shows the ADCON0 to ADCON2 registers in repeat sweep mode 0.

Table 15.6 Repeat Sweep Mode 0 Specifications

Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the selected pins is repeatedly converted to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) • When the TRG bit in the ADCON0 register is 1 (Hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started)
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), AN_0 to AN_7 (8 pins) (1)
Readout of A/D Conversion Result	Readout one of registers AD0 to AD7 that corresponds to the selected pin

NOTES:

1. $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, and $AN3_0$ to $AN3_2$ can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

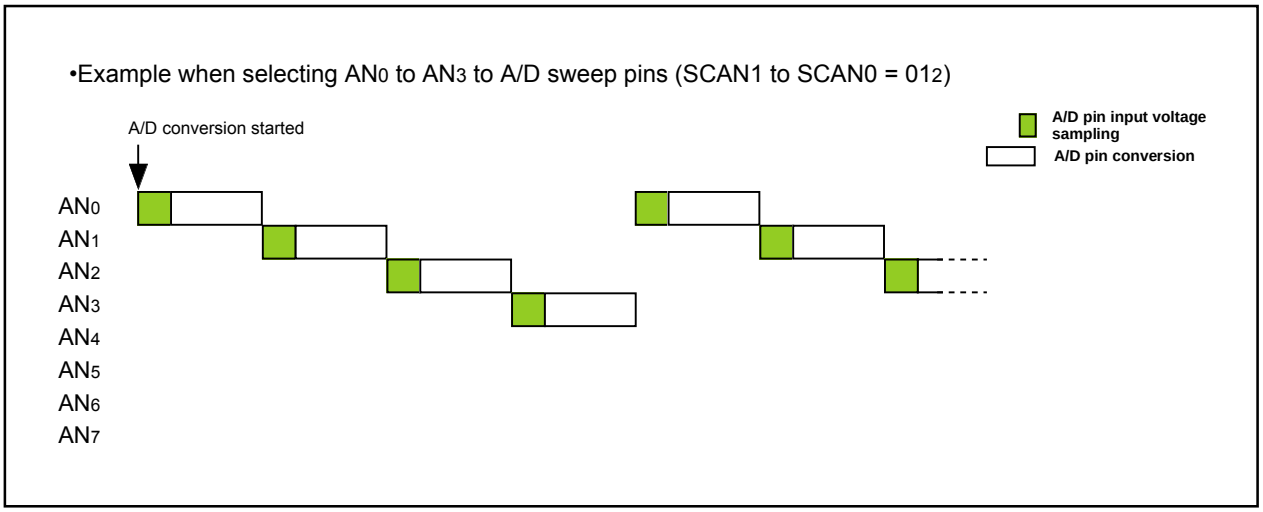


Figure 15.12 Operation Example in Repeat Sweep Mode 0

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
			1	1				ADCON0	03D6$_{16}$	00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 1 1: Repeat sweep mode 0 or repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger ($\overline{\text{ADTRG}}$ trigger)	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	1			0			ADCON1	03D7$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit (2)	When repeat sweep mode 0 is selected, b1 b0 0 0: AN0 to AN1 (2 pins) 0 1: AN0 to AN3 (4 pins) 1 0: AN0 to AN5 (6 pins) 1 1: AN0 to AN7 (8 pins)	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (3)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	0		0				ADCON2	03D4$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Set to 0 in repeat sweep mode 0	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.13 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 0

15.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage is applied to the all selected pins are converted to a digital code, with mainly used in the selected pins. **Table 15.7** shows the repeat sweep mode 1 specifications. **Figure 15.14** shows the operation example in repeat sweep mode 1. **Figure 15.15** shows registers ADCON0 to ADCON2 in repeat sweep mode 1.

Table 15.7 Repeat Sweep Mode 1 Specifications

Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register mainly select pins. Analog voltage applied to the all selected pins is repeatedly converted to a digital code Example : When selecting AN_0 Analog voltage is converted to a digital code in the following order $AN_0 \rightarrow AN_1 \rightarrow AN_0 \rightarrow AN_2 \rightarrow AN_0 \rightarrow AN_3$, and so on.
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) • When the TRG bit in the ADCON0 register is 1 (hardware trigger) The $\overline{ADTRG}$ pin input changes state from “H” to “L” after setting the ADST bit to 1 (A/D conversion started)
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pins Mainly Used in A/D Conversions	Select from AN_0 (1 pins), AN_0 to AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (4 pins) (1)
Readout of A/D Conversion Result	Readout one of registers AD0 to AD7 that corresponds to the selected pin

NOTES:

1. $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, and $AN3_0$ to $AN3_2$ can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

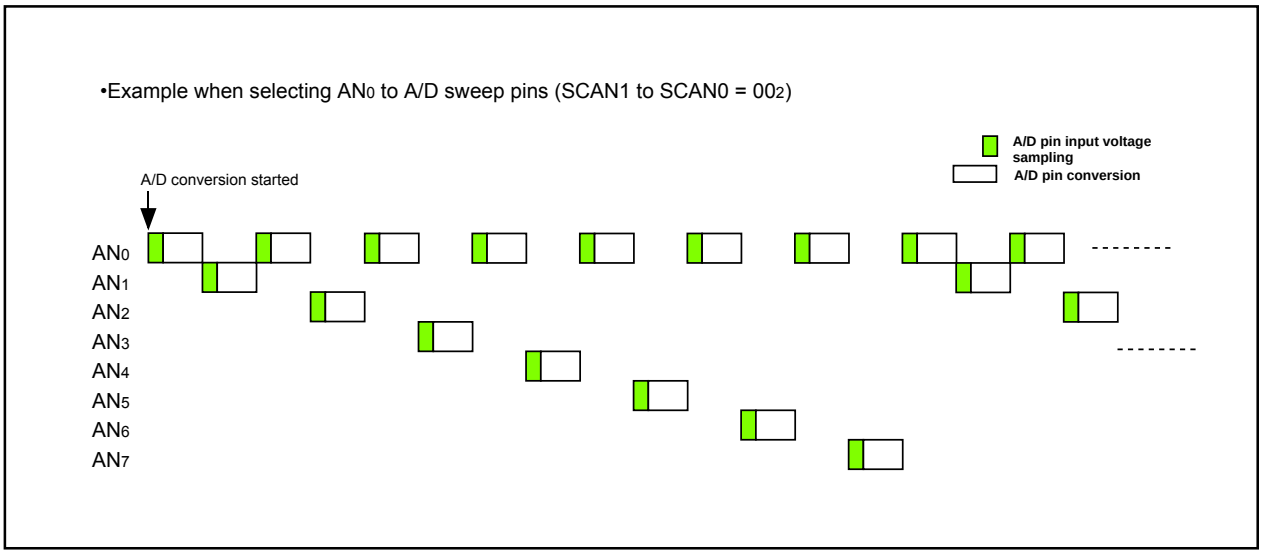


Figure 15.14 Operation Example in Repeat Sweep Mode 1

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

Symbol: ADCON0 | Address: 03D6$_{16}$ | After Reset: 00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 1 1: Repeat sweep mode 0 or repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger ($\overline{\text{ADTRG}}$ trigger)	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 (1)

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

Symbol: ADCON1 | Address: 03D7$_{16}$ | After Reset: 00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit (2)	When repeat sweep mode 0 is selected, b1 b0 0 0: AN$_0$ (1 pin) 0 1: AN$_0$ to AN$_1$ (2 pins) 1 0: AN$_0$ to AN$_2$ (3 pins) 1 1: AN$_0$ to AN$_3$ (4 pins)	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	1: Repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (3)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN$_0$ to AN$_7$. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

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

Symbol: ADCON2 | Address: 03D4$_{16}$ | After Reset: 00$_{16}$

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Set to 0 in repeat sweep mode 1	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.15 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 1

15.1.6 Simultaneous Sample Sweep Mode

In simultaneous sample sweep mode, analog voltages applied to the selected pins are converted one-by-one to a digital code. The input voltages of AN0 and AN1 are sampled simultaneously using two circuits of sample and hold circuit. **Table 15.8** shows the simultaneous sample sweep mode specifications. **Figure 15.16** shows the operation example in simultaneous sample sweep mode. **Figure 15.17** shows registers ADCON0 to ADCON2 and **Figure 15.18** shows ADTRGCON registers in simultaneous sample sweep mode. **Table 15.9** shows the trigger select bit setting in simultaneous sample sweep mode. In simultaneous sample sweep mode, Timer B0 underflow can be selected as a trigger by combining software trigger, $\overline{\text{ADTRG}}$ trigger, Timer B2 underflow, Timer B2 interrupt generation frequency setting counter underflow or A/D trigger mode of Timer B.

Table 15.8 Simultaneous Sample Sweep Mode Specifications

Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the selected pins is converted one-by-one to a digital code. At this time, the input voltage of AN_0 and AN_1 are sampled simultaneously.
A/D Conversion Start Condition	When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) When the TRG bit in the ADCON0 register is 1 (hardware trigger) The trigger is selected by bits TRG1 and HPTRG0 (See **Table 15.9**) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started) Timer B0, B2 or Timer B2 interrupt generation frequency setting counter underflow after setting the ADST bit to 1 (A/D conversion started)
A/D Conversion Stop Condition	A/D conversion completed (If selecting software trigger, the ADST bit is automatically set to 0). Set the ADST bit to 0 (A/D conversion halted)
Interrupt Generation Timing	A/D conversion completed
Analog Input Pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins) (1)
Readout of A/D conversion result	Readout one of registers AN0 to AN7 that corresponds to the selected pin

NOTE:
1. $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, and $AN3_0$ to $AN3_2$ can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

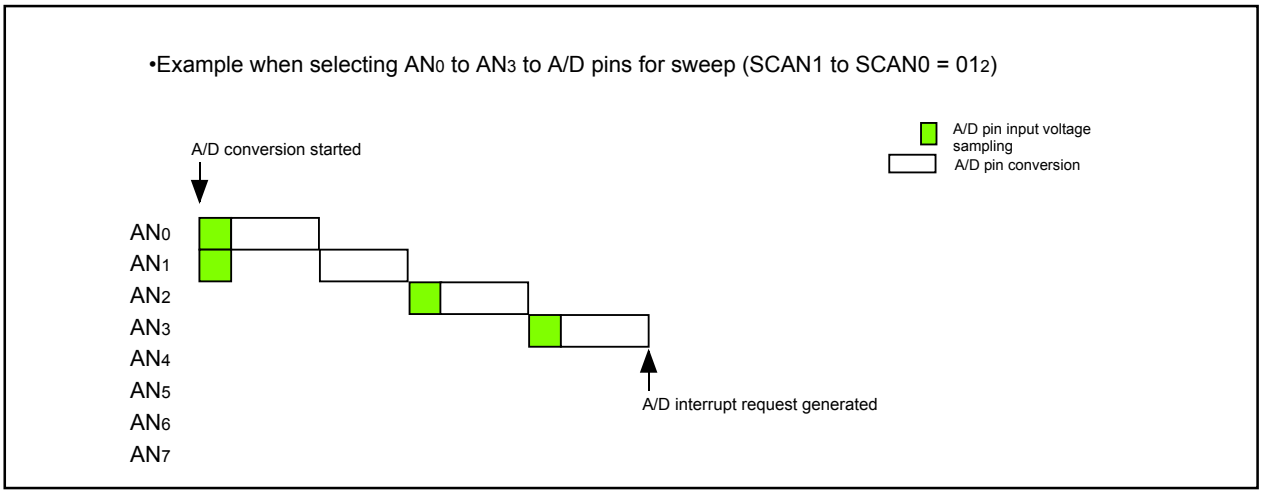


Figure 15.16 Operation Example in Simultaneous Sample Sweep Mode

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
			1	1				ADCON0	03D6$_{16}$	00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 1 0: Single sweep mode or simultaneous sample sweep mode	RW
MD1			RW
TRG	Trigger select bit	See **Table 15.9**	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	1			0			ADCON1	03D7$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit (2)	When simultaneous sample sweep mode is selected, b1 b0 0 0: AN$_0$ to AN$_1$ (2 pins) 0 1: AN$_0$ to AN$_3$ (4 pins) 1 0: AN$_0$ to AN$_5$ (6 pins) 1 1: AN$_0$ to AN$_7$ (8 pins)	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (3)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN$_0$ to AN$_7$. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 μs or more before starting A/D conversion.

A/D Control Register 2(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X			0			1	ADCON2	03D4$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	Set to 1 in simultaneous sample sweep mode	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	See **Table 15.9**	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.17 ADCON0 to ADCON2 Registers in Simultaneous Sample Sweep Mode

A/D Trigger Control Register (1)

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

Symbol	Address	After Reset
ADTRGCON	$03D2_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SSE	A/D operation mode select bit 2	1: Simultaneous sample sweep mode or delayed trigger mode 0, 1	RW
DTE	A/D operation mode select bit 3	0: Other than delayed trigger mode 0, 1	RW
HPTRG0	AN0 trigger select bit	See **Table 15.9**	RW
HPTRG1	AN1 trigger select bit	Set to 0 in simultaneous sample sweep mode	RW
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If ADTRGCON is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.18 ADTRGCON Register in Simultaneous Sample Sweep Mode

Table 15.9 Trigger Select Bit Setting in Simultaneous Sample Sweep Mode

TRG	TRG1	HPTRG0	TRIGGER
0	-	-	Software trigger
1	-	1	Timer B0 underflow (1)
1	0	0	$\overline{\text{ADTRG}}$
1	1	0	Timer B2 or Timer B2 interrupt generation frequency setting counter underflow (2)

NOTES:
1. A count can be started for Timer B2, Timer B2 interrupt generation frequency setting counter underflow or the INT5 pin falling edge as count start conditions of Timer B0.
2. Select Timer B2 or Timer B2 interrupt generation frequency setting counter using the TB2SEL bit in the TB2SC register.

15.1.7 Delayed Trigger Mode 0

In delayed trigger mode 0, analog voltages applied to the selected pins are converted one-by-one to a digital code. The delayed trigger mode 0 used in combination with A/D trigger mode of Timer B. The Timer B0 underflow starts a single sweep conversion. After completing the AN_0 pin conversion, the AN_1 pin is not sampled and converted until the Timer B1 underflow is generated. When the Timer B1 underflow is generated, the single sweep conversion is restarted with the AN_1 pin. **Table 15.10** shows the delayed trigger mode 0 specifications. **Figure 15.19** shows the operation example in delayed trigger mode 0. **Figures 15.20** and **15.21** show each flag operation in the ADSTAT0 register that corresponds to the operation example. **Figure 15.22** shows registers ADCON0 to ADCON2 in delayed trigger mode 0. **Figure 15.23** shows the ADTRGCON register in delayed trigger mode 0 and **Table 15.11** shows the trigger select bit setting in delayed trigger mode 0.

Table 15.10 Delayed Trigger Mode 0 Specifications

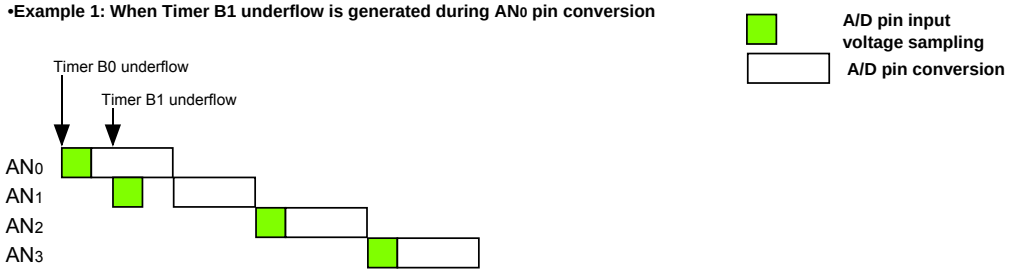
Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the input voltage of the selected pins are converted one-by-one to the digital code. At this time, timer B0 underflow generation starts AN_0 pin conversion. Timer B1 underflow generation starts conversion after the AN_1 pin. (1)
A/D Conversion Start	AN_0 pin conversion start condition •When Timer B0 underflow is generated if Timer B0 underflow is generated again before Timer B1 underflow is generated , the conversion is not affected •When Timer B0 underflow is generated during A/D conversion of pins after the AN_1 pin, conversion is halted and the sweep is restarted from the AN_0 pin again AN_1 pin conversion start condition •When Timer B1 underflow is generated during A/D conversion of the AN_0 pin, the input voltage of the AN_1 pin is sampled. The AN_1 conversion and the rest of the sweep start when AN_0 conversion is completed.
A/D Conversion Stop Condition	•When single sweep conversion from the AN_0 pin is completed •Set the ADST bit to 0 (A/D conversion halted)(2)
Interrupt request generation timing	A/D conversion completed
Analog input pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins) and AN_0 to AN_7 (8 pins)(3)
Readout of A/D conversion result	Readout one of registers AN0 to AN7 that corresponds to the selected pins

NOTES:

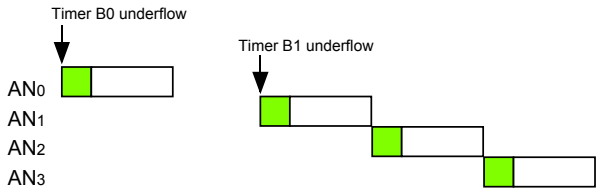
1. Set the larger value than the value of the timer B0 register to the timer B1 register. The count source for timer B0 and timer B1 must be the same.
2. Do not write 1 (A/D conversion started) to the ADST bit in delayed trigger mode 0. When write 1, unexpected interrupts may be generated.
3. $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, and $AN3_0$ to $AN3_2$ can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

•Example when selecting AN_0 to AN_3 to A/D sweep pins (SCAN1 to SCAN0 = 01_2)

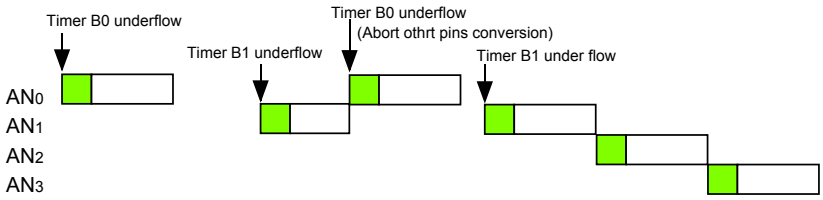
•Example 1: When Timer B1 underflow is generated during AN_0 pin conversion



•Example 2: When Timer B1 underflow is generated after AN_0 pin conversion



•Example 3: When Timer B0 underflow is generated during A/D conversion of any pins except AN_0 pin



•Example 4: When Timer B0 underflow is generated again before Timer B1 underflow is generated after Timer B0 underflow generation

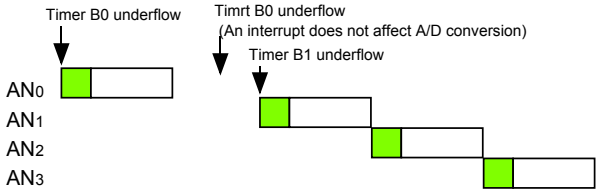
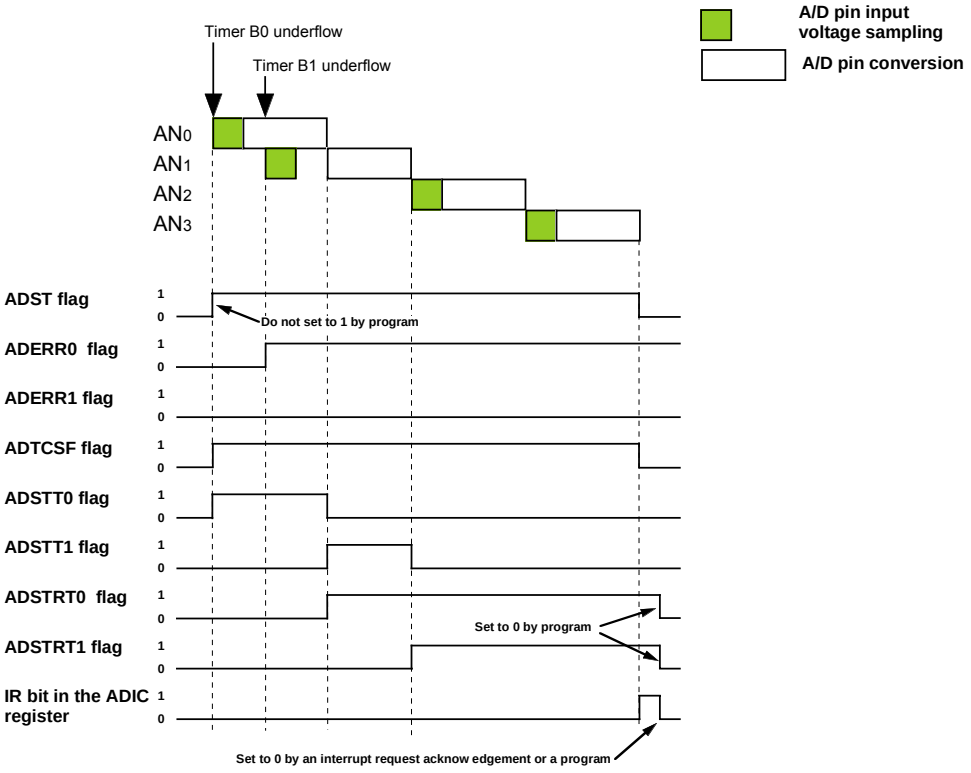


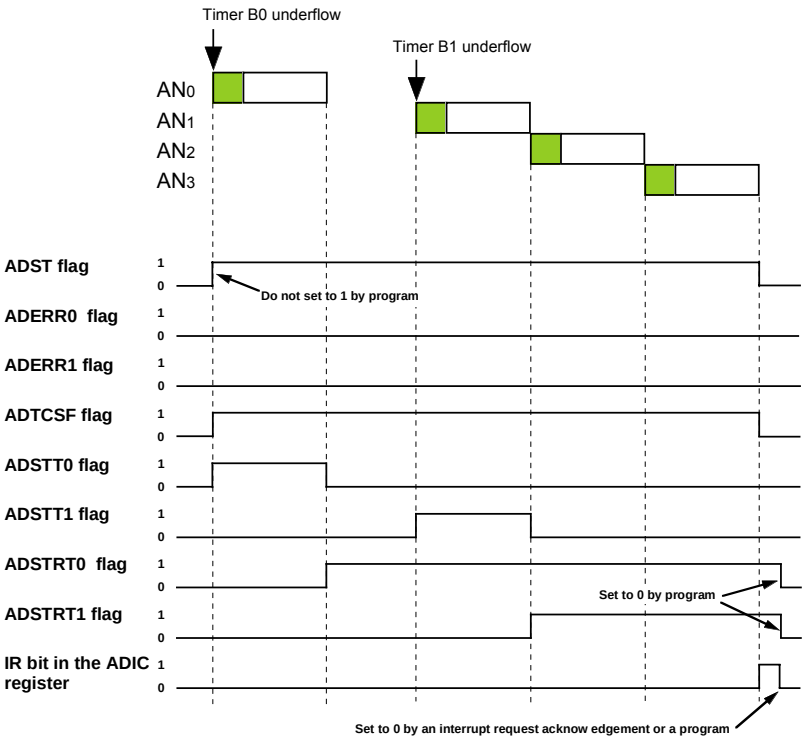
Figure 15.19 Operation Example in Delayed Trigger Mode 0

•Example when selecting AN_0 to AN_3 to A/D sweep pins (SCAN1 to SCAN0 = 01_2)

•Example 1: When Timer B1 underflow is generated during AN_0 pin conversion



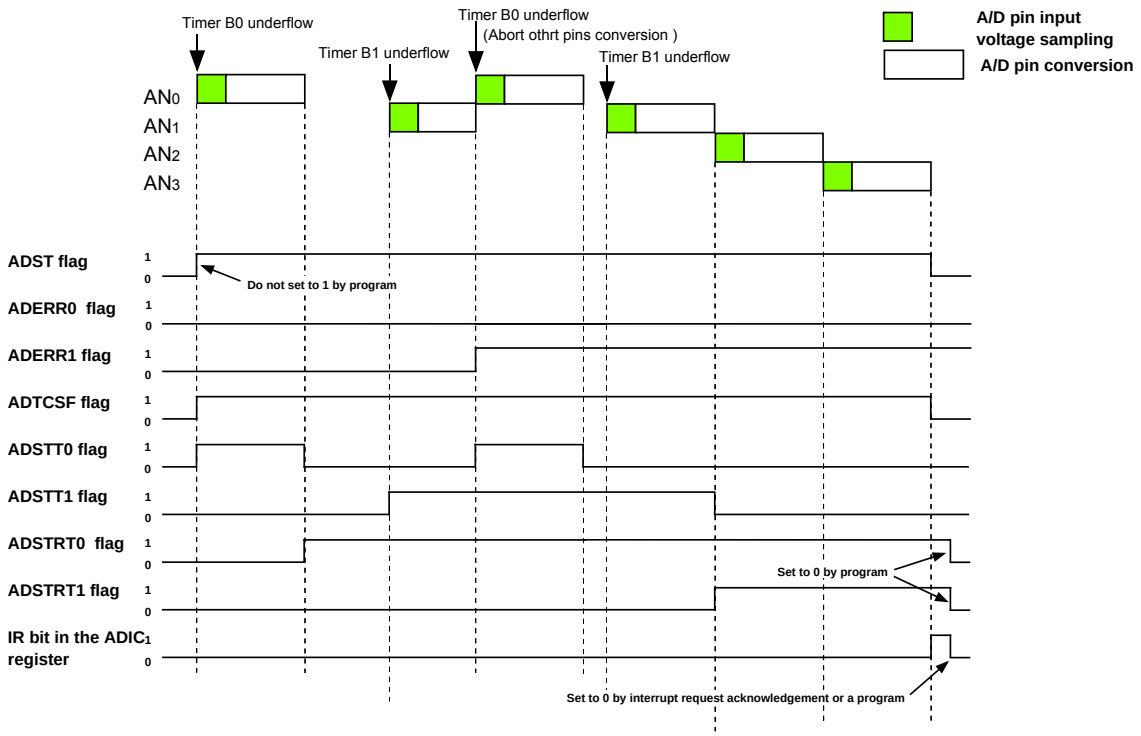
•Example 2: When Timer B1 underflow is generated after AN_0 pin conversion



ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6, and 7 in the ADSTAT0 register

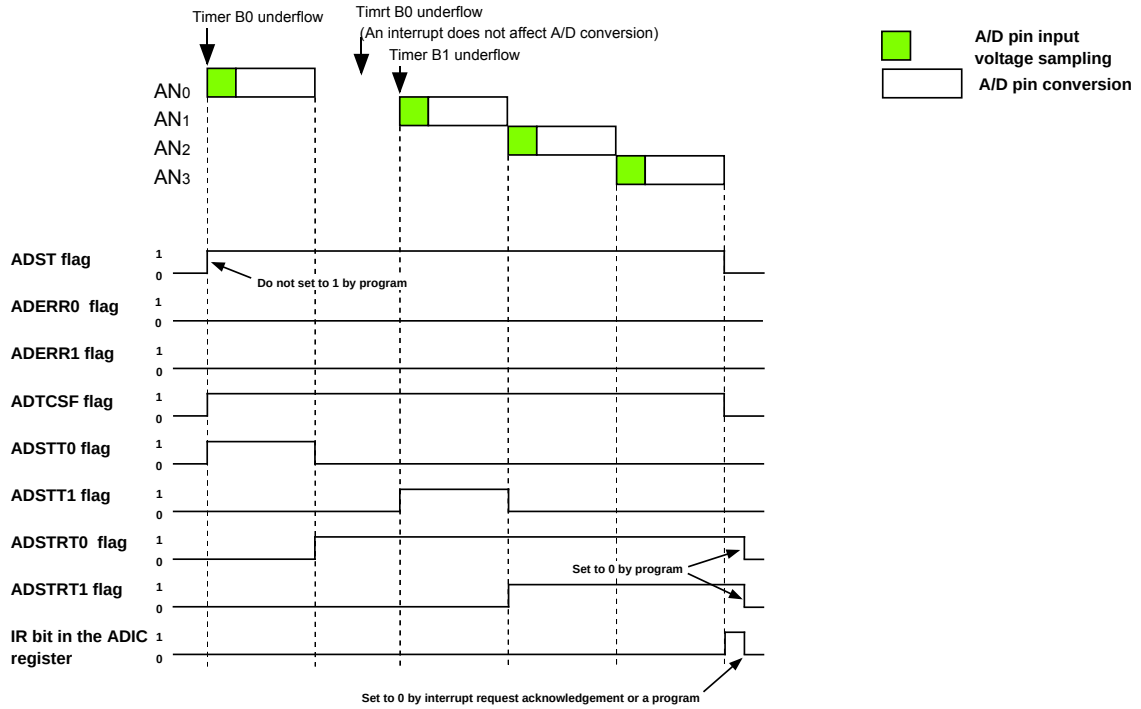
Figure 15.20 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (1)

•Example 3: When Timer B0 underflow is generated during A/D pin conversion of any pins except AN0 pin



ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

•Example 4: After Timer B0 underflow is generated and when Timer B0 underflow is generated again before Timer B1 underflow is genetaed



ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

Figure 15.21 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 0 (2)

A/D Control Register 0 (1)

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

Symbol	Address	After Reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	b2 b1 b0 1 1 1: Set to 111b in delayed trigger mode 0	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 0 0: One-shot mode or delayed trigger mode 0, 1	RW
MD1			RW
TRG	Trigger select bit	See **Table 15.11**	RW
ADST	A/D conversion start flag(2)	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. Do not write 1 in delayed trigger mode 0. When write, set to 0.

A/D Control Register 1(1)

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

Symbol	Address	After Reset
ADCON1	$03D7_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit (2)	When delayed trigger sweep mode 0 is selected, b1 b0 0 0: AN0 to AN1 (2 pins) 0 1: AN0 to AN3 (4 pins) 1 0: AN0 to AN5 (6 pins) 1 1: AN0 to AN7 (8 pins)	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (3)	1: Vref connected	RW
(b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

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

Symbol	Address	After Reset
ADCON2	$03D4_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit (2)	1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
(b3)	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	See **Table 15.11**	RW
(b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.
2. Set to 1 in delayed trigger mode 0.

Figure 15.22 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 0

A/D Trigger Control Register (1)

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

Symbol	Address	After Reset
ADTRGCON	$03D2_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SSE	A/D operation mode select bit 2	Simultaneous sample sweep mode or delayed trigger mode 0, 1	RW
DTE	A/D operation mode select bit 3	Delayed trigger mode 0, 1	RW
HPTRG0	AN_0 trigger select bit	See **Table 15.11**	RW
HPTRG1	AN_1 trigger select bit	See **Table 15.11**	RW
— (b7-b4)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If ADTRGCON is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.23 ADTRGCON Register in Delayed Trigger Mode 0

Table 15.11 Trigger Select Bit Setting in Delayed Trigger Mode 0

TRG	TRG1	HPTRG0	HPTRG1	Trigger
0	0	1	1	Timer B0, B1 underflow

15.1.8 Delayed Trigger Mode 1

In delayed trigger mode 1, analog voltages applied to the selected pins are converted one-by-one to a digital code. When the input of the $\overline{\text{ADTRG}}$ pin (falling edge) changes state from "H" to "L", a single sweep conversion is started. After completing the AN0 pin conversion, the AN1 pin is not sampled and converted until the second $\overline{\text{ADTRG}}$ pin falling edge is generated. When the second $\overline{\text{ADTRG}}$ falling edge is generated, the single sweep conversion of the pins after the AN1 pin is restarted. **Table 15.12** shows the delayed trigger mode 1 specifications. **Figure 15.24** shows the operation example of delayed trigger mode 1. **Figure 15.25** and **15.26** show each flag operation in the ADSTAT0 register that corresponds to the operation example. **Figure 15.27** shows registers ADCON0 to ADCON2 in delayed trigger mode 1. **Figure 15.28** shows the ADTRGCON register in delayed trigger mode 1. **Table 15.13** shows the trigger select bit setting in delayed trigger mode 1.

Table 15.12 Delayed Trigger Mode 1 Specifications

Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltages applied to the selected pins are converted one-by-one to a digital code. At this time, the $\overline{\text{ADTRG}}$ pin falling edge starts AN0 pin conversion and the second $\overline{\text{ADTRG}}$ pin falling edge starts conversion of the pins after AN1 pin
A/D Conversion Start Condition	AN0 pin conversion start condition The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" (falling edge) (1) AN1 pin conversion start condition (2) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" (falling edge) •When the second $\overline{\text{ADTRG}}$ pin falling edge is generated during A/D conversion of the AN0 pin, input voltage of AN1 pin is sampled or after at the time of $\overline{\text{ADTRG}}$ falling edge. The conversion of AN1 and the rest of the sweep starts when AN0 conversion is completed. •When the $\overline{\text{ADTRG}}$ pin falling edge is generated again during single sweep conversion of pins after the AN1 pin, the conversion is not affected
A/D Conversion Stop Condition	•A/D conversion completed •Set the ADST bit to 0 (A/D conversion halted) (3)
Interrupt Request Generation Timing	Single sweep conversion completed
Analog Input Pin	Select from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins) and AN0 to AN7 (8 pins) (4)
Readout of A/D Conversion Result	Readout one of registers AN0 to AN7 that corresponds to the selected pins

NOTES:

1. Do not generate the next $\overline{\text{ADTRG}}$ pin falling edge after the AN1 pin conversion is started until all selected pins complete A/D conversion. When an $\overline{\text{ADTRG}}$ pin falling edge is generated again during A/D conversion, its trigger is ignored. The falling edge of $\overline{\text{ADTRG}}$ pin, which was input after all selected pins complete A/D conversion, is considered to be the next AN0 pin conversion start condition.
2. The $\overline{\text{ADTRG}}$ pin falling edge is detected synchronized with the operation clock fAD. Therefore, when the $\overline{\text{ADTRG}}$ pin falling edge is generated in shorter periods than fAD, the second $\overline{\text{ADTRG}}$ pin falling edge may not be detected. Do not generate the $\overline{\text{ADTRG}}$ pin falling edge in shorter periods than fAD.
3. Do not write 1 (A/D conversion started) to the ADST bit in delayed trigger mode 1. When write 1,unexpected interrupts may be generated.
4. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. However, all input pins need to belong to the same group.

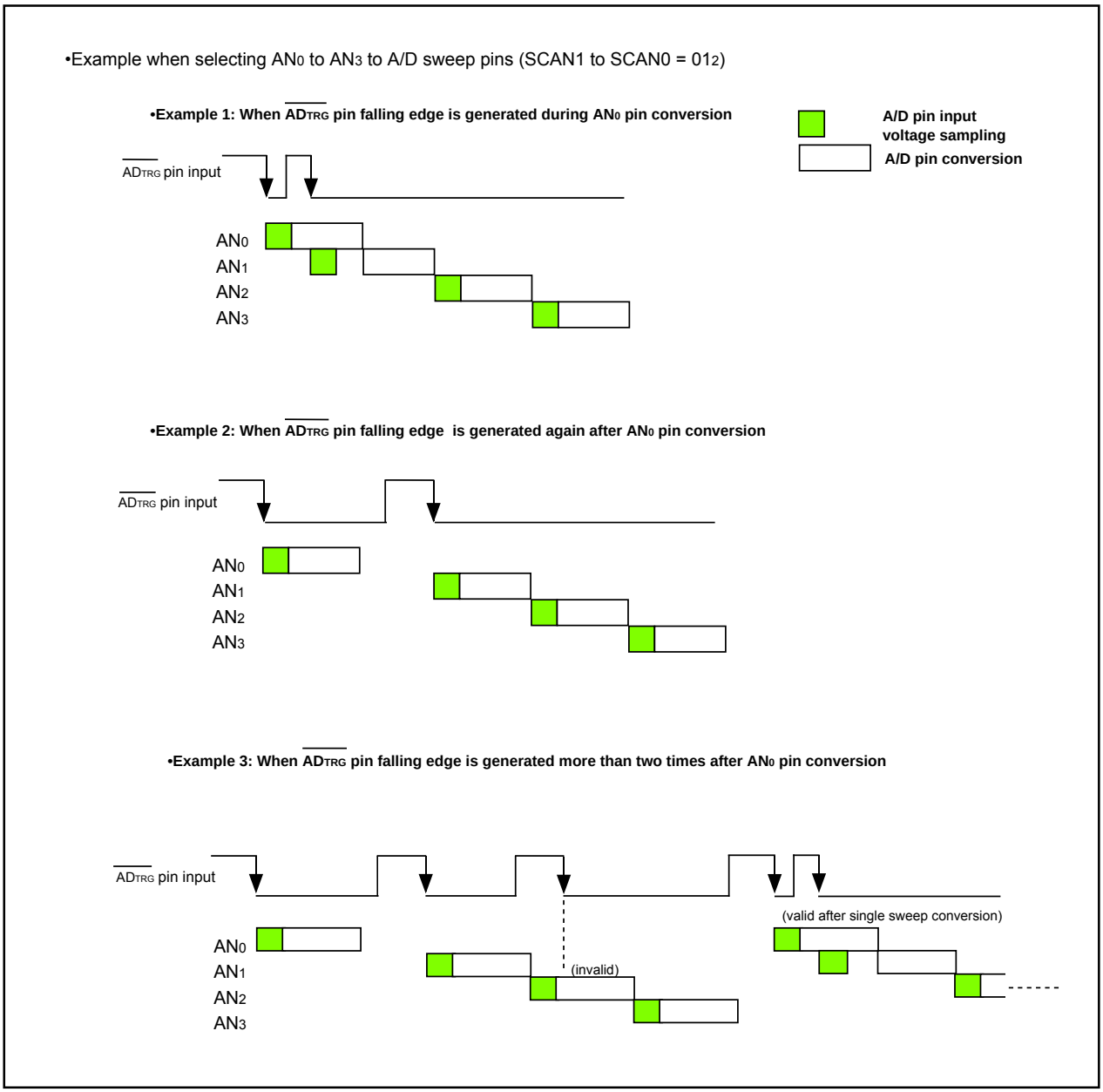
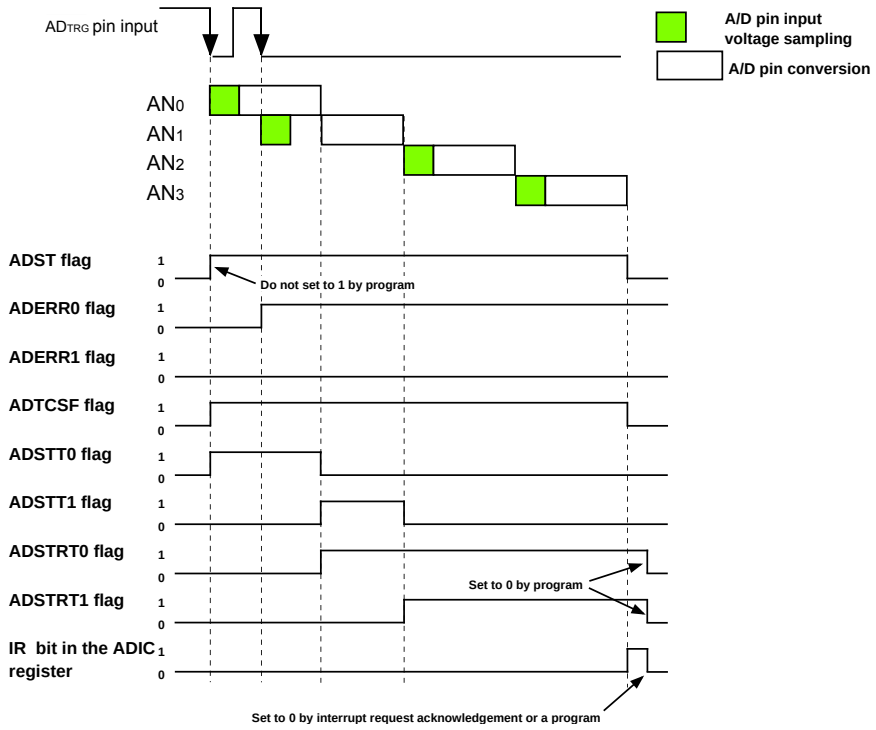


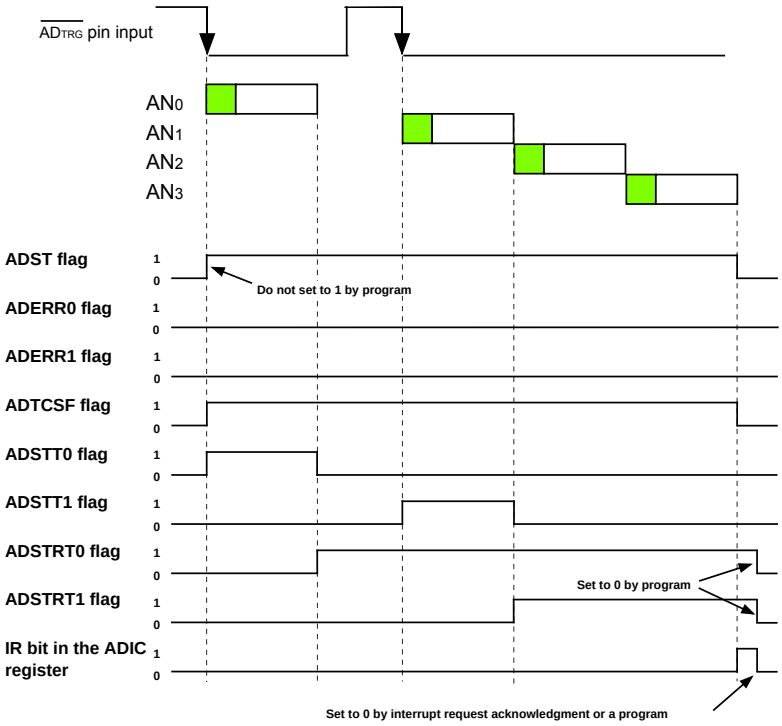
Figure 15.24 Operation Example in Delayed Trigger Mode1

•Example when selecting AN_0 to AN_3 to A/D sweep pins (SCAN1 to SCAN0 = 01_2)

•Example 1: When AD_{TRG} pin falling edge is generated during AN_0 pin conversion



•Example 2: When $\overline{AD_{TRG}}$ pin falling edge is generated again after AN_0 pin conversion



ADST flag: Bit 6 in the ADCON0 register
ADERR0, ADERR1, ADTCSF, ADSTT0, ADSTT1, ADSTRT0 and ADSTRT1 flag: bits 0, 1, 3, 4, 5, 6 and 7 in the ADSTAT0 register

Figure 15.25 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (1)

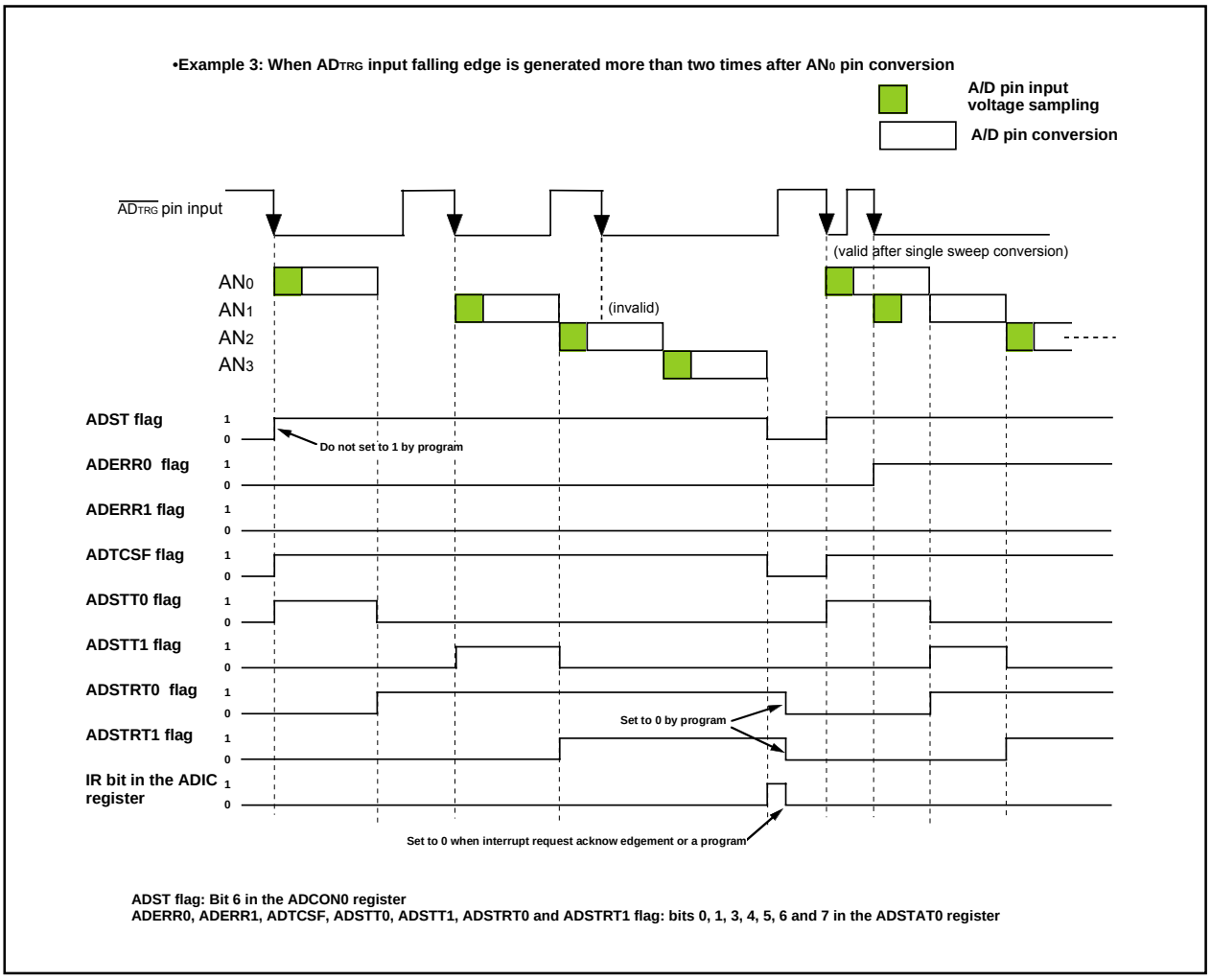


Figure 15.26 Each Flag Operation in ADSTAT0 Register Associated with the Operation Example in Delayed Trigger Mode 1 (2)

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
		0	0	0	1	1	1	ADCON0	03D6$_{16}$	00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	b2 b1 b0 1 1 1: Set to 111b in delayed trigger mode 0	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 0 0: One-shot mode or delayed trigger mode 0, 1	RW
MD1			RW
TRG	Trigger select bit	See **Table 15.13**	RW
ADST	A/D conversion start flag(2)	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. Do not write 1 in delayed trigger mode 0. When write, set to 0.

A/D Control Register 1(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	1			0			ADCON1	03D7$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit (2)	When delayed triger mode 1 is selected, b1 b0 0 0: AN0 to AN1 (2 pins) 0 1: AN0 to AN3 (4 pins) 1 0: AN0 to AN5 (6 pins) 1 1: AN0 to AN7 (8 pins)	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (3)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 μs or more before starting A/D conversion.

A/D Control Register 2(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	1		0			1	ADCON2	03D4$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit (2)	1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	See **Table 15.13**	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.
2. Set to 1 in delayed trigger mode 1.

Figure 15.27 ADCON0 to ADCON2 Registers in Delayed Trigger Mode 1

A/D Trigger Control Register (1)

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

Symbol	Address	After Reset
ADTRGCON	$03D2_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
SSE	A/D operation mode select bit 2	Simultaneous sample sweep mode or delayed trigger mode 0, 1	RW
DTE	A/D operation mode select bit 3	Delayed trigger mode 0, 1	RW
HPTRG0	AN_0 trigger select bit	See **Table 15.13**	RW
HPTRG1	AN_1 trigger select bit	See **Table 15.13**	RW
$\overline{\text{(b7-b4)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If ADTRGCON is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.28 ADTRGCON Register in Delayed Trigger Mode 1

Table 15.13 Trigger Select Bit Setting in Delayed Trigger Mode 1

TRG	TRG1	HPTRG0	HPTRG1	Trigger
0	1	0	0	$\overline{AD_{TRG}}$

15.2 Resolution Select Function

The BITS bit in the ADCON1 register determines the resolution. When the BITS bit is set to 1 (10-bit precision), the A/D conversion result is stored into bits 0 to 9 in the ADi register (i=0 to 7). When the BITS bit is set to 0 (8-bit precision), the A/D conversion result is stored into bits 7 to 0 in the ADi register.

15.3 Sample and Hold

When the SMP bit in the ADCON 2 register is set to 1 (with the sample and hold function), A/D conversion rate per pin increases to 28 ϕAD cycles for 8-bit resolution or 33 ϕAD cycles for 10-bit resolution. The sample and hold function is available in one-shot mode, repeat mode, single sweep mode, repeat sweep mode 0 and repeat sweep mode 1. In these modes, start A/D conversion after selecting whether the sample and hold circuit is to be used or not. In simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode, set to use the Sample and Hold function before starting A/D conversion.

15.4 Power Consumption Reducing Function

When the A/D converter is not used, the VCUT bit in the ADCON1 register isolates the resistor ladder of the A/D converter from the reference voltage input pin (V_{REF}). Power consumption is reduced by shutting off any current flow into the resistor ladder from the V_{REF} pin.

When using the A/D converter, set the VCUT bit to 1 (Vref connected) before setting the ADST bit in the ADCON0 register to 1 (A/D conversion started). Do not set the ADST bit and VCUT bit to 1 simultaneously, nor set the VCUT bit to 0 (Vref unconnected) during A/D conversion.

15.5 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in **Figure 15.29** has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, MCU's internal resistance be R, precision (error) of the A/D converter be X, and the A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

$$\text{VC is generally } VC = VIN\{1-e^{-\frac{1}{C(R0+R)}t}\}$$

$$\text{And when } t = T,\quad VC = VIN - \frac{X}{Y}VIN = VIN\left(1-\frac{X}{Y}\right)$$

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

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

$$\text{Hence,}\quad R0 = -\frac{T}{C\cdot\ln\frac{X}{Y}} - R$$

Figure 15.29 shows analog input pin and externalsensor equivalent circuit. When the difference between VIN and VC becomes 0.1 LSB, we find impedance R0 when voltage between pins. VC changes from 0 to VIN-(0.1/1024) VIN in timer T. (0.1/1024) means that A/D precision drop due to insufficient capacitor chage is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When f(XIN) = 10MHz, T=0.3µs in the A/D conversion mode with sample & hold. Output inpedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3µs, R = 7.8kΩ, C = 1.5pF, X = 0.1, and Y = 1024. Hence,

$$R0 = -\frac{0.3\times10^{-6}}{1.5\times10^{-12}\cdot\ln\frac{0.1}{1024}} - 7.8\times10^{3} \cong 13.9\times10^{3}$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out of be approximately 13.9kΩ.

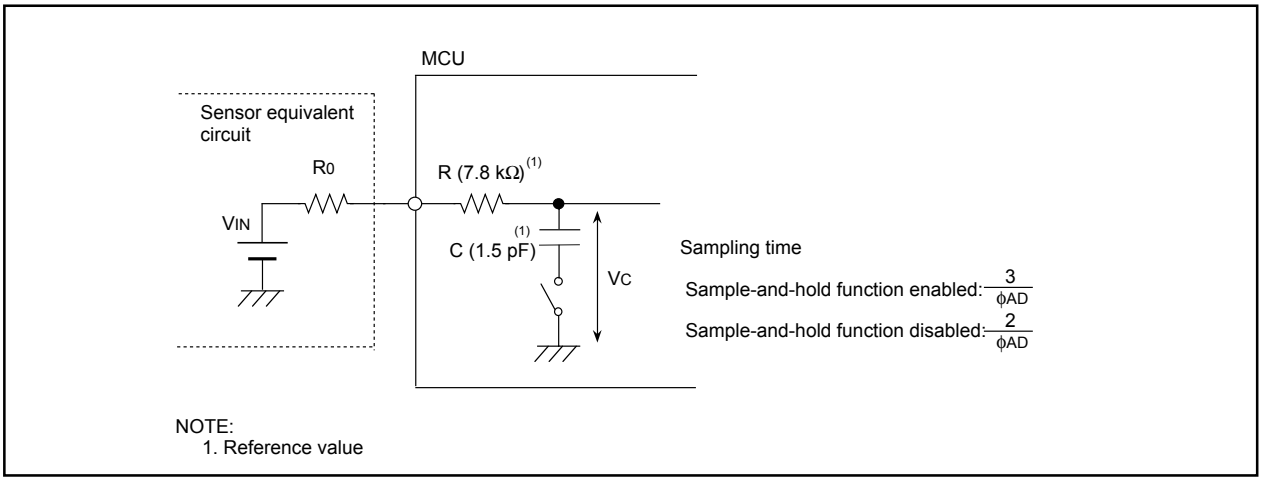


Figure 15.29 Analog Input Pin and External Sensor Equivalent Circuit

16. Multi-master I2C bus Interface

The multi-master I2C bus interface is a serial communication circuit based on Philips I2C bus data transfer format, equipped with arbitration lost detection and synchronous functions. **Figure 16.1** shows a block diagram of the multi-master I2C bus interface and **Table 16.1** lists the multi-master I2C bus interface functions.

The multi-master I2C bus interface consists of the S0D0 register, the S00 register, the S20 register, the S3D0 register, the S4D0 register, the S10 register, the S2D0 register and other control circuits.

Figures 16.2 to **16.8** show the registers associated with the multi-master I2C bus.

Table 16.1 Multi-master I2C bus interface functions

Item	Function
Format	Based on Philips I2C bus standard: 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	Based on Philips I2C bus standard: Master transmit Master receive Slave transmit Slave receive
SCL clock frequency	16.1kHz to 400kHz (at VIIC (1)= 4MHz)
I/O pin	Serial data line SDAMM(SDA) Serial clock line SDLMM(SCL)

NOTE:

1. VIIC=I2C system clock

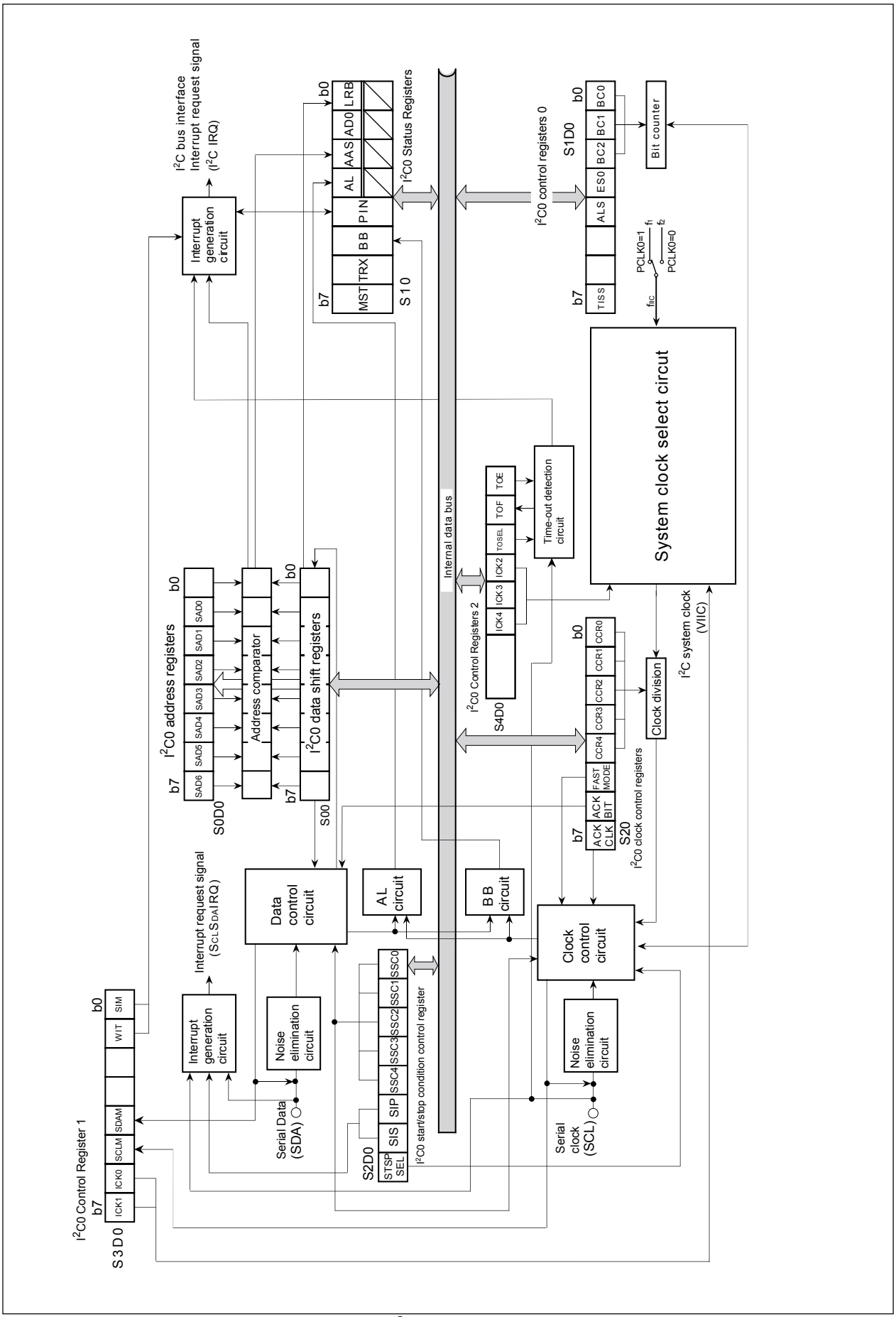
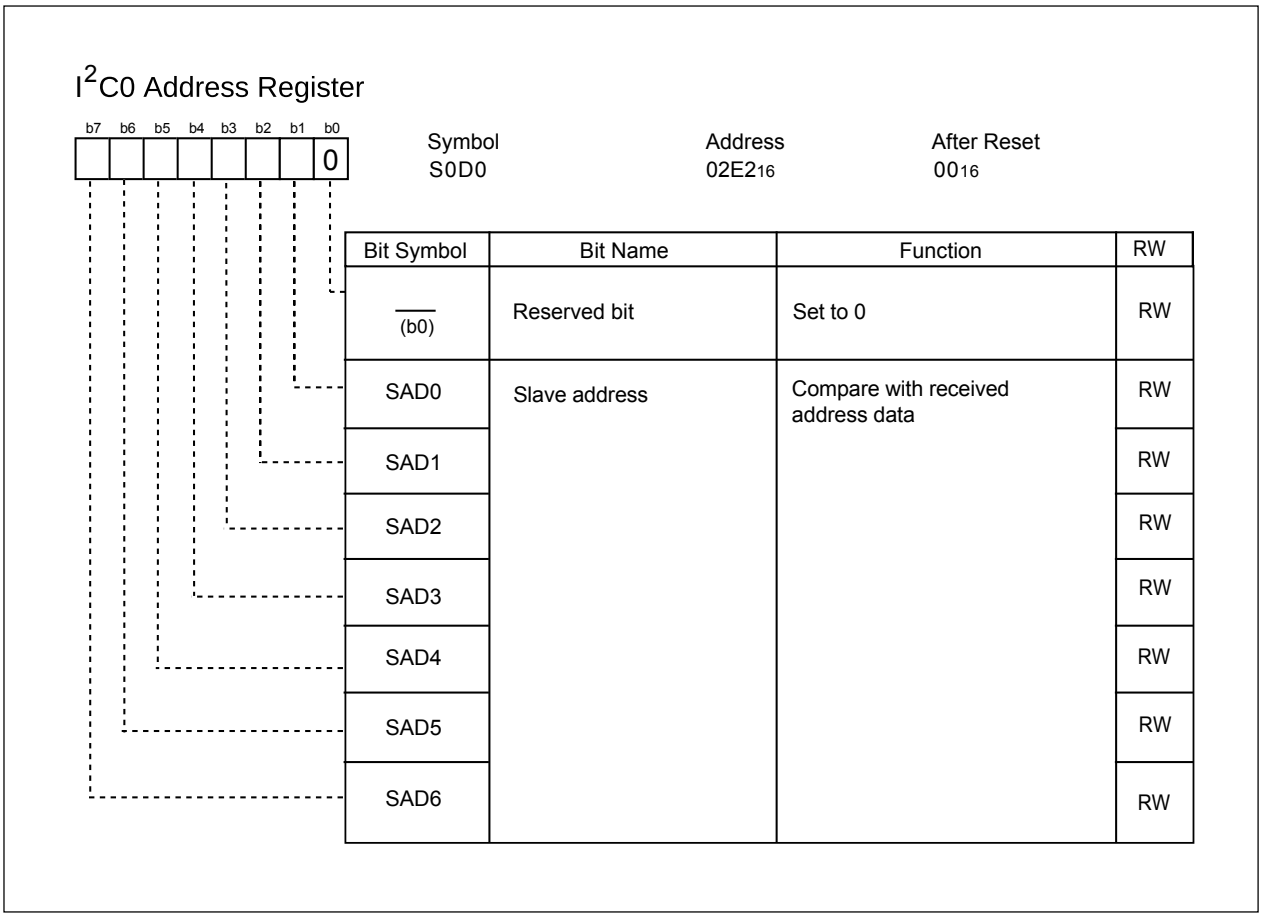


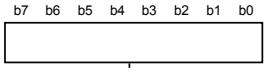
Figure 16.1 Block diagram of multi-master I2C bus interface



Bit Symbol	Bit Name	Function	RW
(b0)	Reserved bit	Set to 0	RW
SAD0	Slave address	Compare with received address data	RW
SAD1			RW
SAD2			RW
SAD3			RW
SAD4			RW
SAD5			RW
SAD6			RW

Figure 16.2 S0D0 Register

I2C0 Data Shift Register

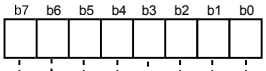


Symbol	Address	After Reset
S00	02E016	XX16

Function	RW
Transmit/receive data are stored. In master transmit mode, the start condition/stop condition are triggered by writing data to the register (refer to **16.9 START Condition Generation Method** and **16.11 STOP Condition Generation Method**). Start transmitting/receiving data while synchronizing with SCL	RW(1)

NOTE:
1. Write is enabled only when the ES0 bit in the S1D0 register is 1 (I2C bus interface is enabled). Write the transmit data after the receive data is read because the S00 register is used to store both the transmit and receive data. When the S00 register is set, bits BC2 to BC0 in the S1D0 register are set to 0002, while bits LRB, AAS, and AL in the S10 register are set to 0 respectively.

I2C0 Clock Control Register



Symbol	Address	After Reset
S20	02E416	0016

Bit Symbol	Bit Name	Function	RW
CCR0	SCL Frequency Control Bits	See **Table 16.3**	RW
CCR1			RW
CCR2			RW
CCR3			RW
CCR4			RW
FAST MODE	SCL Mode Specification Bit	0: Standard clock mode 1: High-speed clock mode	RW
ACKBIT	ACK Bit	0: ACK is returned 1: ACK is not returned	RW
ACK-CLK	ACK Clock Bit	0: No ACK clock 1: With ACK clock	RW

Figure 16.3 S00 and S20 Registers

I2C0 Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	After Reset
S1D0	02E316	0016

Bit Symbol	Bit Name	Function	RW
BC0	Bit counter (Number of transmit/receive bits)(1)	b2 b1 b0 0 0 0: 8 0 0 1: 7 0 1 0: 6 0 1 1: 5 1 0 0: 4 1 0 1: 3 1 1 0: 2 1 1 1: 1	RW
BC1			RW
BC2			RW
ES0	I2C bus interface enable bit	0: Disabled 1: Enabled	RW
ALS	Data format select bit	0: Addressing format 1: Free data format	RW
$\overline{(b5)}$	Reserved bit	Set to 0	RW
IHR	I2C bus interface reset bit	0: Reset release (automatic) 1: Reset	RW
TISS	I2C bus interface pin input level select bit	0: I2C bus input 1: SMBUS input	RW

NOTE:
1.In the following status, the bit counter is set to 000 automatically
- •Start condition/stop condition are detected
- •Immediately after the completion of 1-byte data transmit
- •Immediately after the completion of 1-byte data receive

Figure 16.4 S1D0 Register

I2C0 Status Register

Symbol	Address	After Reset
S10	$02E8_{16}$	$0001000X_2$

Bit Symbol	Bit Name	Function	RW
LRB	Last receive bit	0: Last bit = 0 1: Last bit = 1	RO(1)
ADR0	General call detecting flag	0: No general call detected 1: General call detected	RO(1)
AAS	Slave address comparison flag	0: No address matched 1: Address matched	RO(1)
AL	Arbitration lost detection flag	0: Not detected 1: Detected	RO(2)
PIN	I2C bus interface interrupt request bit	0: Interrupt request issued 1: No interrupt request issued	RO(2)
BB	Bus busy flag	0: Bus free 1: Bus busy	RO(1)
TRX	Communication mode select bits 0	0: Receive mode 1: Transmit mode	RW(3)
MST	Communication mode select bit 1	0: Slave mode 1: Master mode	RW(3)

NOTES:
1. This bit is read only if it is used for the status check.
   To write to this bit, refer to **16.9 START Condition Generation Method** and **16.11 STOP Condition Generation Method**.
2. Read only. When write, set to 0.
3. To write to these bits, refer to **16.9 START Condition Generation Method** and **16.11 STOP Condition Generation Method.**

Figure 16.5 S10 Register

I2C0 Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
S3D0	02E616	001100002

Bit Symbol	Bit Name	Function	RW
SIM	The interrupt enable bit for STOP condition detection	0: Disable the I2C bus interface interrupt of STOP condition detection 1: Enable the I2C bus interface interrupt of STOP condition detection	RW
WIT	The interrupt enable bit for data receive completion	0: Disable the I2C bus interface interrupt of data receive completion 1: Enable the I2C bus interface interrupt of data receive completion When setting NACK (ACK bit = 0), write 0	RW
PED	SDA/port function switch bit(1)	0: SDA I/O pin 1: Port output pin	RW
PEC	SCL/port function switch bit(1)	0: SCL I/O pin 1: Port output pin	RW
SDAM	The logic value monitor bit of SDA output	0: SDA output logic value = 0 1: SDA output logic value = 1	RO
SCLM	The logic value monitor bit of SCL output	0: SCL output logic value = 0 1: SCL output logic value = 1	RO
ICK0	I2C bus system clock selection bits, if bits ICK4 to ICK2 in the S4D0 register is 0002	b7 b6 0 0 : VIIC =1/2 fIIC 0 1 : VIIC =1/4 fIIC 1 0 : VIIC =1/8 fIIC 1 1 : Reserved (2)	RW
ICK1			RW

NOTE:
1. Bits PED and PEC are enabled when the ES0 bit in the S1D0 register is set to 1 (I2C bus interface enabled).
2. When the PCLK0 bit in the PCLKR register is set to 0, fIIC=f2. When the PCLK0 bit in the PCLKR register is set to 1, fIIC=f1.

Figure 16.6 S3D0 Register

I2C0 Control Register 2

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol	Address	After Reset
S4D0	02E716	0016

Bit Symbol	Bit Name	Function	RW
TOE	Time out detection function enable bit	0: Disabled 1: Enabled	RW
TOF	Time out detection flag	0: Not detected 1: Detected	RO
TOSEL	Time out detection time select bit	0: Long time 1: Short time	RW
ICK2	I2C bus system clock select bits	b5 b4 b3 0 0 0 VIIC set by ICK1 and ICK0 bits in S3D0 register 0 0 1 VIIC = 1/2.5 fIIC 0 1 0 VIIC = 1/3 fIIC 0 1 1 VIIC = 1/5 fIIC 1 0 0 VIIC = 1/6 fIIC (1)	RW
ICK3			RW
ICK4			RW
(b6)	Reserved bit	Set to 0	RW
SCPIN	STOP condition detection interrupt request bit	0: No I2C bus interface interrupt request 1: I2C bus interface interrupt request	RW

NOTE:
1. When the PCLK0 bit in the PCLKR register is set to 0, fIIC = f2. When the PCLK0 bit is set to 1, fIIC=f1.

Figure 16.7 S4D0 Register

I2C0 Start/stop Condition Control Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: S2D0 | Address: 02E516 | After Reset: 000110102

Bit Symbol	Bit Name	Function	RW
SSC0	START/STOP condition setting bits(1)	Setting for detection condition of START/STOP condition. See **Table 16.2**	RW
SSC1			RW
SSC2			RW
SSC3			RW
SSC4			RW
SIP	SCL/SDA interrupt pin polarity select bit	0: Active in falling edge 1: Active in rising edge	RW
SIS	SCL/SDA interrupt pin select bit	0: SDA enabled 1: SCL enabled	RW
STSPSEL	START/STOP condition generation select bit	0: Short setup/hold time mode 1: Long setup/hold time mode	RW

NOTE:
1. Do not set 000002 or odd values.

Figure 16.8 S2D0 Register

Table 16.2 Recommended setting (SSC4-SSC0) start/stop condition at each oscillation frequency

Oscillation f1 (MHz)	I2C bus system clock select	I2C bus system clock(MHz)	SSC4-SSC0(1)	SCL release time (cycle)	Setup time (cycle)	Hold time (cycle)
10	1 / 2f1(2)	5	XXX11110	6.2 µs (31)	3.2 µs (16)	3.0 µs (15)
8	1 / 2f1(2)	4	XXX11010	6.75 µs(27)	3.5 µs (14)	3.25 µs(13)
			XXX11000	6.25 µs(25)	3.25 µs (13)	3.0 µs (12)
8	1 / 8f1(2)	1	XXX00100	5.0 µs (5)	3.0 µs (3)	2.0 µs (2)
4	1 / 2f1(2)	2	XXX01100	6.5 µs (13)	3.5 µs (7)	3.0 µs (6)
			XXX01010	5.5 µs (11)	3.0 µs (6)	2.5 µs (5)
2	1 / 2f1(2)	1	XXX00100	5.0 µs (5)	3.0 µs (3)	2.0 µs (2)

NOTES:
1. Do not set odd values or 000002 to START/STOP condition setting bits (SSC4 to SSC0)
2. When the PCLK0 bit in the PCLKR register is set to 1.

16.1 I2C0 Data Shift Register (S00 register)

The S00 register is an 8-bit data shift register to store a received data and to write a transmit data. When a transmit data is written to the S00 register, the transmit data is synchronized with a SCL clock and the data is transferred from bit 7. Then, every one bit of the data is transmitted, the register's content is shifted for one bit to the left. When the SCL clock and the data is imported into the S00 register from bit 0. Every one bit of the data is imported, the register's content is shifted for one bit to the left. **Figure 16.9** shows the timing to store the receive data to the S00 register.

The S00 register can be written when the ES0 bit in the S1D0 register is set to 1 (I2C0 bus interface enabled). If the S00 register is written when the ES0 bit is set to 1 and the MST bit in the S10 register is set to 1 (master mode), the bit counter is reset and the SCL clock is output. Write to the S00 register when the START condition is generatedor when an "L" signal is applied to the SCL pin. The S00 register can be read anytime regardless of the ES0 bit value.

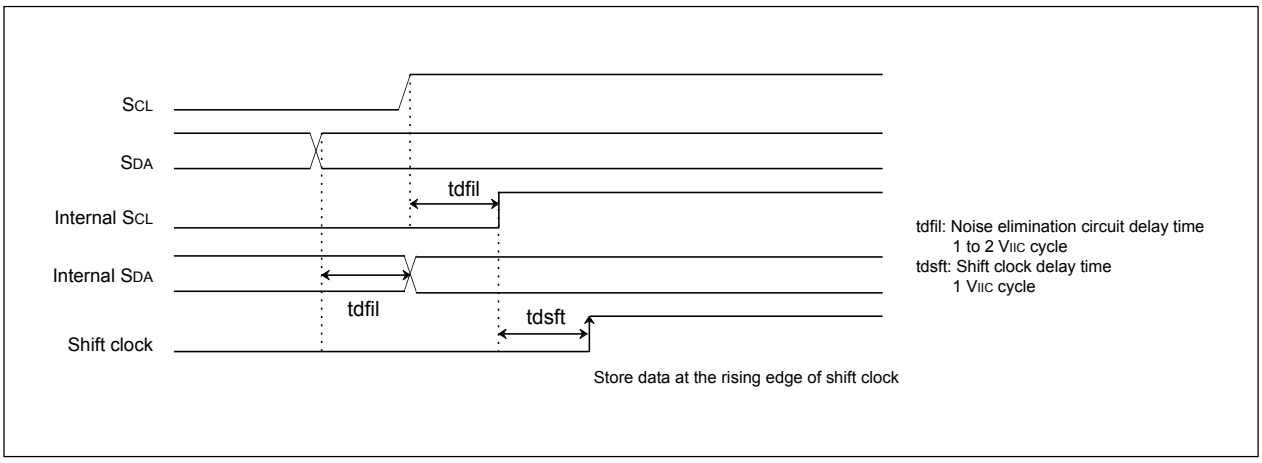


Figure 16.9 The Receive Data Storing Timing of S00 Register

16.2 I2C0 Address Register (S0D0 register)

The S0D0 register consists of bits SAD6 to SAD0, total of 7. At the addressing is formatted, slave address is detected automatically and the 7-bit received address data is compared with the contents of bits SAD6 to SAD0.

16.3 I2C0 Clock Control Register (S20 register)

The S20 register is used to set theACK control, SCL mode and the SCL frequency.

16.3.1 Bits 0 to 4: SCL Frequency Control Bits (CCR0–CCR4)

These bits control the SCL frequency. See **Table 16.3** .

16.3.2 Bit 5: SCL Mode Specification Bit (FAST MODE)

The FAST MODE bit selects SCL mode. When the FAST MODE bit is set to 0, standard clock mode is entered. When it is set to 1, high-speed clock mode is entered.

When using the high-speed clock mode I2C bus standard (400 kbits/s maximum) to connect buses, set the FAST MODE bit to 1 (select SCL mode as high-speed clock mode) and use the I2C bus system clock (VIIC) at 4 MHz or more frequency.

16.3.3 Bit 6: ACK Bit (ACKBIT)

The ACKBIT bit sets the SDA status when an ACK clock(1) is generated. When the ACKBIT bit is set to “0”, ACK is returned and te clock applied to SDA becomes "L" when ACK clock is generated. When it is set to 1, ACK is not returned and the clock clock applied to SDA maintains "H" at ACK clock generation.

When the ACKBIT bit is set to 0, the address data is received. When the slave address matches with the address data, SDA becomes "L" automatically (ACK is returned). When the slave address and the address data are not matched, SDA becomes "H" (ACK is not returned).

NOTE:

1. ACK clock: Clock for acknowledgment

16.3.4 Bit 7: ACK Clock Bit (ACK-CLK)

The ACK-CLK bit set a clock for data transfer acknowledgement. When the ACK-CLK bit is set to 0, ACK clock is not generated after data is transferred. When it is set to 1, a master generates ACK clock every one-bit data transfer is completed. The device, which transmits address data and control data, leave SDA pin open (apply "H" signal to SDA) when ACK clock is generated. The device which receives data, receives the generated ACKBIT bit.

NOTE:

1.Do not rewrite the S20 register, other than the ACKBIT bit during data transfer. If data is written to other than the ACKBIT bit during transfer, the I2C bus clock circuit is reset and the data may not be transferred successfully.

Table 16.3 Setting values of S20 register and SCL frequency

Setting value of CCR4 to CCR0					SCL frequency (at VIIC=4MHz, unit : kHz) (1)	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clock mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	- (2)	333
0	0	1	0	0	- (2)	250
0	0	1	0	1	100	400 (3)
0	0	1	1	0	83.3	166
↓	↓	↓	↓	↓	500 / CCR value (3)	1000 / CCR value (3)
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

NOTES:

1. The duty of the SCL clock output is 50 %. The duty becomes 35 to 45 % only when high-speed clock mode is selected and the CCR value = 5 (400 kHz, at VIIC = 4 MHz). "H" duration of the clock fluctuates from –4 to +2 I2C system clock cycles in standard clock mode, and fluctuates from –2 to +2 I2C system clock cycles in high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because the "L" is extended instead of "H" reduction. These are the values when the SCL clock synchronization by the synchronous function is not performed. The CCR value is the decimal notation value of the CCR4 to CCR0 bits.

2. Each value of the SCL frequency exceeds the limit at VIIC = 4 MHz or more. When using these setting values, use VIIC = 4 MHz or less. Refer to **Figure 16.6**.

3. The data formula of SCL frequency is described below:

VIIC/(8 x CCR value) Standard clock mode

VIIC/(4 x CCR value) High-speed clock mode (CCR value ≠ 5)

VIIC/(2 x CCR value) High-speed clock mode (CCR value = 5)

Do not set 0 to 2 as the CCR value regardless of the VIIC frequency. Set 100 kHz (max.) in standard clock mode and 400 kHz (max.) in high-speed clock mode to the SCL frequency by setting the CCR4 to CCR0 bits.

16.4 I2C0 Control Register 0 (S1D0)

The S1D0 register controls data communication format.

16.4.1 Bits 0 to 2: Bit Counter (BC0–BC2)

Bits BC2 to BC0 decide how many bits are in one byte data transferred next. After the selected numbers of bits are transferred successfully, I2C bus interface interrupt request is gnerated and bits BC2 to BC0 are reset to 0002. At this time, if the ACK-CLK bit in the S20 register is set to 1 (with ACK clock), one bit for ACK clock is added to the numbers of bits selected by the BC2 to BC0 bits.

In addition, bits BC2 to BC0 become 0002 even though the START condition is detected and the address data is transferred in 8 bits.

16.4.2 Bit 3: I2C Interface Enable Bit (ES0)

The ES0 bit enables to use the multi-master I2C bus interface. When the ES0 bit is set to 0, I2C bus interface is disabled and the SDA and SCL pins are placed in a high-h-impedance state. When the ES0 bit is set to 1, the interface is enabled.

When the ES0 bit is set to 0, the process is followed.

1)The bits in the S10 register are set as MST = 0, TRX = 0, PIN = 1, BB = 0, AL = 0, AAS = 0, ADR0 = 0

2)The S00 register cannot be written.

3)The TOF bit in the S4D0 register is set to 0 (time-out detection flag is not detected)

4)The I2C system clock (VIIC) stops counting while the internal counter and flags are reset.

16.4.3 Bit 4: Data Format Select Bit (ALS)

The ALS bit determines whether the salve address is recognized. When the ALS bit is set to 0, an addressing format is selected and a address data is recognized. Only if the comparison is matched between the slave address stored into the S0D0 register and the received address data or if the general call is received, the data is transferred. When the ALS bit is set to 1, the free data format is selected and the slave address is not recognized.

16.4.4 Bit 6: I2C bus Interface Reset Bit (IHR)

The IHR bit is used to reset the I2C bus interface circuit when the error communication occurs.

When the ES0 bit in the S1D0 register is set to 1 (I2C bus interface is enabled), the hardware is reset by writing 1 to the IHR bit. Flags are processed as follows:

1)The bits in the S10 register are set as MST = 0, TRX = 0, PIN to 1, BB = 0, AL = 0, AAS = 0, and ADR0 = 0

2)The TOF bit in the S4D0 register is set to 0 (time-out detection flag is not detected)

3)The internal counter and flags are reset.

The I2C bus interface circuit is reset after 2.5 VIIC cycles or less, and the IHR bit becomes 0 automatically by writing 1 to the IHR bit. **Figure 16.10** shows the reset timing.

16.4.5 Bit 7: I2C bus Interface Pin Input Level Select Bit (TISS)

The TISS bit selects the input level of the SCL and SDA pins for the multi-master I2C bus interface. When the TISS bit is set to 1, the P20 and P21 become the SMBus input level.

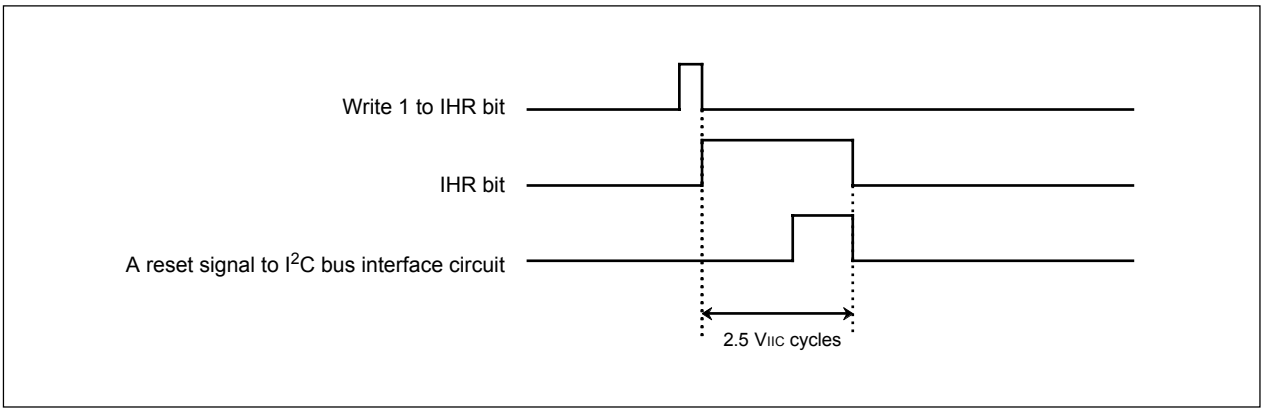


Figure 16.10 The timing of reset to the I2C bus interface circuit

16.5 I2C0 Status Register (S10 register)

The S10 register monitors the I2C bus interface status. When using the S10 register to check the status, use the 6 low-order bits for read only.

16.5.1 Bit 0: Last Receive Bit (LRB)

The LRB bit stores the last bit value of received data. It can also be used to confirm whether ACK is received. If the ACK-CLK bit in the S20 register is set to 1 (with ACK clock) and ACK is returned when the ACK clock is generated, the LRB bit is set to 0. If ACK is not returned, the LRB bit is set to 1. When the ACK-CLK bit is set to 0 (no ACK clock), the last bit value of received data is input. When writing data to the S00 register, the LRB bit is set to 0.

16.5.2 Bit 1: General Call Detection Flag (ADR0)

When the ALS bit in the S1D0 register is set to 0 (addressing format), this ADR0 flag is set to 1 by receiving the general calls(1),whose address data are all 0, in slave mode.
The ADR0 flag is set to 0 when STOP or START conditions is detected or when the IHR bit in the S1D0 register is set to 1 (reset).

NOTE:
1. General call: A master device transmits the general call address 00_{16} to all slaves. When the master device transmits the general call, all slave devices receive the controlled data after general call.

16.5.3 Bit 2: Slave Address Comparison Flag (AAS)

The AAS flag indicates a comparison result of the slave address data after enabled by setting the ALS bit in the S1D0 register to 0 (addressing format).
The AAS flag is set to 1 when the 7 bits of the address data are matched with the slave address stored into the S0D0 register, or when a general call is received, in slave receive mode. The AAS flag is set to 0 by writing data to the S00 register. When the ES0 bit in the S1D0 register is set to 0 (I2C bus interface disabled) or when the IHR bit in the S1D0 register is set to 1 (reset), the AAS flag is also set to 0.

16.5.4 Bit 3: Arbitration Lost Detection Flag (AL)(1)

In master transmit mode, if an "L" signal is applied to the SDA pin by other than the MCU, the AL flag is set to 1 by determining that the arbitration is los and the TRX bit in the S10 register is set to 0 (receive mode) at the same time. The MST bit in the S10 register is set to 0 (slave mode) after transferring the bytes which lost the arbitration.
The arbitration lost can be detected only in master transmit mode. When writing data to the S00 register, the AL flag is set to 0. When the ES0 bit in the S1D0 register is set to 0 (I2C bus interface disabled) or when the IHR bit in the S1D0 register is set to 1 (reset), the AL flag is set to 0.

NOTE:
1. Arbitration lost: communication disabled as a master

16.5.5 Bit 4: I2C bus Interface Interrupt Request Bit (PIN)

The PIN bit generates an I2C bus interface interrupt request signal. Every one byte data is ransferred, the PIN bit is changed from 1 to 0. At the same time, an I2C bus interface interrupt request is generated. The PIN bit is synchronized with the last clock of the internal transfer clock (when ACK-CLK=1, the last clock is the ACK clock: when the ACK-CLK=0, the last clock is the 8th clock) and it becomes 0. The interrupt request is generated on the falling edge of the PIN bit. When the PIN bit is set to 0, the clock applied to SCL maintains "L" and further clock generation is disabled. When the ACK-CLK bit is set to 1 and the WIT bit in the S3D0 register is set to 1 (enable the I2C bus interface interrupt of data receive completion). The PIN bit is synchronized with the last clock and the falling edge of the ACK clock. Then, the PIN bit is set to 0 and I2C bus interface interrupt request is generated. **Figure 16.11** shows the timing of the I2C bus interface interrupt request generation.

The PIN bit is set to 1 in one of the following conditions:

- •When data is written to the S00 register
- •When data is written to the S20 register (when the WIT bit is set to 1 and the internal WAIT flag is set to 1)
- •When the ES0 bit in the S1D0 register is set to 0 (I2C bus interface disabled)
- •When the IHR bit in the S1D0 register is set to 1(reset)

The PIN bit is set to 0 in one of the following conditions:

- •With completion of 1-byte data transmit (including a case when arbitration lost is detected)
- •With completion of 1-byte data receive
- •When the ALS bit in the S1D0 register is set to 0 (addressing format) and slave address is matched or general call address is received successfully in slave receive mode
- •When the ALS bit is set to 1 (free format) and the address data is received successfully in slave receive mode

16.5.6 Bit 5: Bus Busy Flag (BB)

The BB flag indicates the operating conditions of the bus system. When the BB flag is set to 0, a bus system is not in use and a START condition can be generated. The BB flag is set and reset based on an input signal of the SCL and SDA pins either in master mode or in slave mode. When the START condition is detected, the BB flag is set to 1. On the other hand, when the STOP condition is detected, the BB flag is set to 0. Bits SSC4 to SSC0 in the S2D0 register decide to detect between the START condition and the STOP condition. When the ES0 bit in the S1D0 register is set to 0 (I2C bus interface disabled) or when the IHR bit in the S1D0 register is set to 1 (reset), the BB flag is set to 0. Refer to **16.9 START Condition Generation Method and 16.11 STOP Condition Generation Method**.

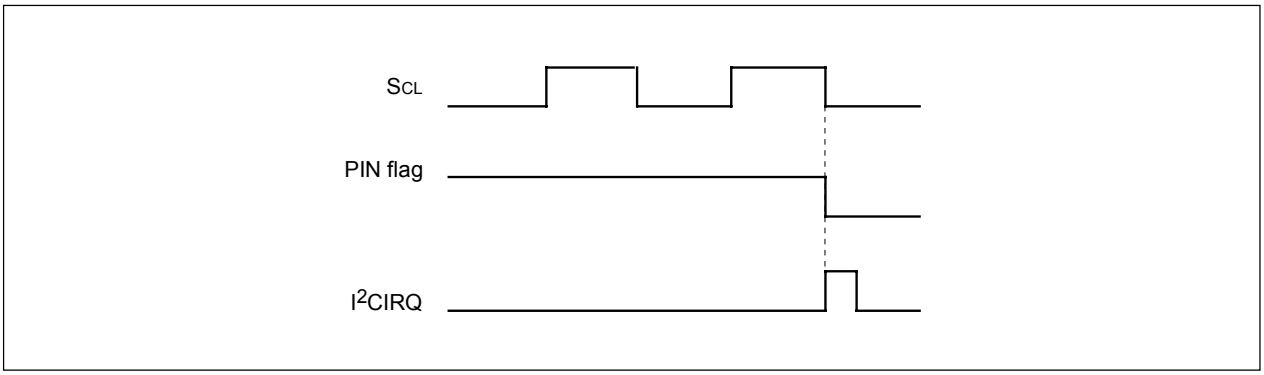


Figure 16.11 Interrupt request signal generation timing

16.5.7 Bit 6: Communication Mode Select Bit (Transfer Direction Select Bit: TRX)

This TRX bit decides a transfer direction for data communication. When the TRX bit is set to 0, receive mode is entered and data is received from a transmit device. When the TRX bit is set to 1, transmit mode is entered, and address data and control data are output to the SDAMM, synchronized with a clock generated in the SCLMM.

The TRX bit is set to 1 automatically in the following condition:

- In slave mode, when the ALS in the S1D0 register to 0(addressing format), the AAS flag is set to 1 (address match) after the address data is received, and the received R/$\overline{W}$ bit is set to 1

The TRX bit is set to 0 in one of the following conditions:

- When an arbitration lost is detected
- When a STOP condition is detected
- When a START condition is detected
- When a START condition is disabled by the START condition duplicate protect function (1)
- When the MST bit in the S10 register is set to 0(slave mode) and a start condition is detected
- When the MST bit is set to 0 and the ACK non-return is detected
- When the ES0 bit is set to 0(I2C bus interface disabled)
- When the IHR bit in the S1D0 register is set to 1(reset)

16.5.8 Bit 7: Communication mode select bit (master/slave select bit: MST)

The MST bit selects either master mode or slave mode for data communication. When the MST bit is set to 0, slave mode is entered and the START/STOP condition generated by a master device are received. The data communication is synchronized with the clock generted by the master. When the MST bit is set to 1, master mode is entered and the START/STOP condition is generated.

Additionally, clocks required for the data communication are generated on the SCLMM.

The MST bit is set to 0 in one of the following conditions.

- After 1-byte data of a master whose arbtration is lost if arbitration lost is detected
- When a STOP condition is detected
- When a START condition is detected
- When a start condition is disabled by the START condition duplicate protect function (1)
- When the IHR bit in the S1D0 register is set to 1(reset)
- When the ES0 bit is set to 0(I2C bus interface disabled)

NOTE:

1. START condition duplicate protect function:
   When the START condition is generated, after confirming that the BB flag in the S1D0 register is set to 0 (bus free), all the MST, TRX and BB flags are set to 1 at the same time. However, if the BB flag is set to 1 immediately after the BB flag setting is confirmed because a START condition is generated by other master device, bits MST and TRX cannot be written. The duplicate protect function is valid from the rising edge of the BB flag until slave address is received. Refer to **16.9 START Condition Generation Method** for details.

16.6 I2C0 Control Register 1 (S3D0 register)

The S3D0 register controls the I2C bus interface circuit.

16.6.1 Bit 0 : Interrupt Enable Bit by STOP Condition (SIM)

The SIM bit enables the I2C bus interface interrupt request by detecting a STOP condition. If the SIM bit is set to “1”, the I2C bus interface interrupt request is generated by the STOP condition detect (no need to change in the PIN flag).

16.6.2 Bit 1: Interrupt Enable Bit at the Completion of Data Receive (WIT)

If the WIT bit is set to 1 while the ACK-CLK bit in the S20 register is set to 1 (ACK clock), the I2C bus interface interrupt request is generated and the PIN bit is set to 1 at the falling edge of the last data bit clock. Then an "L" signal is applied to the SCLMM and the ACK clock generation is controlled. **Table 16.4** and **Figure 16.12** show the interrupt generation timing and the procedure of communication restart. After the communication is restarted, the PIN bit is set to 0 again, synchronized with the falling edge of the ACK clock, and the I2C bus interface interrupt request is generated.

Table16.4 Timing of Interrupt Generation in Data Receive Mode

I2C bus Interface Interrupt Generation Timing	Procedure of Communication Restart
1) Synchronized with the falling edge of the last data bit clock	Set the ACK bit in the S20 register. Set the PIN bit to 1. (Do not write to the S00 register. The ACK clock operation may be unstable.)
2) Synchronized with the falling edge of the ACK clock	Set the S00 register

The internal WAIT flag can be read by reading the WIT bit. The internal WAIT flag is set to 1 after writing data to the S00 register and it is set to 0 after writing to the S20 register.
Consequently, the I2C bus interface interrupt request generated by the timing 1) or 2) can be determined. (See **Figure 16.12**)
When the data is transmitted and the address data is received immediately after the START condition, the WAIT flag remains 0 regardless of the WIT bit setting, and the I2C bus interface interrupt request is only generated at the falling edge of the ACK clock. Set the WIT bit to 0 when the ACK-CLK bit in the S20 register is set to 0 (no ACK clock).

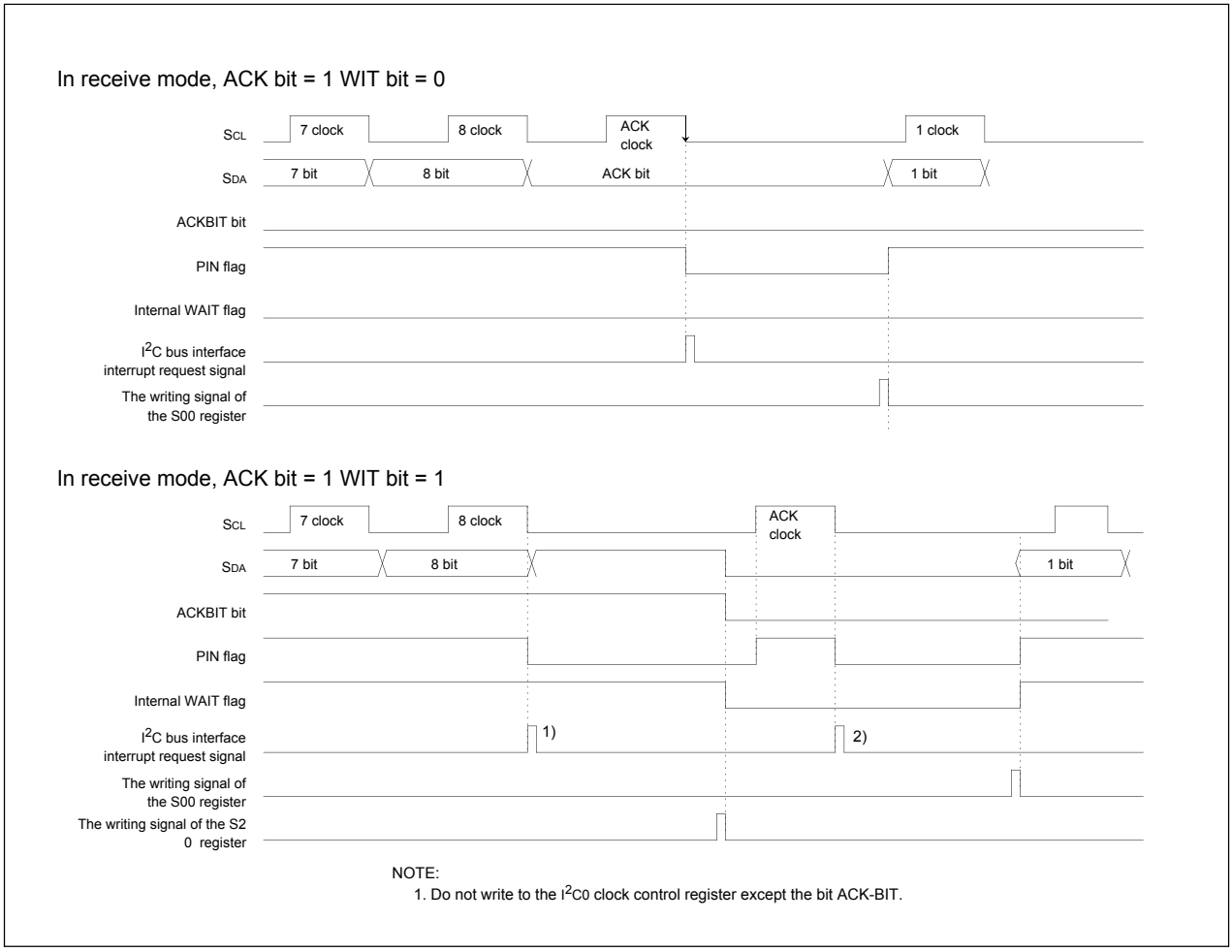


Figure 16.12 The timing of the interrupt generation at the completion of the data receive

16.6.3 Bits 2,3 : Port Function Select Bits PED, PEC

If the ES0 bit in the S1D0 register is set to 1 (I^2C bus interface enabled), the SDAMM functions as an output port. When the PED bit is set to 1 and the SCLMM functions as an output port when the PEC bit is set to 1. Then the setting values of bits P2_0 and P2_1 in the port P2 register are output to the I^2C bus, regardless of he internal SCL/SDA output signals. (SCL/SDA pins are onnected to I^2C bus interface circuit)

The bus data can be read by reading the port pi direction register in input mode, regardless of the setting values of the PED and PEC bits. **Table 16.5** shows the port specification.

Table 16.5 Port specifications

Pin Name	ES9 Bit	PED Bit	P20 Port Direction Register	Function
P20	0	-	0/1	Port I/O function
	1	0	-	SDA I/O function
	1	1	-	SDA input function, port output function
Pin Name	ES0 Bit	PEC Bit	P21 Port Direction Register	Function
P21	0	-	0/1	Port I/O function
	1	0	-	SCL I/O function
	1	1	-	SCL input function, port output funcion

16.6.4 Bits 4,5 : SDA/SCL Logic Output Value Monitor Bits SDAM/SCLM

Bits SDAM/SCLM can monitor the logic value of the SDA and SCL output signals from the I2C bus interface circuit. The SDAM bit monitors the SDA output logic value. The SCLM bit monitors the SCL output logic value. The SDAM and SCLM bits are read-only. When write, set them to 0.

16.6.5 Bits 6,7 : I2C System Clock Select Bits ICK0, ICK1

The ICK1 bit, ICK0 bit, bits ICK4 to ICK2 in the S4D0 register, and the PCLK0 bit in the PCLKR register can select the system clock (VIIC) of the I2C bus interface circuit.

The I2C bus system clock VIIC can be selected among 1/2 fIIC, 1/2.5 fIIC, 1/3 fIIC, 1/4 fIIC, 1/5 fIIC, 1/6 fIIC and 1/8 fIIC. fIIC can be selected between f1 and f2 by the PCLK0 bit setting.

Table 16.6 I2C system clock select bits

I3CK4[S4D0]	ICK3[S4D0]	ICK2[S4D0]	ICK1[S3D0]	ICK0[S3D0]	I2C system clock
0	0	0	0	0	VIIC = 1/2 fIIC
0	0	0	0	1	VIIC = 1/4 fIIC
0	0	0	1	0	VIIC = 1/8 fIIC
0	0	1	X	X	VIIC = 1/2.5 fIIC
0	1	0	X	X	VIIC = 1/3 fIIC
0	1	1	X	X	VIIC = 1/5 fIIC
1	0	0	X	X	VIIC = 1/6 fIIC

(Do not set the combination other than the above)

16.6.6 Address Receive in STOP/WAIT Mode

When WAIT mode is entered after the CM02 bit in the CM0 register is set to 0 (do not stop the peripheral function clock in wait mode), the I2C bus interface circuit can receive address data in WAIT mode. However, the I2C bus interface circuit is not operated in STOP mode or in low power consumption mode, because the I2C bus system clock VIIC is not supplied.

16.7 I2C0 Control Register 2 (S4D0 Register)

The S4D0 register controls the error communication detection.

If the SCL clock is stopped counting dring data transfer, each device is stopped, staying online. To avoid the situation, the I2C bus interface circuit has a function to detect the time-out when the SCL clock is stopped in high-level ("H") state for a specific period, and to generate an I2C bus interface interrupt request. See **Figure 16.13**.

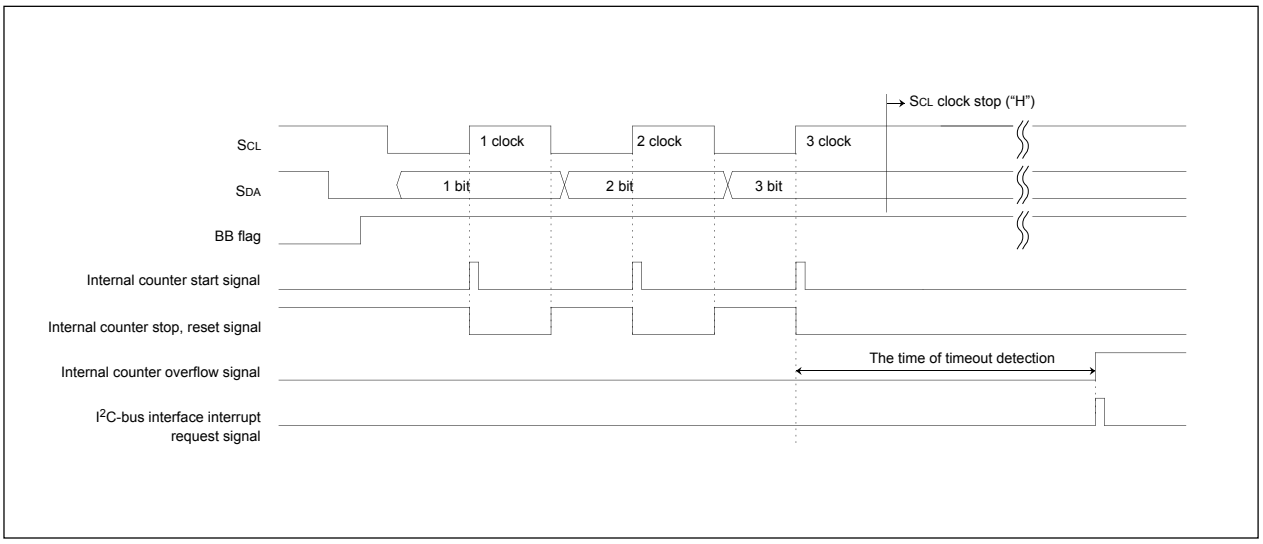


Figure 16.13 The timing of time-out detection

16.7.1 Bit0: Time-Out Detection Function Enable Bit (TOE)

The TOE bit enables the time-out detection function. When the TOE bit is set to 1, time-out is detected and the I2C bus interface interrupt request is generated when the following conditions are met.

1) the BB flag in the S10 register is set to 1 (bus busy)
2) the SCL clock stops for time-out detection period while high-level ("H") signal is maintained (see **Table 16.7**)

The internal counter measures the time-out detection time and the TOSEL bit selects between two modes, long time and short time. When time-out is detected, set the ES0 bit to 0 (I2C bus interface disabled) and reset the counter.

16.7.2 Bit1: Time-Out Detection Flag (TOF)

The TOF flag indicates the time-out detection. If the internal counter which measures the time-out period overflows, the TOF flag is set to 1 and the I2C bus interface interrupt request is generated at the same time.

16.7.3 Bit2: Time-Out Detection Period Select Bit (TOSEL)

The TOSEL bit selects time-out detection period from long time mode and short time mode. When the TOSEL bit is set to 0, long time mode is selected. When it is set to 1, short time mode is selected, respectively. The internal counter increments as a 16-bit counter in long time mode, while the counter increments as a 14-bit counter in short time mode, based on the I2C system clock (VIIC) as a counter source. **Table 16.7** shows examples of time-out detection period.

Table 16.7 Examples of Time-out Detection Period (Unit: ms)

VIIC(MHz)	Long time mode	Short time mode
4	16.4	4.1
2	32.8	8.2
1	65.6	16.4

16.7.4 Bits 3,4,5: I2C System Clock Select Bits (ICK2-4)

Bits ICK4 to ICK2, and bits ICK1 and ICK0 in the S3D0 register, and the PCLK0 bit in the PCLKR register select the system clock (VIIC) of the I2C bus interface circuit. See **Table 16.6** for the setting values.

16.7.5 Bit7: STOP Condition Detection Interrupt Request Bit (SCPIN)

The SCPIN bit monitors the stop condition detection interrupt. The SCPIN bit is set to 1 when the I2C bus interface interrupt is generated by detecting the STOP condition. When this bit is set to 0 by program, it becomes 0. However, no change occurs even if it is set to 1.

16.8 I2C0 START/STOP Condition Control Register (S2D0 Register)

The S2D0 register controls the START/STOP condition detections.

16.8.1 Bit0-Bit4: START/STOP Condition Setting Bits (SSC0-SSC4)

The SCL release time and the set-up and hold times are mesured on the base of the I2C bus system clock (VIIC). Therefore, the detection conditions changes, depending on the oscillation frequency (XIN) and the I2C bus system clock select bits. It is necessary to set bits SSC4 to SSC0 to the appropriate value to set the SCL release time, the set-up and hold times by the system clock frequency (See **Table 16.10**). Do not set odd numbers or 00000_2 to bits SSC4 to SSC0. **Table 16.2** shows the reference value to bits SSC4 to SSC0 at each oscillation frequency in standard clock mode. The detection of START/STOP conditions starts immediately after the ES0 bit in the S1D0 register is set to 1 (I2C bus interface enabled).

16.8.2 Bit5: SCL/SDA Interrupt Pin Polarity Select Bit (SIP)

The The SIP bit detect the rising edge or the falling edge of the SCLMM or SDAMM to generate SCL/SDA interrupts. The SIP bit selects the polarity of the SCLMM or the SDAMM for interrupt.

16.8.3 Bit6 : SCL/SDA Interrupt Pin Select Bit (SIS)

The SIS bit selects a pin to enable SCL/SDA interrupt.

NOTE:

1. The SCL/SDA interrupt request may be set when changing the SIP, SIS and ES0 bit settings in the S1D0 register. When using the SCL/SDA interrupt, set the above bits, while the SCL/SDA interrupt is disabled. Then, enable the SCL/SDA interrupt after setting the SCL/SDA bit in the IR register to 0.

16.8.4 Bit7: START/STOP Condition Generation Select Bit (STSPSEL)

The STSPSEL bit selects the set-up/hold times, based on the I2C system clock cycles, when the START/STOP condition is generated (See **Table 16.8**). Set the STSPSEL bit to 1 if the I2C bus system clock frequency is over 4MHz.

16.9 START Condition Generation Method

Set the MST bit, TRX bit and BB flags in the S10 register to 1 and set the PIN bit and 4 low-order bits in the S10 register to 0 simultaneously, to enter START condition standby mode, when the ES0 bit in the S1D0 register is set to 1 (I^2C bus interface enabled) and the BB flag is set to 0 (bus free). When the slave address is written to the S00 register next, START condition is generated and the bit counter is reset to 000_2 and 1-byte SCL signal is output. The START condition generation timing varies between standard clock mode and high-speed clock mode. See **Figure 16.16 and Table 16.8**.

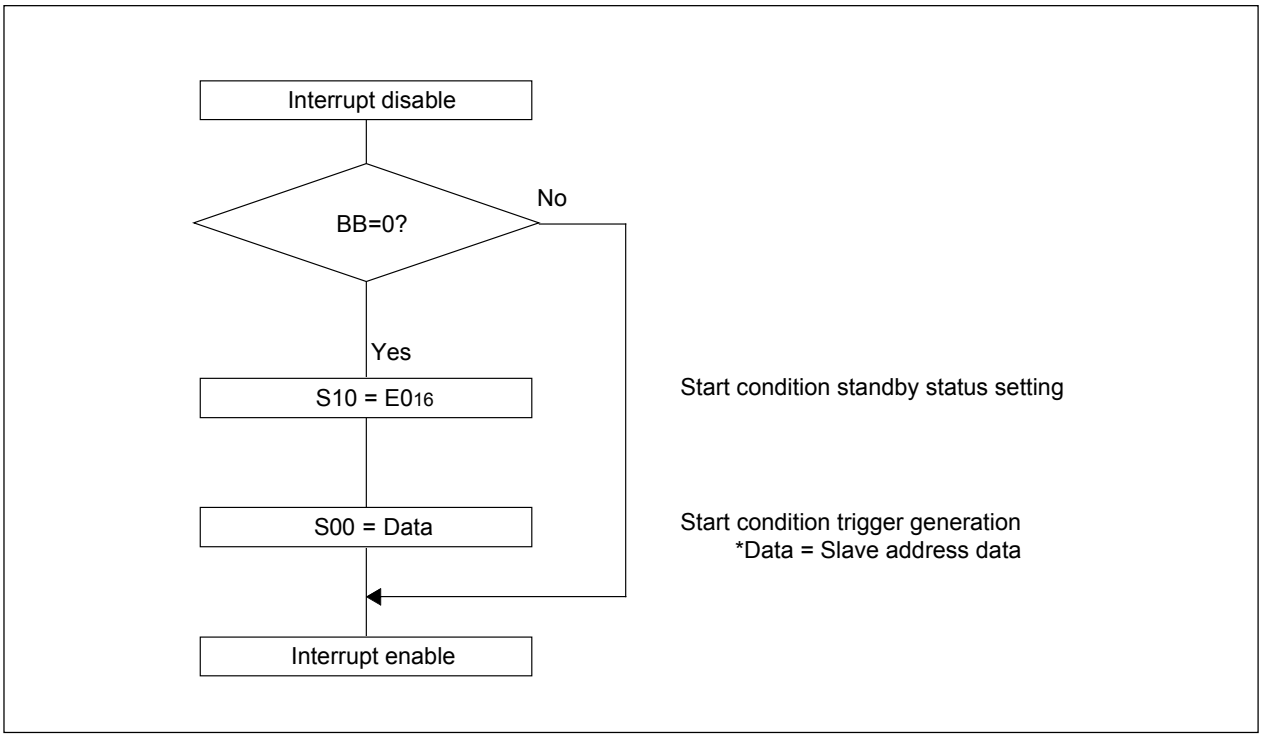


Figure 16.14 Start condition generation flow chart

16.10 START Condition Duplicate Protect Function

A START condition is generated when verifying that the BB flag in the S10 register does not use buses. However, if the BB flag is set to 1 (bus busy) by the START condition which other master device generates immediately after the BB flag is verified, the START condition is suspended by the START condition duplicate protect function. When the START condition duplicate protect function starts, it operates as follows:

- •Disable the start condition standby setting
  If the function has already been set, first exit START condition standby mode and then set bits MST and TRX in the S10 register to 0.
- •Writing to the S00 register is disabled. (The START condition trigger generation is disabled)
- •If the START condition generation is interrupted, the AL flag in the S10 register becomes 1.(arbitration lost detection)

The START condition duplicate protect function is valid between the SDA falling edge of the START condition and the receive completion of the slave address. **Figure16.15** shows the duration of the START condition duplicate protect function.

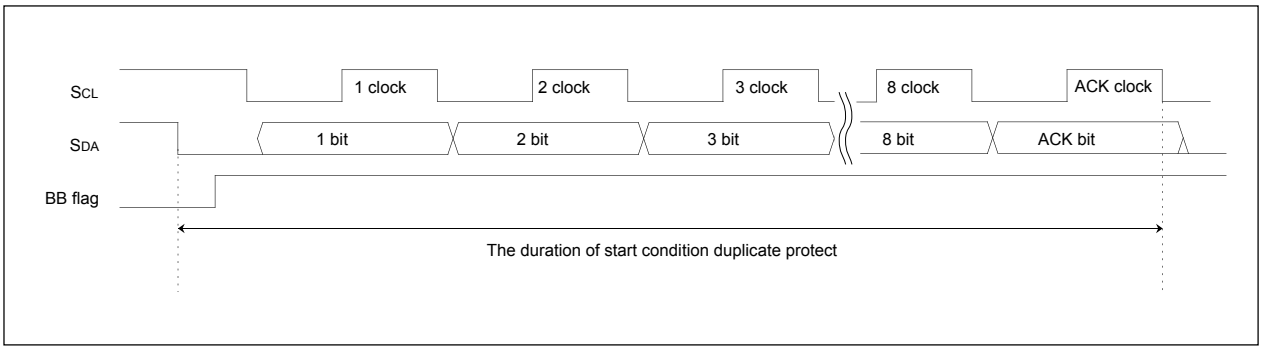


Figure 16.15 The duration of the start condition duplicate protect function

16.11 STOP Condition Generation Method

When the ES0 bit in the S1D0 register is set to 1 (I2C bus interface enabled) and bits MST and TRX in the S10 register are set to 1 at the same time, set the BB flag, PIN bit and 4 low-order bits in the S10 register to 0 simultaneously, to enter STOP condition standby mode. When dummy data is written to the S00 register next, the STOP condition is generated. The STOP condition generation timing varies between standard clock mode and high-speed clock mode. See **Figure 16.17** and **Table 16.8**.

Until the BB flag in the S10 register becomes 0 (bus free) after an instruction to generate the STOP condition is executed, do not write data to registers S10 and S00. Otherwise, the STOP condition waveform may not be generated correctly.

If an input signal level of the SCL pin is set to low ("L") after the instruction to generate the STOP condition is executed, a signal level of the SCL pin becomes high ("H"), and the BB flag is set to 0 (bus free), the MCU outputs an "L" signal to SCL pin.

In that case, the MCU can stop an "L" signal output to the SCL pin by generating the STOP condition, writing 0 to the ES0 bit in the S1D0 register (disabled), or writing 1 to the IHR bit in the S1D0 register (reset release).

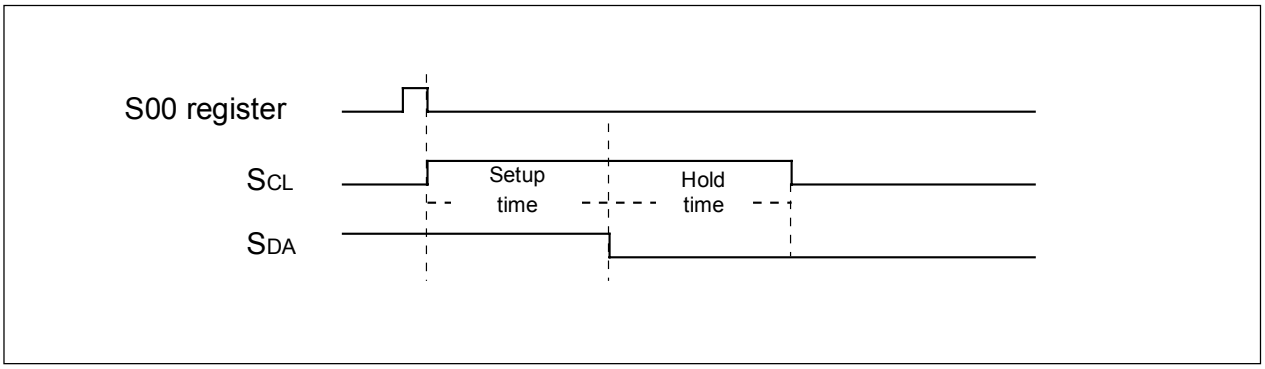


Figure 16.16 Start condition generation timing diagram

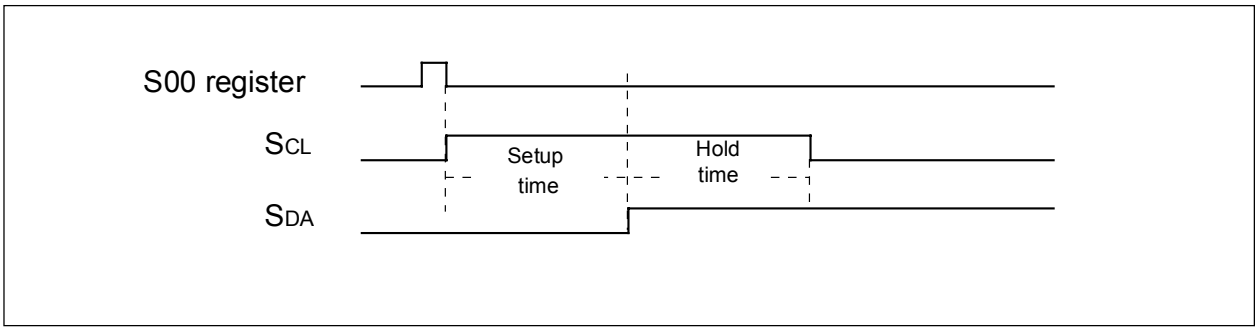


Figure 16.17 Stop condition generation timing diagram

Table 16.8 Start/Stop generation timing table

	Start/Stop Condition Generation Select Bit	Standard Clock Mode	High-speed Clock Mode
Setup time	0	5.0 µs (20 cycles)	2.5 µs (10 cycles)
	1	13.0 µs (52 cycles)	6.5 µs (26 cycles)
Hold time	0	5.0 µs (20 cycles)	2.5 µs (10 cycles)
	1	13.0 µs (52 cycles)	6.5 µs (26 cycles)

NOTE:
1. Actual time at the time of V_{IIC} = 4MHz, The contents in () denote cycle numbers.

As mentioned above, when bits MST and TRX are set to 1, START condition or STOP condition mode is entered by writing 1 or 0 to the BB flag in the S10 register and writing 0 to the PIN bit and 4 low-order bits in the S10 register at the same time. Then SDA_{MM} is left open in the START condition standby mode and SDA_{MM} is set to low-level ("L") in the STOP condition standby mode. When the S00 register is set, the START/STOP conditions are generated. In order to set bits MST and TRX to 1 without generating the START/STOP conditions, write 1 to the 4 low-order bits simultaneously. **Table 16.9** lists functions along with the S10 register settings.

Table 16.9 S10 Register Settings and Functions

S10 Register Settings								Function
MST	TRX	BB	PIN	AL	AAS	AS0	LRB	
1	1	1	0	0	0	0	0	Setting up the START condition stand by in master transmit mode
1	1	0	0	0	0	0	0	Setting up the STOP condition stand by in master transmit mode
0/1	0/1	-	0	1	1	1	1	Setting up each communication mode (refer to **16.5 I²C status register**)

16.12 START/STOP Condition Detect Operation

Figure 16.18, **Figure 16.19** and **Table 16.10** show START/STOP condition detect operations. Bits SSC4 to SSC0 in the S2D0 register set the START/STOP conditions. The START/STOP condition can be detected only when the input signal of the SCLMM and SDAMM met the following conditions: the SCL release time, the set-up time, and the hold time (see **Table 16.10**). The BB flag in the S10 register is set to 1 when the START condition is detected and it is set to 0 when the STOP condition is detected. The BB flag set and reset timing varies between standard clock mode and high-speed clock mode. See **Table 16.10**.

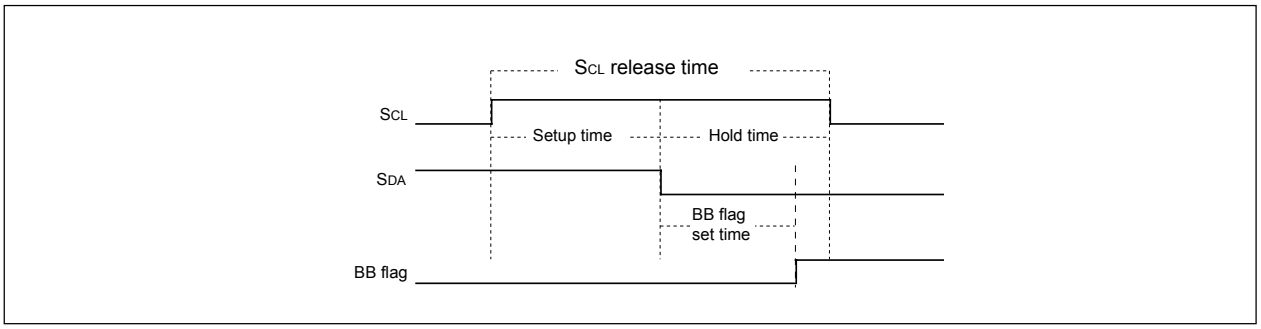


Figure 16.18 Start condition detection timing diagram

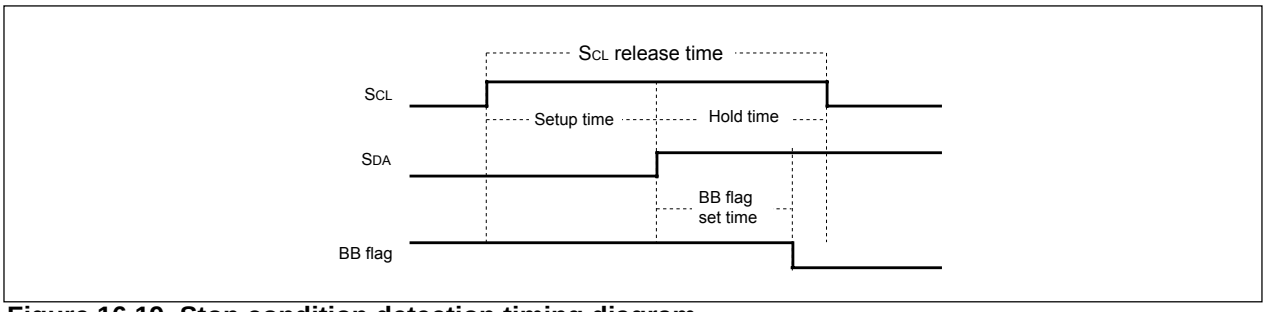


Figure 16.19 Stop condition detection timing diagram

Table 16.10 Start/Stop detection timing table

	Standard clock mode	High-speed clock mode
SCL release time	SSC value + 1 cycle (6.25µs)	4 cycles (1.0µs)
Setup time	$\frac{\text{SSC value}}{2}$ + 1 cycle < 4.0µs (3.25µs)	2 cycles (0.5µs)
Hold time	$\frac{\text{SSC value}}{2}$ cycle < 4.0µs (3.0µs)	2 cycles (0.5µs)
BB flag set/reset time	$\frac{\text{SSC value - 1}}{2}$ +2 cycles (3.375µs)	3.5 cycles (0.875µs)

NOTE:

1. Unit : number of cycle for I2C system clock VIIC
   The SSC value is the decimal notation value of bits SSC4 to SSC0. Do not set 0 or odd numbers to the SSC setting. The values in ( ) are examples when the S2D0 register is set to 18_{16} at VIIC = 4 MHz.

16.13 Address Data Communication

This section describes data transmit control when a master transferes data or a slave receives data in 7-bit address format. **Figure 16.20 (1)** shows a master transmit format.

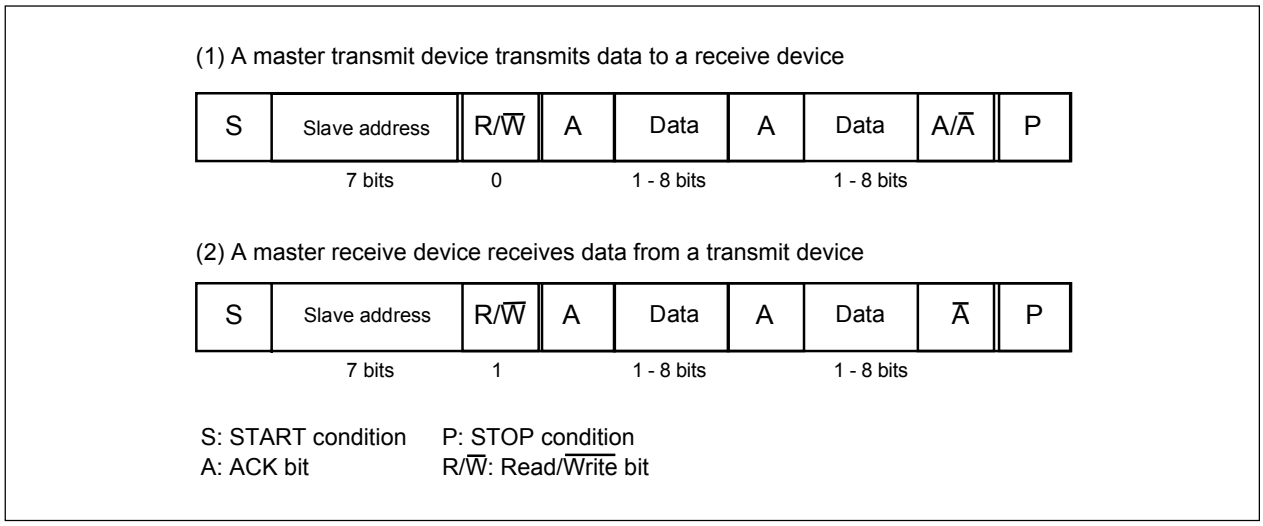


Figure 16.20 Address data communication format

16.13.1 Example of Master Transmit

For example, a master transmits data as shown below when following conditions are met: standard clock mode, SCL clock frequency of 100kHz and ACK clock added.

1) Set s slave address to the 7 high-order bits in the S0D0 register
2) Set 85_{16} to the S20 register, 000_2 to bits ICK4 to ICK2 in the S4D0 register and 00_{16} to the S3D0 registe to generate an ACK clock and set SCL clock frequency t 100 kHz (f_1=8MHz, f_{IIC}=f1)
3) Set 00_{16} to the S10 register to reset transmit/receive
4) Set 08_{16} to the S1D0 register to enable data communication
5) Confirm whether the bus is free by BB flag setting in the S10 register
6) Set $E0_{16}$ to the S10 register to enter START condition standby mode
7) Set the destination address in 7 high-order bits and 0 to a least significant bit in the S00 register to generate START condition. At this time, the first byte consisting of SCL and ACK clock are automatically generated
8) Set a transmit data to the S00 register. At this time, SCL and an ACK clock are automatically generated
9) When transmitting more than 1-byte control data, repeat the above step 8).
10) Set $C0_{16}$ in the S10 register to enter STOP condition standby mode if ACK is not returned from the slave receiver or if the transmit is completed
11) Write dummy data to the S00 regiser to generate STOP condition

16.13.2 Example of Slave Receive

For example, a slave receives data as shown below when following conditions are met: high-speed clock mode, SCL frequency of 400 kHz, ACK clock added and addressing format.

1) Set a slave address in the 7 high-order bits in the S0D0 register
2) Set A5$_{16}$ to the S20 register, 000$_2$ to bits ICK4 to ICK2 in the S4D0 register, and 00$_{16}$ to the S3D0 register to generate an ACK clock and set SCL clock frequency at 400kHz (f_1 = 8 MHz, f_{IIC} = f_1)
3) Set 00$_{16}$ to the S10 register to reset transmit/receive mode
4) Set 08$_{16}$ to the S1D0 register to enable data communication
5) When a START condition is received, addresses are compared
6) •When the transmitted addresses are all 0 (general call), the ADR0 bit in the S10 register is set to 1 and an I2C bus interface interrupt request signal is generated.
   •When the transmitted addresses match with the address set in 1), the ASS bit in the S10 register is set to 1 and an I2C bus interface interrupt request signal is generated.
   •In other cases, bits ADR0 and AAS are set to 0 and I2C bus interface interrupt request signal is not generated.
7) Write dummy data to the S00 register.
8) After receiving 1-byte data, an ACK-CLK bit is automatically returned and an I2C bus interface interrupt request signal is generated.
9) To determine whether the ACK should be returned depending on contents in the received data, set dummy data to the S00 register to receive data after setting the WIT bit in te S3D0 register to 1 (enable the I2C bus interface interrupt of data receive completion). Because the I2C bus interface interrupt is generated when the 1-byte data is received, set the ACKBIT bit to 1 or 0 to output a signal from the ACKBIT bit.
10) When receiving more than 1-byte control data, repeat steps 7) and 8) or 7) and 9).
11) When a STOP condition is detected, the communication is ended.

16.14 Precautions

(1) Access to the registers of I2C bus interface circuit

The following is precautions when read or write the control registers of I2C bus interface circuit

•S00 register

Do not rewrite the S00 register during data transfer. If the bits in the S00 register are rewritten, the bit counter for transfer is reset and data may not be transferred successfully.

•S1D0 register

Bits BC2 o BC0 are set to 0002 when START condition is detected or when 1-byte data transfer is completed. Do not read or write the S1D0 register at this timing. Otherwise, data may be read or written unsuccessfully. **Figures 16.22** and **16.23** show the bit counter reset timing.

•S20 register

Do not rewrite the S20 register except the ACKBIT bit during transfer. If the bits in the S20 register except ACKBIT bit are rewritten, the I2C bus clock circuit is reset and data may be transferred incompletely.

•S3D0 register

Rewrite bits ICK4 to ICK0 in the S3D0 register when the ES0 bit in the S1D0 register is set to 0 (I2C bus interface is disabled). When the WIT bit is read, the internal WAIT flag is read. Therefore, do not use the bit managing instruction(read-modify-write instruction) to access the S3D0 register.

•S10 register

Do not use the bit managing instruction (read-modify-write instruction) because all bits in the S10 register will be changed, depending on the communication conditions. Do not read/write when te communication mode select bits, bits MST and TRX, are changing their value. Otherwise, data may be read or written unsuccessfully. **Figures 16.21** to **16.23** show the timing when bits MST and TRX change.

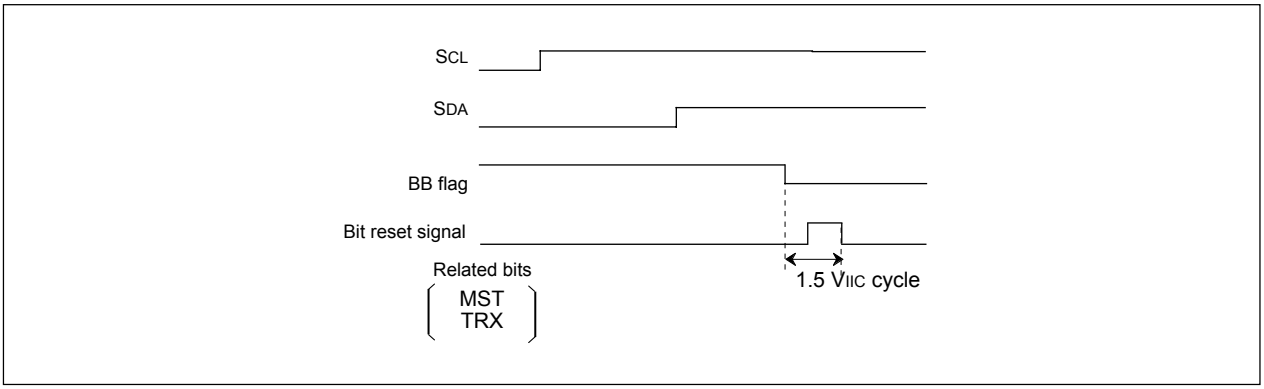


Figure 16.21 The bit reset timing (The STOP condition detection)

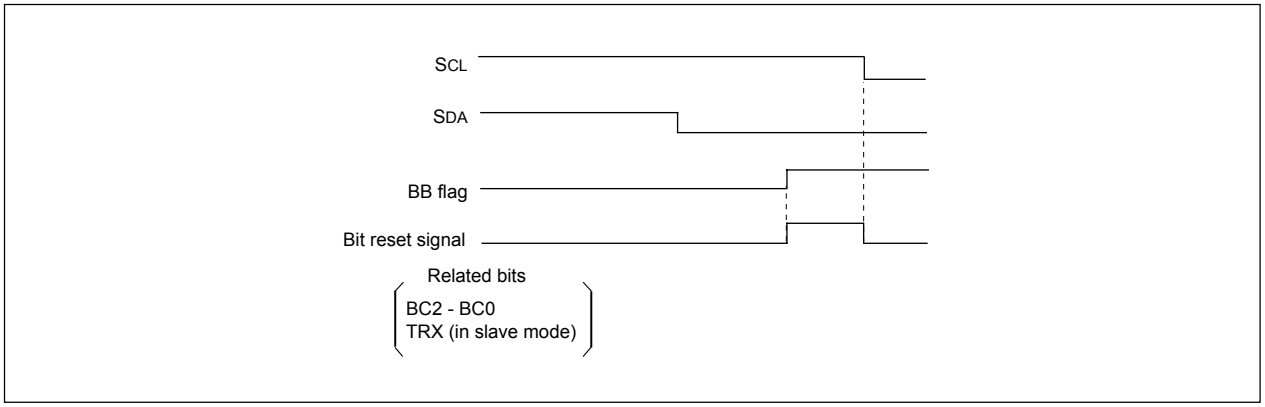


Figure 16.22 The bit reset timing (The START condition detection)

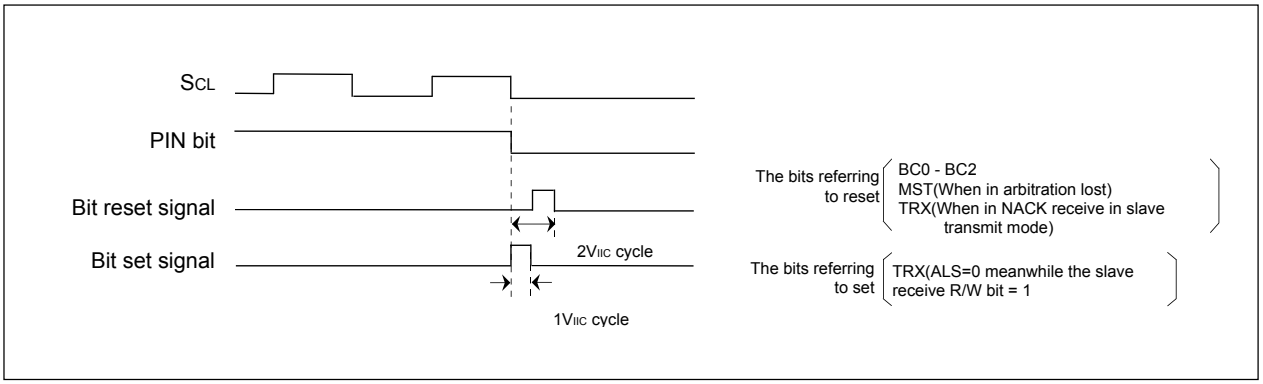


Figure 16.23 Bit set/reset timing (at the completion of data transfer)

(2) Generation of RESTART condition

In order to generate a RESTART condition after 1-byte data transfer, write E016 to the S10 register, enter START condition standby mode and leave the SDAMM open. Generate a START condition trigger by setting the S00 register after inserting a sufficient software wait until the SDAMM outputs a high-level ("H") signal. **Figure 16.24** shows the RESTART condition generation timing.

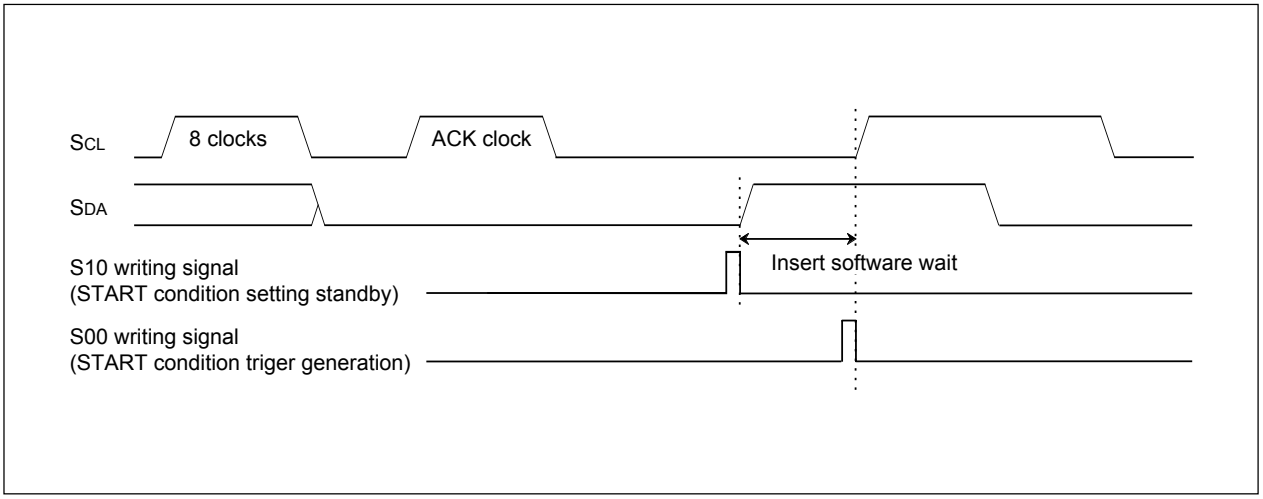


Figure 16.24 The time of generation of RESTART condition

(3) limitation of CPU clock

When the CM07 bit in the CM0 register is set to 1 (subclock), each register of the I2C bus interface circuit cannot be read or written. Read or write data when the CM07 bit is set to 0 (main clock, PLL clock, or on-chip oscillator clock).

17. CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) calculation detects an error in data blocks. The microcomputer uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) or CRC-16 ($X^{16} + X^{15} + X^2 + 1$) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of bytes. The code is updated in the CRC data register everytime one byte of data is transferred to a CRC input register. The data register must be initialized before use. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure 17.1 shows the block diagram of the CRC circuit. **Figure 17.2** shows the CRC-related registers. **Figure 17.3** shows the calculation example using the CRC_CCITT operation.

17.1 CRC Snoop

The CRC circuit includes the ability to snoop reads and writes to certain SFR addresses. This can be used to accumulate the CRC value on a stream of data without using extra bandwidth to explicitly write data into the CRCIN register. All SFR addresses after 0020_{16} are subject to the CRC snoop. The CRC snoop is useful to snoop the writes to a UART TX buffer, or the reads from a UART RX buffer.

To snoop an SFR address, the target address is written to the CRC snoop Address Register (CRCSAR). The two most significant bits of this register enable snooping on reads or writes to the target address. If the target SFR is written to by the CPU or DMA, and the CRC snoop write bit is set (CRCSW=1), the CRC will latch the data into the CRCIN register. The new CRC code will be set in the CRCD register.

Similarly, if the target SFR is read by the CRC or DMA, and the CRC snoop read bit is set (CRCSR=1), the CRC will latch the data from the target into the CRCIN register and calculate the CRC.

The CRC circuit can only calculate CRC codes on data byte at a time. Therefore, if a target SFR is accessed in word (16 bit), only one low-order byte data is stored into the CRCIN register.

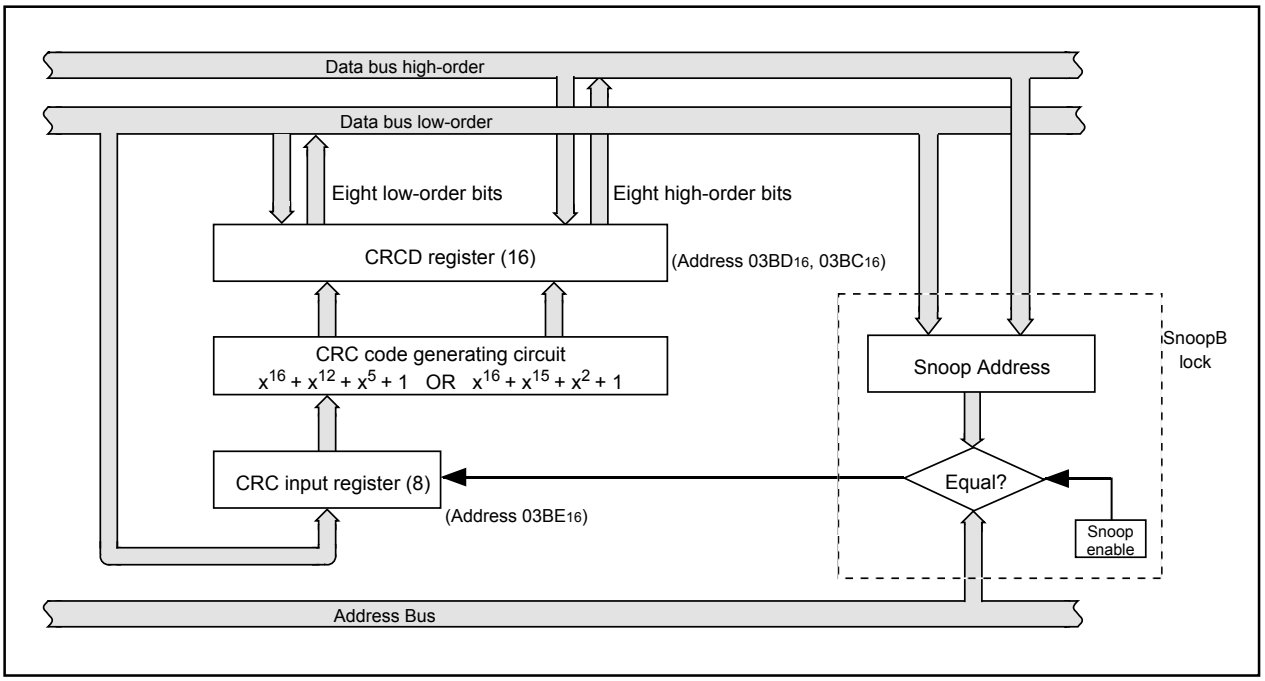
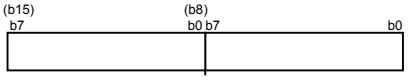


Figure 17.1 CRC circuit block diagram

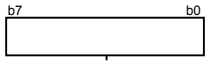
CRC Data Register



Symbol	Address	After Reset
CRCD	$03BD_{16}$ to $03BC_{16}$	Undefined

Function	Setting Range	RW
CRC calculation result output	0000_{16} to $FFFF_{16}$	RW

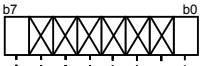
CRC Input Register



Symbol	Address	After Reset
CRCIN	$03BE_{16}$	Undefined

Function	Setting Range	RW
Data input	00_{16} to FF_{16}	RW

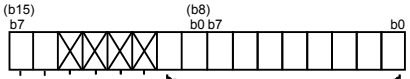
CRC Mode Register



Symbol	Address	After Reset
CRCMR	$03B6_{16}$	$0XXXXXX0_2$

Bit Symbol	Bit Name	Function	RW
CRCPS	CRC mode polynomial selection bit	0: $X^{16}+X^{12}+X^5+1$ (CRC-CCITT) 1: $X^{16}+X^{15}+X^2+1$ (CRC-16)	RW
$\overline{\text{(b6-b1)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
CRCMS	CRC mode selection bit	0: LSB first 1: MSB first	RW

SFR Snoop Address Register

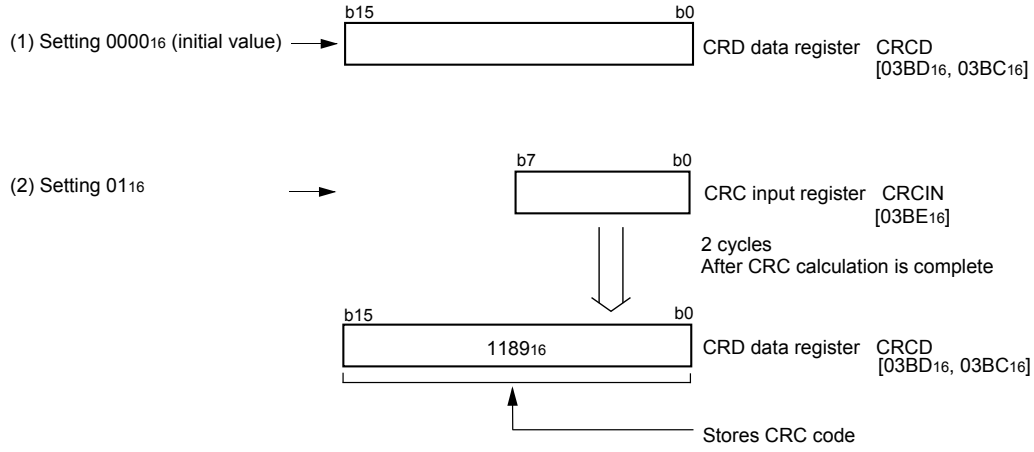


Symbol	Address	After Reset
CRCSAR	$03B5_{16}$ to $03B4_{16}$	$00XXXXXX$ $XXXXXXXX_2$

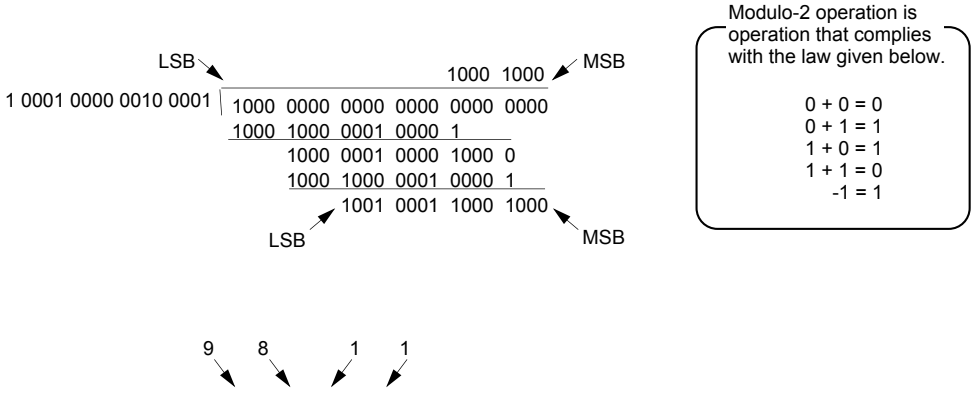
Bit Symbol	Bit Name	Function	RW
CRCSAR9-0	CRC mode polynomial selection bit	SFR address to snoop	RW
$\overline{\text{(b13-b10)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
CRCSR	CRC snoop on read enable bit	0: Disabled 1: Enabled(1)	RW
CRCSW	CRC snoop on write enable bit	0: Disabled 1: Enabled(1)	RW

NOTE:
1. Set bits CRCSR and CRCSW to 0 if the PLC07 bit in the PLC0 register is set to 1 (PLL on) and the PM20 bit in the PM2 register is set to 0 (SFR access 2 wait).

Figure 17.2 CRCD, CRCIN, CRCMR, CRCSAR Register



The code resulting from sending 01_{16} in LSB first mode is (10000 0000).This the CRC code in the generating polynomial, $(X^{16} + X^{12} + X^5 + 1)$, becomes the remainder resulting from dividing (1000 0000)X^{16} by (1 0001 0000 0010 0001) in conformity with the modulo-2 operation.



Thus the CRC code becomes (1001 0001 1000 1000). Since the operation is in LSB first mode, the (1001 0001 1000 1000) corresponds to 1189_{16} in hexadecimal notation. If the CRC operation in MSB first mode is necessary, set the CRC mode selection bit to 1. CRC data register stores CRC code for MSB first mode.

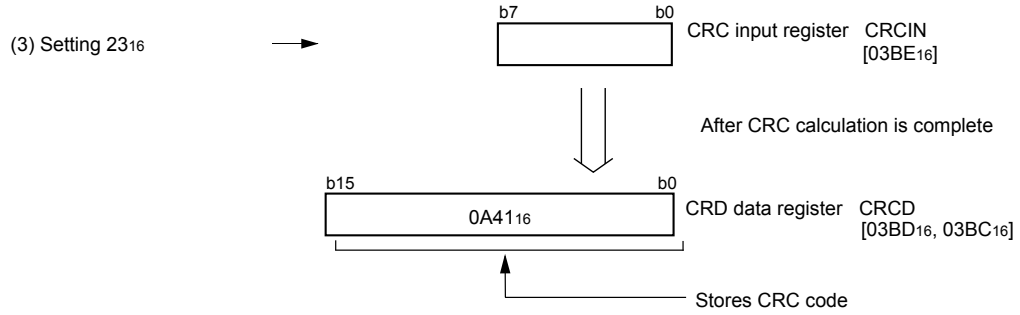


Figure 17.3 CRC Calculation

18. Programmable I/O Ports

Note

Ports P04 to P07, P10 to P14 , P34 to P37 and P95 to P97 are not available in 64-pin package.

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 71 lines P0, P1, P2, P3, P6, P7, P8, P9, P10 (except P94) for the 80-pin package, or 55 lines P00 to P03, P15 to P17, P2, P30 to P33, P6, P7, P8, P90 to P93, P10 for the 64-pin package. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high in sets of 4 lines.
Figures 18.1 to **18.4** show the I/O ports. **Figure 18.5** shows the I/O pins.
Each pin functions as an I/O port, a peripheral function input/output.
For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, set the direction bit for that pin to 0 (input mode). Any pin used as an output pin for peripheral functions is directed for output no matter how the corresponding direction bit is set.

18.1 Port Pi Direction Register (PDi Register, i = 0 to 3, 6 to 10)

Figure 18.6 shows the direction registers.
This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

18.2 Port Pi Register (Pi Register, i = 0 to 3, 6 to 10)

Figure 18.7 shows the Pi registers.
Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.
For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

18.3 Pull-up Control Register 0 to 2 (PUR0 to PUR2 Registers)

Figure 18.8 shows registers PUR0 to PUR2.
Registers PUR0 to PUR2 select whether the pins, divided into groups of four pins, are pulled up or not. The pins, selected by setting the bits in registers PUR0 to PUR2 to 1 (pull-up), are pulled up when the direction registers are set to 0 (input mode). The pins are pulled up regardless of the pins’ function.

18.4 Port Control Register (PCR Register)

Figure 18.9 shows the port control register.
When the P1 register is read after setting the PCR0 bit in the PCR register to 1, the corresponding port latch can be read no matter how the PD1 register is set.

18.5 Pin Assignment Control Register (PACR)

Figure 18.10 shows the PACR register. After reset, set bits PACR2 to PACR0 in the PACR register before a signal is input or output to each pin. When bits PACR2 to PACR0 are not set, some pins do not function as I/O ports.

Bits PACR2 to PACR0: control pins to be used
Value after reset: 000_2.
To select the 80-pin package, set the bits to 011_2.
To select the 64-pin package, set the bits to 010_2.

U1MAP bit: controls pin assignments for the UART1 function.
To assign the UART1 function to $P6_4$/CTS_1/RTS_1, $P6_5$/CLK_1, $P6_6$/RxD_1, and $P6_7$/TxD_1, set the U1MAP bit to 0 ($P6_7$ to $P6_4$).
To assign the function to $P7_0$/CTS_1/RTS_1, $P7_1$/CLK_1, $P7_2$/RxD_1, and $P7_3$/TxD_1, set the U1MAP bit to 1 ($P7_3$ to $P7_0$)

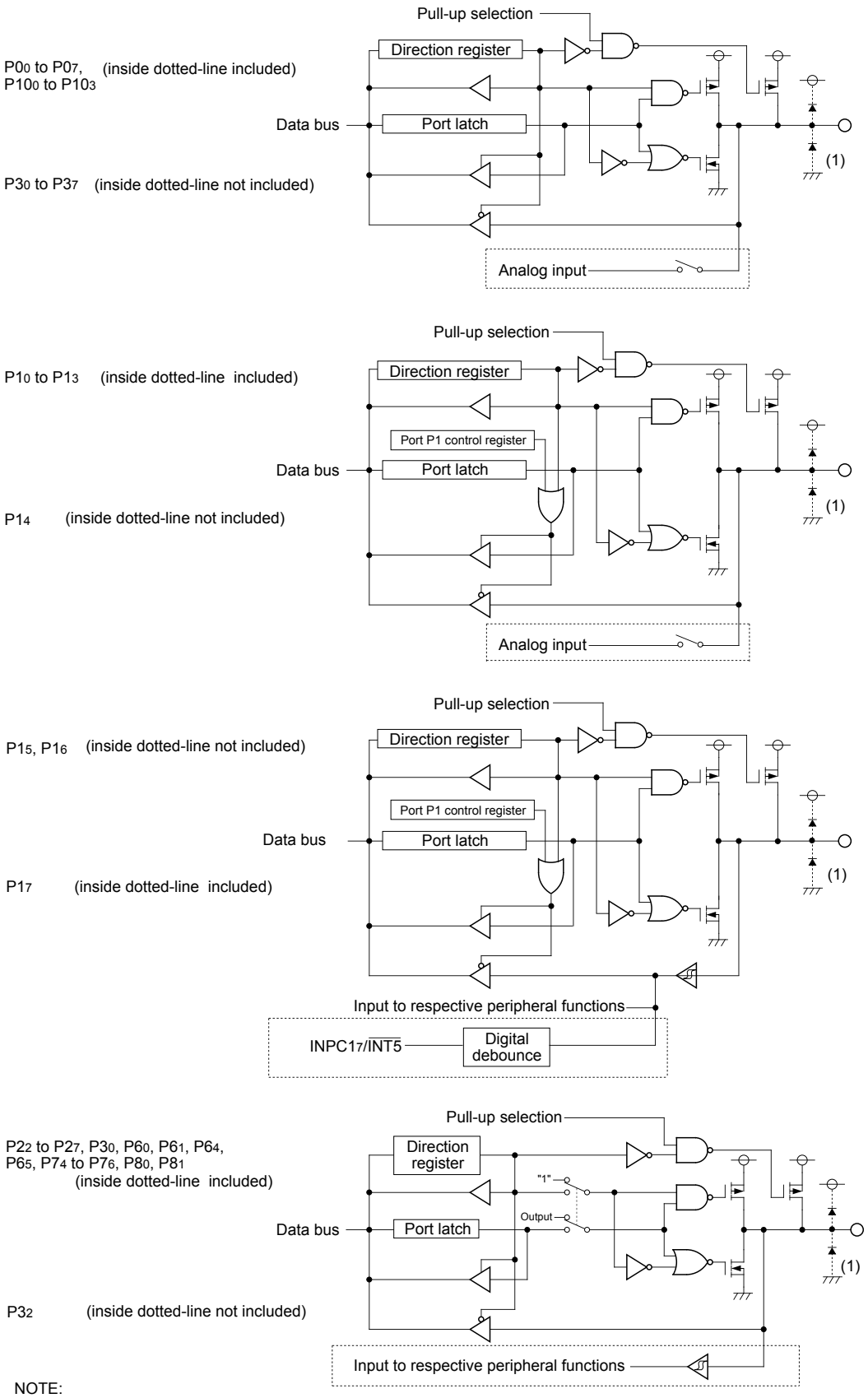
The PRC2 bit in the PRCR protects the PACR register. Set the PACR register after setting the PRC2 bit in the PRCR register.

18.6 Digital Debounce Function

Two digital debounce function circuits are provided. Level is determined when level is held, after applying either a falling edge or rising edge to the pin, longer than the programmed filter width time. This enables noise reduction.
This function is assigned to $\overline{INT5}$/INPC17 and $\overline{NMI}$/$\overline{SD}$. Digital filter width is set in the NDDR register and the P17DDR register respectively. **Figure 18.11** shows the NDDR register and the P17DDR register. Additionally, a digital debounce function is disabled to the port $P1_7$ input and the port $P8_5$ input.

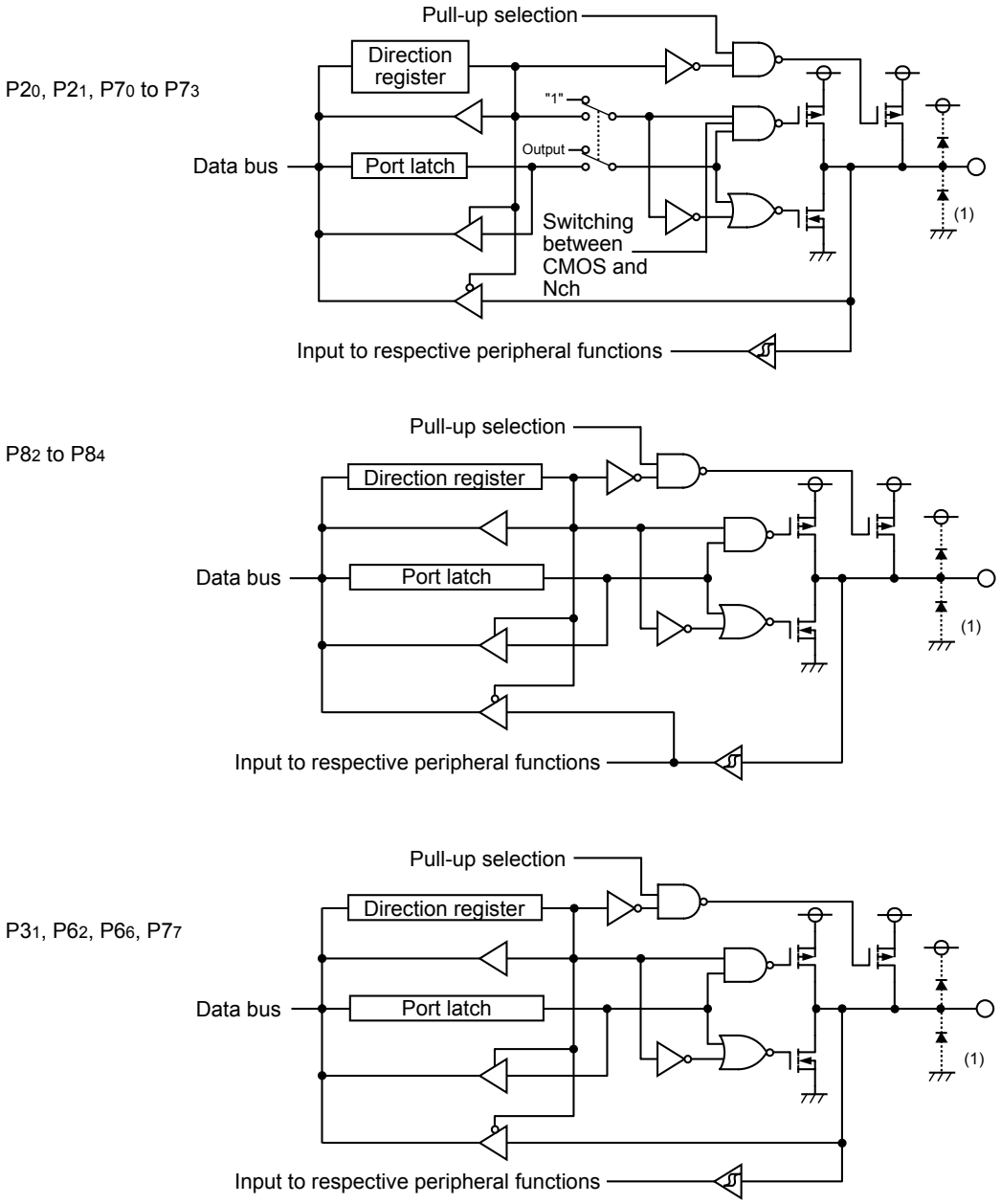
Filter width : (n+1) x 1/f8 n: count value set in the NDDR register and P17DDR register

The NDDR register and the P17DDR register decrement count value with f8 as the count source. The NDDR register and the P17DDR register indicate count time. Count value is reloaded if a falling edge or a rising edge is applied to the pin.
The NDDR register and the P17DDR register can be set 00_{16} to FF_{16} when using the digital debounce function. Setting to FF_{16} disables the digital filter. See **Figure 18.12** for details.



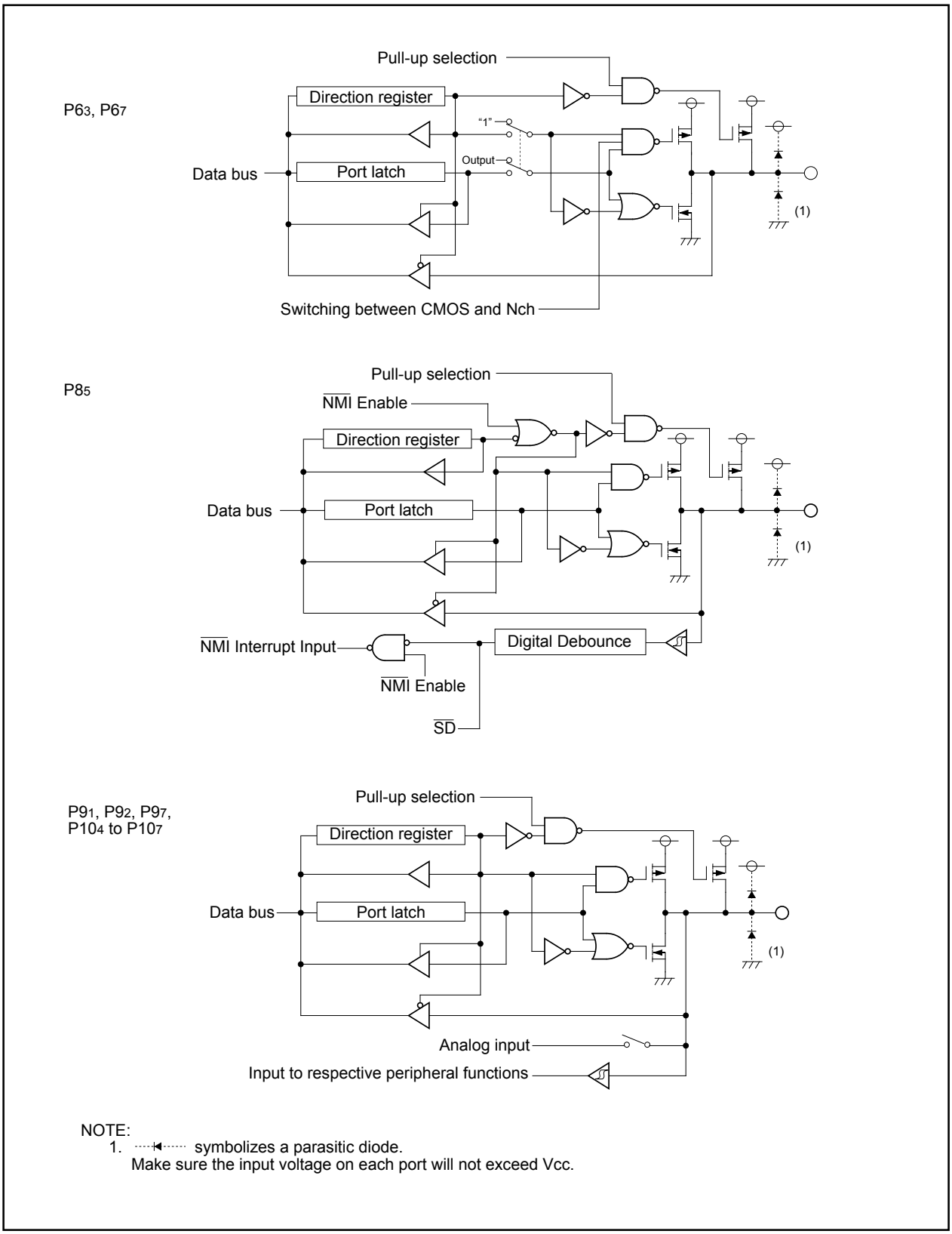
NOTE:
1. ⋯◂⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 18.1 I/O Ports (1)



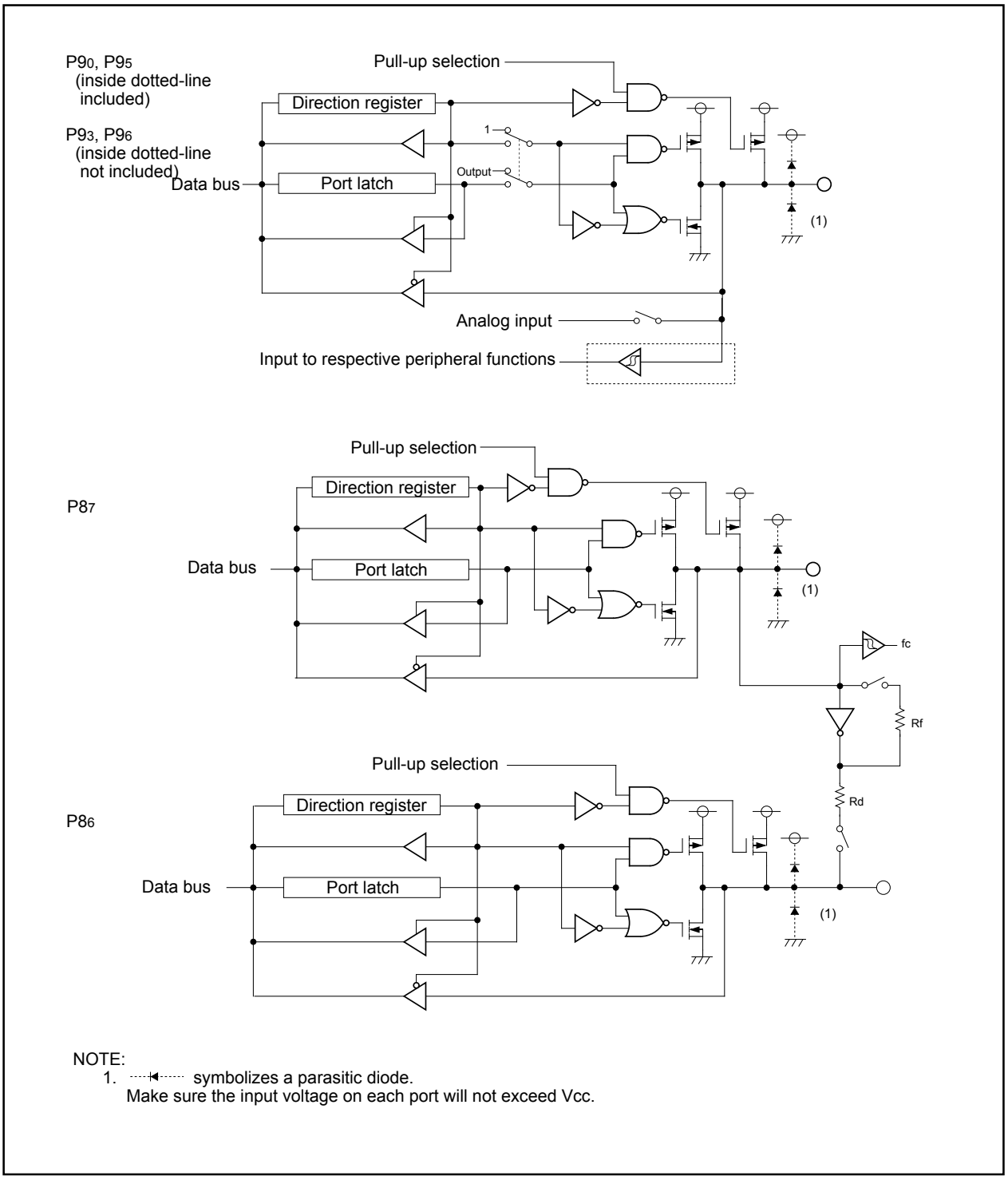
NOTE:
1. ·····◀····· symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 18.2 I/O Ports (2)



NOTE:
1. ⋯◀⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 18.3 I/O Ports (3)



NOTE:
1. ⋯◂⋯ symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 18.4 I/O Ports (4)

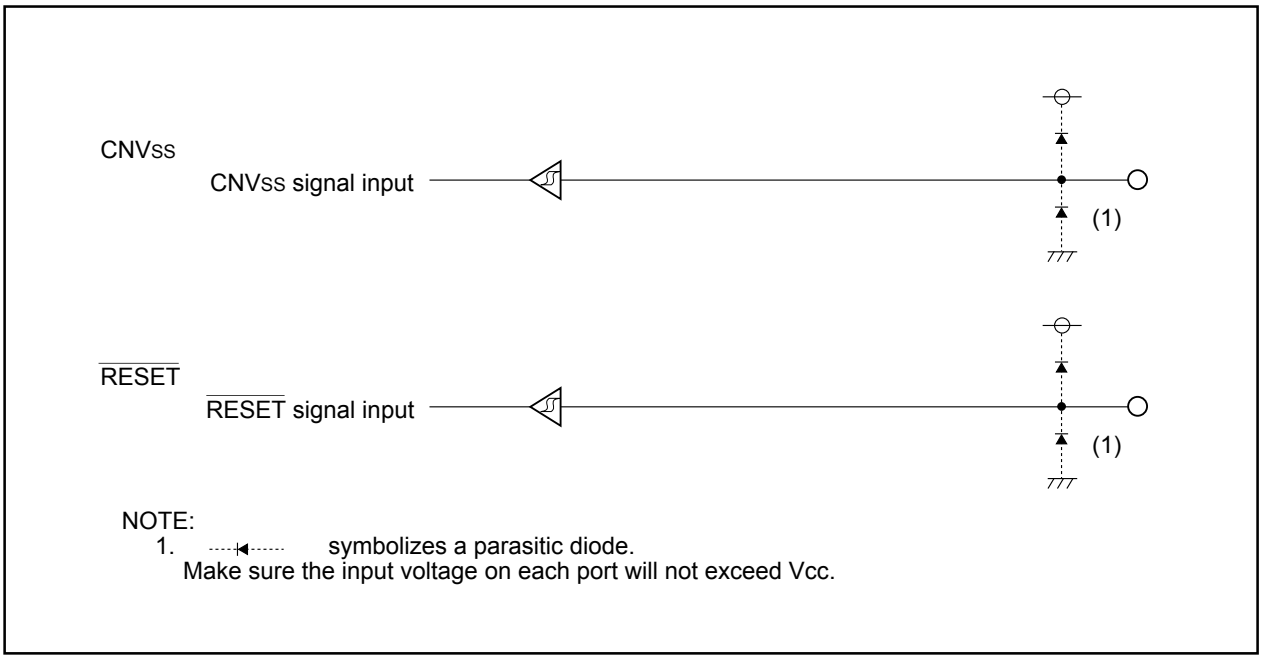
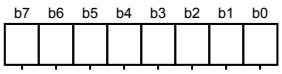


Figure 18.5 I/O Pins

Port Pi Direction Register (i=0 to 3, 6 to 8, and 10) (1)

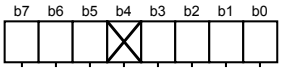


Symbol	Address	After Reset
PD0 to PD3	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$	00$_{16}$
PD6 to PD8	03EE$_{16}$, 03EF$_{16}$, 03F2$_{16}$,	00$_{16}$
PD10	03F6$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi$_0$ direction bit	0: Input mode (Functions as an input port) 1: Output mode (Functions as an output port) (i = 0 to 3, 6 to 8, and 10)	RW
PDi_1	Port Pi$_1$ direction bit		RW
PDi_2	Port Pi$_2$ direction bit		RW
PDi_3	Port Pi$_3$ direction bit		RW
PDi_4	Port Pi$_4$ direction bit		RW
PDi_5	Port Pi$_5$ direction bit		RW
PDi_6	Port Pi$_6$ direction bit		RW
PDi_7	Port Pi$_7$ direction bit		RW

NOTE:
1. Set the PACR register.
   In 80-pin package, set bits PACR2, PACR1, PACR0 to 011$_2$.
   In 64-pin package, set bits PACR2, PACR1, PACR0 to 010$_2$.

Port P9 Direction Register (1,2)



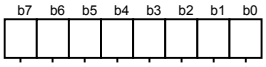
Symbol	Address	After Reset
PD9	03F3$_{16}$	000X0000$_2$

Bit Symbol	Bit Name	Function	RW
PD9_0	Port P9$_0$ direction bit	0: Input mode (Functions as an input port) 1: Output mode (Functions as an output port)	RW
PD9_1	Port P9$_1$ direction bit		RW
PD9_2	Port P9$_2$ direction bit		RW
PD9_3	Port P9$_3$ direction bit		RW
$\overline{\text{(b4)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
PD9_5	Port P9$_5$ direction bit	0: Input mode (Functions as an input port) 1: Output mode (Functions as an output port)	RW
PD9_6	Port P9$_6$ direction bit		RW
PD9_7	Port P9$_7$ direction bit		RW

NOTES:
1. Make sure the PD9 register is written to by the next instruction after setting the PRC2 bit in the PRCR register to 1(write enabled).
2. Set the PACR register.
   In 80-pin package, set bits PACR2, PACR1, PACR0 to 011$_2$.
   In 64-pin package, set bits PACR2, PACR1, PACR0 to 010$_2$.

Figure 18.6 PD0 to PD3 and PD6 to PD10 Registers

Port Pi Register (i=0 to 3, 6 to 8 and 10)(1)

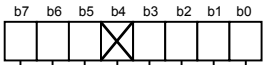


Symbol	Address	After Reset
P0 to P3	03E0₁₆, 03E1₁₆, 03E4₁₆, 03E5₁₆	Undefined
P6 to P8	03EC₁₆, 03ED₁₆, 03F0₁₆	Undefined
P10	03F4₁₆	Undefined

Bit Symbol	Bit Name	Function	RW
Pi_0	Port Pi0 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0: "L" level 1: "H" level (1) (i = 0 to 3, 6 to 8 and 10)	RW
Pi_1	Port Pi1 bit		RW
Pi_2	Port Pi2 bit		RW
Pi_3	Port Pi3 bit		RW
Pi_4	Port Pi4 bit		RW
Pi_5	Port Pi5 bit		RW
Pi_6	Port Pi6 bit		RW
Pi_7	Port Pi7 bit		RW

NOTE:
1. Set the PACR register.
   In 80-pin package, set bits PACR2, PACR1, PACR0 to 011₂.
   In 64-pin package, set bits PACR2, PACR1, PACR0 to 010₂.

Port P9 Register (1)



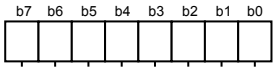
Symbol	Address	After Reset
P9	03F1₁₆	Undefined

Bit Symbol	Bit Name	Function	RW
P9_0	Port P90 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register (except for P85) 0: "L" level 1: "H" level	RW
P9_1	Port P91 bit		RW
P9_2	Port P92 bit		RW
P9_3	Port P93 bit		RW
$\overline{(b4)}$	Nothing is assigned (2)		-
P9_5	Port P95 bit		RW
P9_6	Port P96 bit		RW
P9_7	Port P97 bit		RW

NOTES:
1. Set the PACR register.
   In 80-pin package, set bits PACR2, PACR1, PACR0 to 011₂.
   In 64-pin package, set bits PACR2, PACR1, PACR0 to 010₂.
2. Nothing is assigned. If necessary, set to 0. When read, the content is 0.

Figure 18.7 P0 to P3 and P6 to P10 Registers

Pull-up Control Register 0 (1)

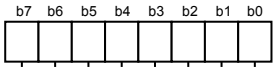


Symbol	Address	After Reset
PUR0	03FC16	0016

Bit Symbol	Bit Name	Function	RW
PU00	P00 to P03 pull-up	0: Not pulled up 1: Pulled up (1)	RW
PU01	P04 to P07 pull-up		RW
PU02	P10 to P13 pull-up		RW
PU03	P14 to P17 pull-up		RW
PU04	P20 to P23 pull-up		RW
PU05	P24 to P27 pull-up		RW
PU06	P30 to P33 pull-up		RW
PU07	P34 to P37 pull-up		RW

NOTE:
1. The pin for which this bit is 1 (pulled up) and the direction bit is 0 (input mode) is pulled up.

Pull-up Control Register 1

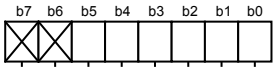


Symbol	Address	After Reset
PUR1	03FD16	0016

Bit Symbol	Bit Name	Function	RW
—— (b3-b0)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—
PU14	P60 to P63 pull-up	0: Not pulled high 1: Pulled high (1)	RW
PU15	P64 to P67 pull-up		RW
PU16	P70 to P73 pull-up		RW
PU17	P74 to P77 pull-up		RW

NOTE:
1. The pin for which this bit is 1 (pulled up) and the direction bit is 0 (input mode) is pulled up.

Pull-up Control Register 2



Symbol	Address	After Reset
PUR2	03FE16	0016

Bit Symbol	Bit Name	Function	RW
PU20	P80 to P83 pull-up	0: Not pulled up 1: Pulled up (1)	RW
PU21	P84 to P87 pull-up		RW
PU22	P90 to P93 pull-up		RW
PU23	P95 to P97 pull-up		RW
PU24	P100 to P103 pull-up		RW
PU25	P104 to P107 pull-up		RW
—— (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. The pin for which this bit is 1 (pulled up) and the direction bit is 0 (input mode) is pulled up.

Figure 18.8 PUR0 to PUR2 Registers

Port Control Register

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

Symbpl	Address	After Reset
PCR	$03FF_{16}$	00_{16}

Bit symbol	Bit Name	Function	RW
PCR0	Port P1 control bit	Operation performed when the P1 register is read 0: When the port is set for input, the input levels of $P1_0$ to $P1_7$ pins are read. When set for output, the port latch is read. 1: The port latch is read regardless of whether the port is set for input or output.	RW
—— (b7-b1)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

Figure 18.9 PCR Register

Pin Assignment Control Register (1)

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

Symbpl	Address	After Reset
PACR	$025D_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
PACR0	Pin enabling bit	010 : 64 pin 011 : 80 pin All other values are reserved. Do not use.	RW
PACR1			RW
PACR2			RW
—— (b6-b3)	Reserved bits	Nothing is assigned. If necessary, set to 0. When read, the content is 0	—
U1MAP	UART1 pin remapping bit	UART1 pins assigned to 0 : $P6_7$ to $P6_4$ 1 : $P7_3$ to $P7_0$	RW

NOTE:
1. Set the PACR register by the next instruction after setting the PRC2 bit in the PRCR register to 1 (write enable).

Figure 18.10 PACR Register

$\overline{NMI}$ Digital Debounce Register (1,2)

b7	b0	Symbol	Address	After Reset
		NDDR	033E16	FF16

Function	Setting Range	RW
If the set value =n, - n = 0 to FE16; a signal with pulse width, greater than (n+1)/f8, is input into $\overline{NMI}$ / $\overline{SD}$ - n = FF16; the digital debounce filter is disabled and all signals are input	0016 to FF16	RW

NOTES:
1. Set the PACR register by the next instruction after setting the PRC2 bit in the PRCR register to 1 (write enable).
2. When using the $\overline{NMI}$ interrupt to exit from stop mode, set the NDDR registert to FF16 before entering stop mode.

P17 Digital Debounce Register(1)

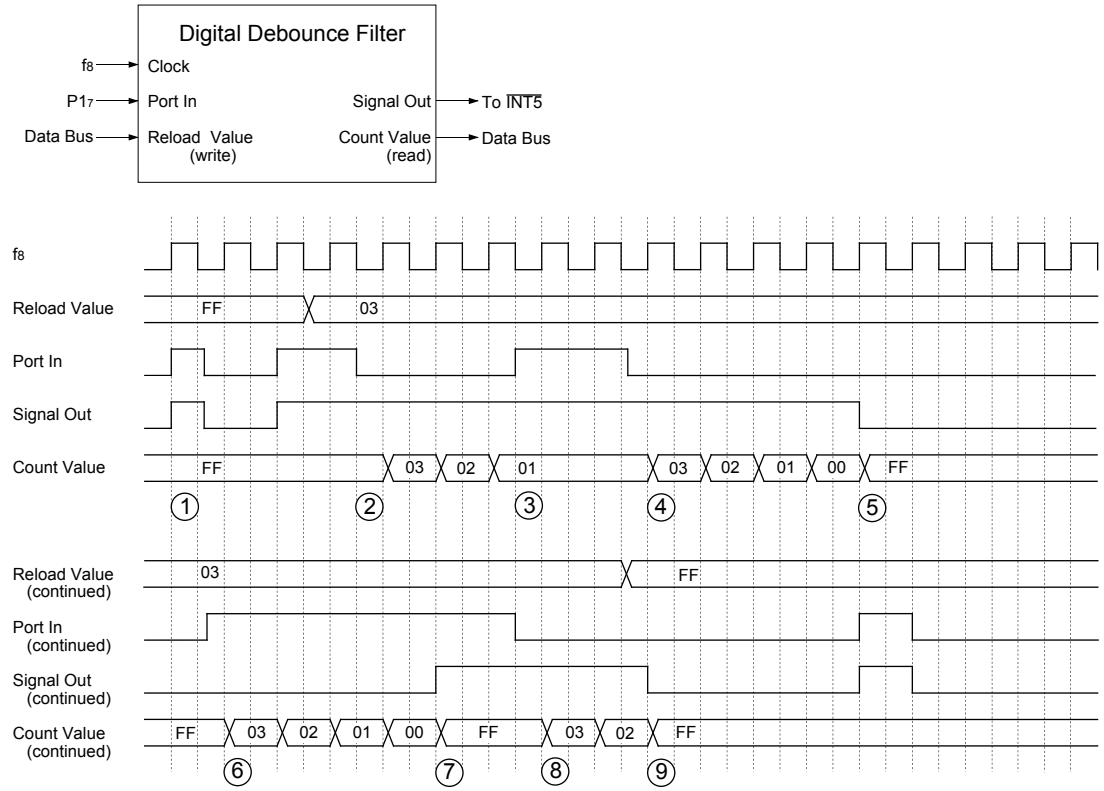
b7	b0	Symbol	Address	After Reset
		P17DDR	033F16	FF16

Function	Setting Range	RW
If the set value =n, - n = 0 to FE16; a signal with pulse width, greater than (n+1)/f8, is input into INPC17/ $\overline{INT5}$ - n = FF16; the digital debounce filter is disabled and all signals are input	0016 to FF16	RW

NOTE:
1. When using the $\overline{INT5}$ interrupt to exit from stop mode, set the P17DDR registert to FF16 before entering stop mode.

Figure 18.11 NDDR and P17DDR Registers

• Example of $\overline{INT5}$ Digital Debounce Function (if P17DDR = 03_{16})



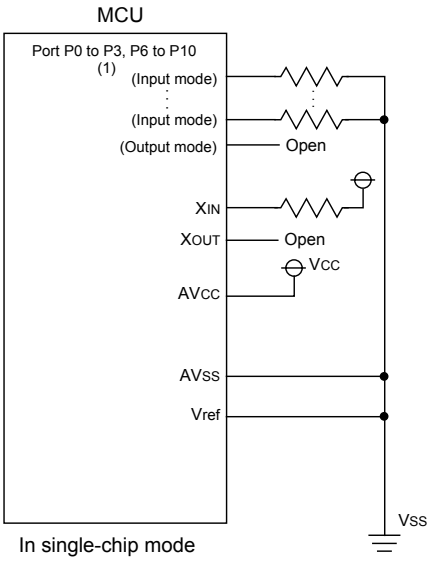
1. (Condition after reset). P17DDR=FF_{16}. Pin input signal will be output directly.
2. Set the P17DDR register to 03_{16}. The P17DDR register starts decrement along the f8 as a counter source, if the pin input level (e.g.,"L") and the signal output level (e.g.,"H") are not matched.
3. The P17DDR register will stops counting when the pin input level and the signal output level are matched (e.g., both levels are "H") while counting.
4. If the pin input level (e.g.,"L") and the signal output level (e.g.,"H") are not matched the P17DDR register will start decrement again after the setting value is reloaded.
5. When the P17DDR register is underflow, it stops counting and the signal output will output the same as pin input level (e.g."L").
6. If the pin input level (e.g.,"H") and the signal output level (e.g., "L") are not matched again, the P17DDR register will start decrement again after the setting value is reloaded.
7. When the P17DDR register is underflow, it stops counting and the signal output will output the same as pin input level (e.g."H").
8. If the pin input level (e.g.,"H") and the signal output level (e.g., "L") are not matched again, the P17DDR register will start decrement again after the setting value is reloaded.
9. Set the P17DDR register to FF_{16}. The P17DDR register starts counting after the setting value is reloaded. Pin input signal will be output directly.

Figure 18.12 Functioning of Digital Debounce Filter

Table 18.1 Unassigned Pin Handling in Single-chip Mode

Pin Name	Setting
Ports P0 to P3, P6 to P10	Enter input mode and connect each pin to VSS via a resistor (pull-down); or enter output mode and leave the pins open (1,2,4)
XOUT	Leave pin open (3)
XIN	Connect pin to VCC via a resistor (pull-up) (5)
AVCC	Connect pin to VCC
AVSS, VREF	Connect pin to VSS

NOTES:
1. If the port enters output mode and is left open, it is in input mode before output mode is entered by program after reset. While the port is in input mode, voltage level on the pins is indeterminate and power consumption may increase. Direction register setting may be changed by noise or failure caused by noise. Configure direction register settings regulary to increase the reliability of the program.
2. Use the shortest possible wiring to connect the MCU pins to unassigned pins (within 2 cm).
3. When the external clock is applied to the XIN pin, set the pin as written above.
4. In the 64-pin package, set bits PACR2, PACR1, and PACR0 in the PACR register to 010_2. In the 80-pin package, set bits PACR2, PACR1, and PACR0 to 011_2.
5. When the main clock oscillation is not used, set the CM05 bit in the CM0 register to 1 (main clock stops) to reduce power consumption.



NOTE:
1. When using the 64-pin package, set bits PACR2, PACR1, and PACR0 to 010_2.
When using the 80-pin package, set bits PACR2, PACR1, and PACR0 to 011_2.

Figure 18.13 Unassigned Pins Handling

19. Flash Memory Version

19.1 Flash Memory Performance

In the flash memory version, rewrite operation to the flash memory can be performed in three modes: CPU rewrite mode, standard serial I/O mode, and parallel I/O mode.

Table 19.1 lists specifications of the flash memory version. (Refer to **Table 1.1** or **Table 1.2** for the items not listed in **Table 19.1.**

Table 19.1 Flash Memory Version Specifications

Item		Specification
Flash memory operating mode		3 modes (CPU rewrite, standard serial I/O, parallel I/O)
Erase block		See **Figure 19.1 and 19.2** Flash Memory Block Diagram
Program method		In units of word
Erase method		Block erase
Program, erase control method		Program and erase controlled by software command
Protect method		Blocks 0 to 4 are write protected by bit FMR16. In addition, the block 0 and block 1 are write protected by bit FMR02.
Number of commands		5 commands
Program/Erase Endurance(1)	Block 0 to 4 (program area)	100 times 1,000 times (See **Table 1.5 to 1.8**)
	Block A and B (data are) (2)	100 times 10,000 times (See **Table 1.5 to 1.8**)
Data Retention		20 years (Topr = 55ºC)
ROM code protection		Parallel I/O and standard serial I/O modes are supported.

NOTES:

1. Program and erase endurance definitionProgram and erase endurance are the erase endurance of each block. If the program and erase endurance are n times (n=100,1000,10000), each block can be erased n times. For example, if a 2-Kbyte block A is erased after writing 1 word data 1024 times, each to different addresses, this is counted as one program and erasure. However, data cannot be written to the same address more than once without erasing the block. (Rewrite disabled)
2. To use the limited number of erasure efficiently, write to unused address within the block instead of rewrite. Erase block only after all possible address are used. For example, an 8-word program can be written 128 times before erase is necessary. Maintaining an equal number of erasure between Block A and B will also improve efficiency. We recommend keeping track of the number of times erasure is used.

Table 19.2 Flash Memory Rewrite Modes Overview

Flash Memory Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	Software command execution by CPU rewrites the user ROM area. EW mode 0: Rewritable in area other than flash memory EW mode 1: Rewritable in flash memory	A dedicated serial programmer rewrites the user ROM area. Standard serial I/O mode 1: Clock synchronous serial I/O Standard serial I/O mode 2: UART	A dedicated parallel programmer rewrites the user ROM area.
Areas which can be rewritten	User ROM area	User ROM area	User ROM area
Operation mode	Single chip mode	Boot mode	Parallel I/O mode
ROM programmer	None	Serial programmer	Parallel programmer

19.1.1 Boot Mode

The MCU enters boot mode when a hardware reset is performed while a high-level ("H") signal is applied to pins CNVSS and P86 or while an "H" signal is applied to pins CNVSS and P16 and a low-level ("L") signal is applied to the P85. A program in the boot ROM area is executed.

The boot ROM area is reserved. The boot ROM area stores the rewrite control program for a standard serial I/O mode before shipping. Do not rewrite the boot ROM area.

19.2 Memory Map

The flash memory contains the user ROM area and the boot ROM area (reserved area). **Figures 19.1** and **19.2** show a block diagram of the flash memory. The user ROM area has space to store the MCU operation program in single-chip mode and two 2-Kbyte spaces: the block A and B.

The user ROM area is divided into several blocks. The user ROM area can be rewritten in CPU rewrite, standard serial input/output, or parallel input/output mode.

However, to rewrite program in block 0 and 1 in CPU rewrite mode, set the FMR02 bit in the FMR0 register to 1 (block 0, 1 rewrite enabled) and the FMR16 bit in the FMR1 register to 1 (blocks 0 to 4 rewrite enabled). Also, to rewrite program in blocks 2 to 4 in CPU rewrite mode, set the FMR16 bit in the FMR1 register to 1 (blocks 0 to 4 rewrite enabled). When the PM10 bit in the PM1 register is set to 1 (data space access enabled), block A and B can be available for use.

The boot ROM area (4-byte) is a reserved area. This boot ROM area has a standard serial I/O mode control program stored before shipping. Do not rewrite the boot ROM area.

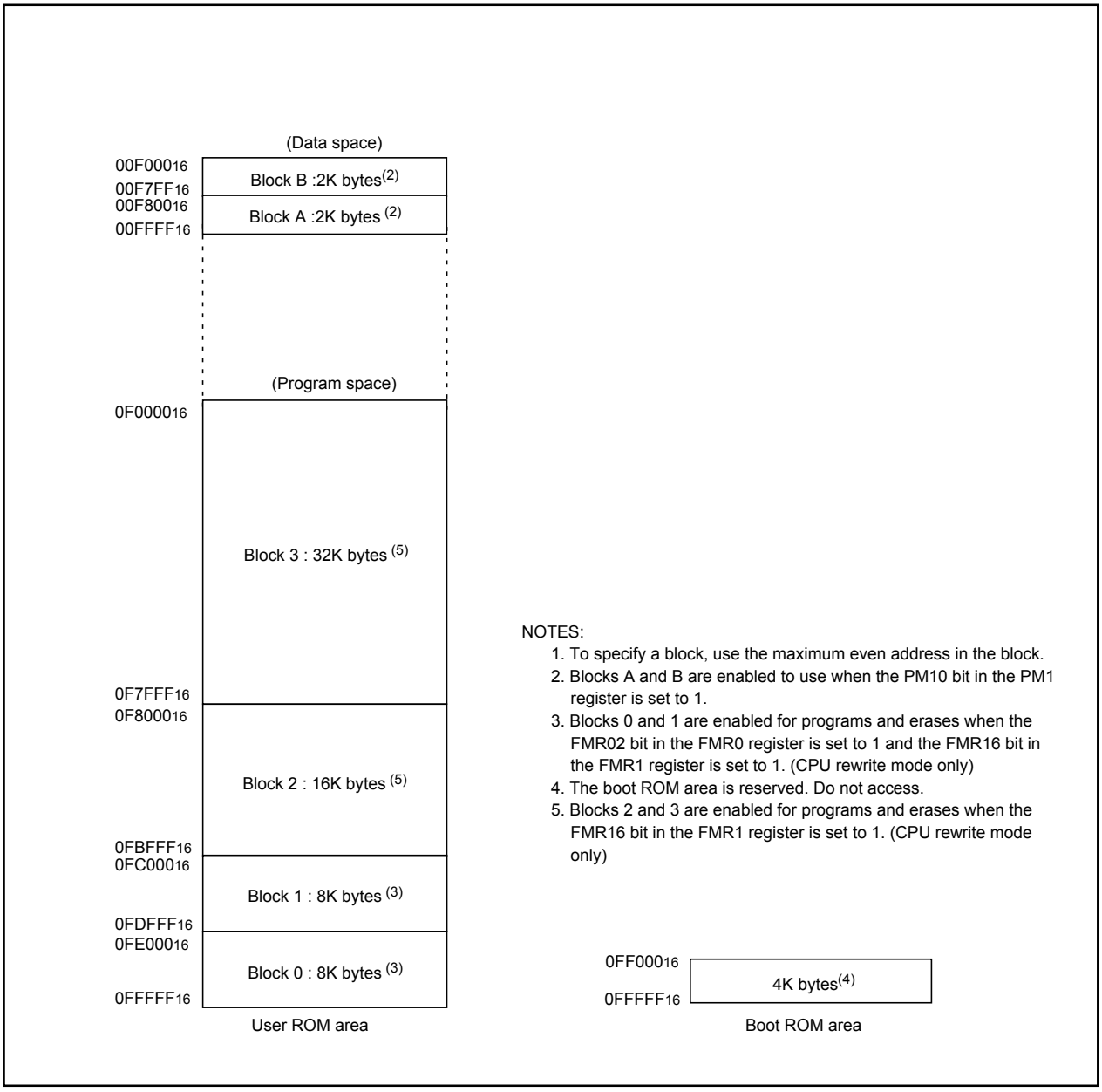


Figure 19.1 Flash Memory Block Diagram (ROM capacity 64 Kbytes)

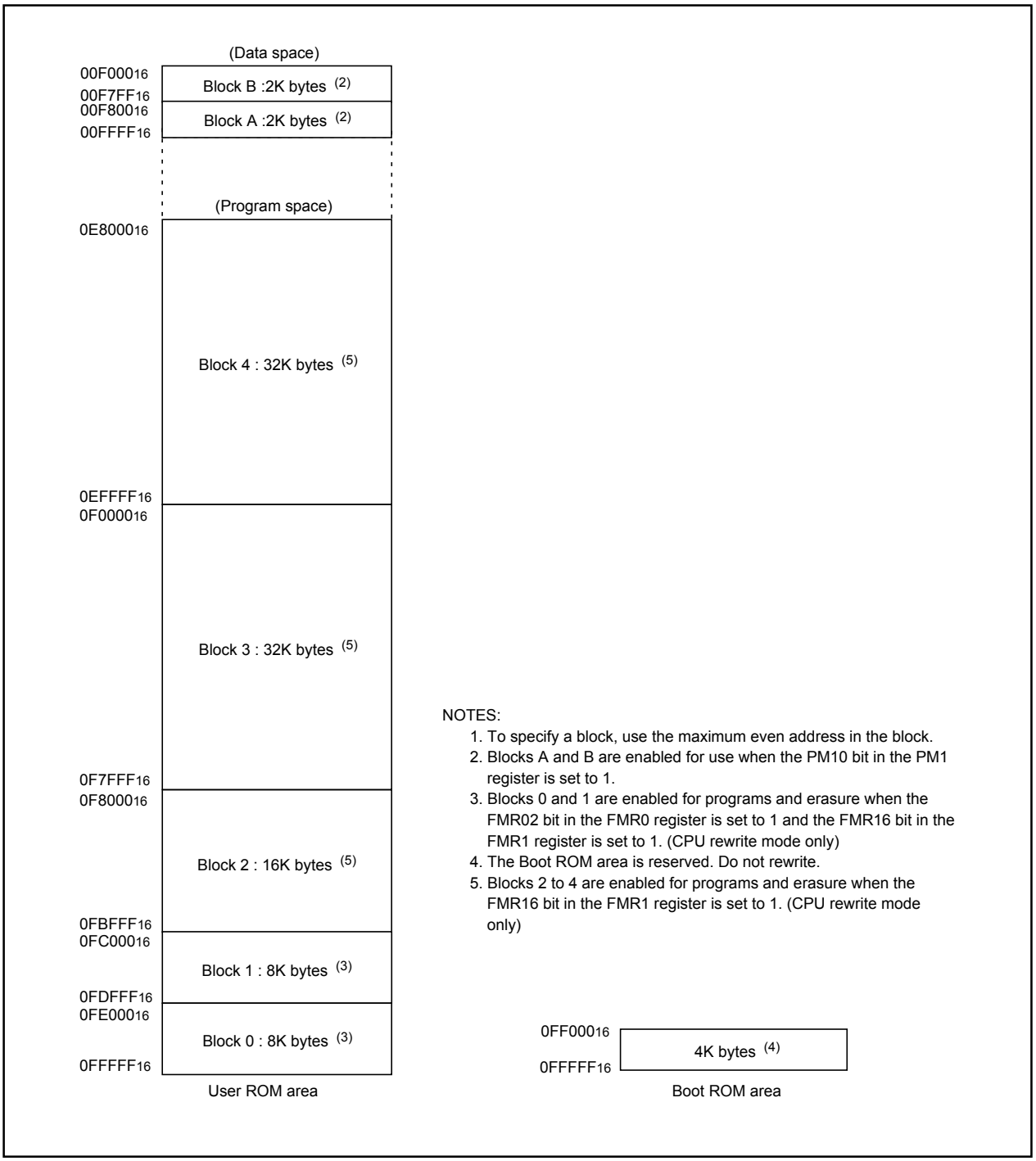


Figure 19.2 Flash Memory Block Diagram (ROM capacity 96 Kbytes)

19.3 Functions To Prevent Flash Memory from Rewriting

The flash memory has a built-in ROM code protect function for parallel I/O mode and a built-in ID code check function for standard input/output mode to prevent the flash memory from reading or rewriting.

19.3.1 ROM Code Protect Function

The ROM code protect function disables reading or changing the contents of the on-chip flash memory in parallel I/O mode. **Figure 19.3** shows the ROMCP address. The ROMCP address is located in a user ROM area. To enable ROM code protect, set the ROMCP1 bit to 00_2, 01_2, or 10_2 and set the bits 5 to 0 to 111111_2.

To cancel ROM code protect, erase the block including the the ROMCP register in CPU rewrite mode or standard serial I/O mode.

19.3.2 ID Code Check Function

Use the ID code check function in standard serial input/output mode. Unless the flash memory is blank, the ID code sent from the programmer and the 7-byte ID code written in the flash memory are compared for match. If the ID codes do not match, the commands sent from the programmer are not acknowledged. The ID code consists of 8-bit data, starting with the first byte, into addresses, $0FFFDF_{16}$, $0FFFE3_{16}$, $0FFFEB_{16}$, $0FFFEF_{16}$, $0FFFF3_{16}$, $0FFFF7_{16}$, and $0FFFFB_{16}$. The flash memory must have a program with the ID code set in these addresses.

ROM Code Protect Control Address(5)

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

Symbol: ROMCP
Address: 0FFFFF16
Factory Setting: FF16 (4)

Bit Symbol	Bit Name	Function	RW
$\overline{\text{(b5-b0)}}$	Reserved Bit	Set to 1	RW
ROMCP1	ROM Code Protect Level 1 Set Bit (1, 2, 3, 4)	b7 b6 00: 01: } Enables protect 10: 11: Disables protect	RW RW

NOTES:
1. When the ROM code protect is active by the ROMCP1 bit setting, the flash memory is protected against reading or rewriting in parallel I/O mode.
2. Set the bit 5 to bit 0 to 1111112 when the ROMCP1 bit is set to a value other than 112. When the bit 5 to bit 0 are set to values other than 1111112, the ROM code protection may not become active by setting the ROMCP1 bit to a value other than 112.
3. To make the ROM code protection inactive, erase a block including the ROMCP address in standard serial I/O mode or CPU rewrite mode.
4. The ROMCP address is set to FF16 when a block, including the ROMCP address, is erased.
5. When a value of the ROMCP address is 0016 or FF16, the ROM code protect function is disabled.

Figure 19.3 ROMCP Address

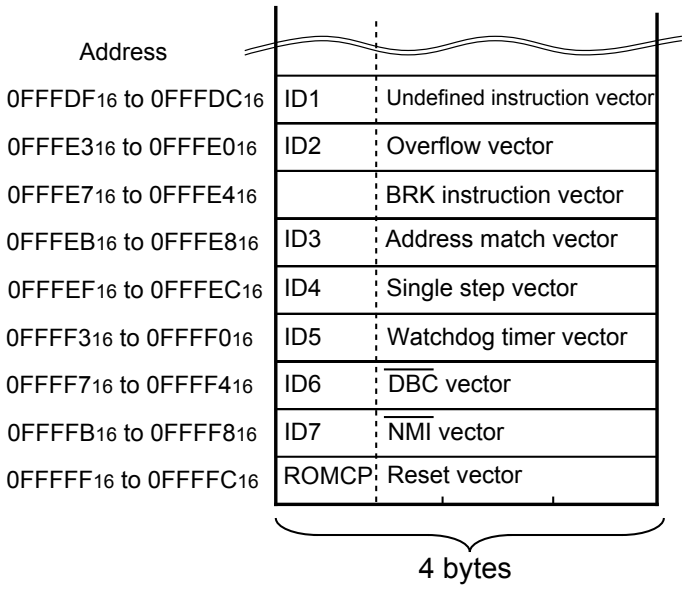


Figure 19.4 Address for ID Code Stored

19.4 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten when the CPU executes software commands. The user ROM area can be rewritten with MCU mounted on a board without using the ROM writer. The program and block erase commands are executed only in the user ROM area.

When the interrupt requests are generated during the erase operation in CPU rewirte mode, the flash memory offers an erase suspend function to suspend the erase operation and process the interrupt operation. During the erase suspend function is operated, the user ROM area can be read by program.

Erase-write(EW) 0 mode and erase-write 1 mode are provided as CPU rewrite mode. **Table 19.3** lists differences between EW mode 0 and EW mode 1. One wait is required for the CPU erase-write control.

Table 19.3 EW Mode 0 and EW Mode 1

Item	EW mode 0	EW mode 1
Operation mode	Single chip mode	Single chip mode
Areas in which a rewrite control program can be located	User ROM area	User ROM area
Areas where rewrite control program can be executed(2)	The rewrite control program must be transferred to any other than the flash memory (e.g., RAM) before being executed	The rewrite control program can be excuted in the user ROM area
Areas which can be rewritten	User ROM area	User ROM area However, this excludes blocks with the rewrite control program
Software command Restrictions	None	• Program, block erase command Cannot be executed in a block having the rewite control program • Read Status Register command Cannot be executed
Mode after programming or erasing	Read Status Register Mode	Read Array mode
CPU state during auto-write and auto-erase	Operating	In a hold state (I/O ports retain the state before the command is excuted(1)
Flash memory status detection	• Read bits FMR00, FMR06, and FMR07 in the FMR0 register by program • Execute the read status register command to read bits SR7, SR5, and SR4.	Read bits FMR00, FMR06, and FMR07 in the FMR0 registerby program
Condition for transferring to erase-suspend(3)	Set bits FMR40 and FMR41 in the FMR4 register to 1 by program.	The FMR40 bit in the FMR4 register is set to 1 and the interruput request of an acknowledged interrupt is generated

NOTES:

1. Do not generate a DMA transfer.
2. Block 1 and Block 0 are enabled for rewrite by setting FMR02 bit in the FMR0 register to 1 and setting FMR16 bit in the FMR1 register to 1. Block 2 to Block 4 are enabled for rewrite by setting FMR16 bit in the FMR1 register to 1.
3. The time, until entering erase suspend and reading flash is enabled, is maximum *td(SR-ES)* after satisfying the conditions.

19.4.1 EW Mode 0

The MCU enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to 1 (CPU rewrite mode enabled) and is ready to accept software commands. EW mode 0 is selected by setting the FMR11 bit in the FMR1 register to 0.

To set the FMR01 bit to 1, set to 1 after first writing 0. The software commands control programming and erasing. The FMR0 register or the status register indicates whether a programming or erasing operations is completed.

When entering the erase-suspend during the auto-erasing, set the FMR40 bit to 1 (erase-suspend enabled) and the FMR41 bit to 1 (suspend request). After waiting for td(SR-ES) and verifying the FMR46 bit is set to 1 (auto-erase stop), access to the user ROM area. When setting the FMR41 bit to 0 (erase restart), auto-erasing is restarted.

19.4.2 EW Mode 1

EW mode 1 is selected by setting the FMR11 bit to 1 after the FMR01 bit is set to 1 (set to 1 after first writing 0).

The FMR0 register indicates whether or not a programming or an erasing operation is completed. Read status register cannot be read in EW mode 1.

When an erase/program command is initiated, the CPU halts all program execution until the command operation is completed or erase-suspend request is generated.

When enabling an erase-suspend function, set the FMR40 bit to 1 (erase suspend enabled) and execute block erase commands. Also, the interrupt to transfer to erase-suspend must be set enabled preliminarily. When entering erase-suspend after td(SR-ES) from an interrupt is requested, interrupts can be accepted.

When an interrupt request is generated, the FMR41 bit is automatically set to 1 (suspend request) and an auto-erasing is suspended. If an auto-erasing has not completed (when the FMR00 bit is 0) after an interrupt process is completed, set the FMR41 bit to 0 (erase restart) and execute block erase commands again.

19.5 Register Description

Figure 19.5 shows the flash memory control register 0 and flash memory control register 1. **Figure 19.6** shows the flash memory control register 4.

19.5.1 Flash Memory Control Register 0 (FMR0)

•FMR 00 Bit

The FMR00 bit indicates the operating state of the flash memory. Its value is 0 while the program, erase, or erase-suspend command is being executed, otherwise, it is 1.

•FMR01 Bit

The MCU can accept commands when the FMR01 bit is set to 1 (CPU rewrite mode). To set the FMR01 bit to 1, first set it to 0 and then 1. The FMR01 bit is set to 0 only by writing 0.

•FMR02 Bit

The combined settings of bits FMR02 and FMR16 enable program and erase in the user ROM area. See **Table 19.4** for setting details. To set the FMR02 bit to 1, first set it to 0 and then 1. The FMR02 bit is valid only when the FMR01 bit is set to 1 (CPU rewrite mode enable).

•FMSTP Bit

The FMSTP bit initializes the flash memory control circuits and minimizes power consumption in the flash memory. Access to the on-chip flash memory is disabled when the FMSTP bit is set to 1. Set the FMSTP bit by program in a space other than the flash memory.

Set the FMSTP bit to 1 if one of the following occurs:

•A flash memory access error occurs during erasing or programming in EW mode 0 (FMR00 bit does not switch back to 1 (ready)).

•Low-power consumption mode or on-chip oscillator low-power consumption mode is entered.

Figure 19.9 shows a flow chart illustrating how to start and stop the flash memory before and after entering low power mode. Follow the procedure in this flow chart.

When entering stop or wait mode while the CPU rewrite mode is disabled, do not set the FMR0 register because the on-chip flash memory is automatically turned off and turned back on when exiting.

•FMR06 Bit

The FMR06 bit is a read-only bit indicating an auto-program operation state. The FMR06 bit is set to 1 when a program error occurs; otherwise, it is set to 0. For details, refer to **19.8.4 Full Status Check**.

•FMR07 Bit

The FMR07 bit is a read-only bit indicating an auto-erase operation status. The FMR07 bit is set to 1 when an erase error occurs; otherwise, it is set to 0. For details, refer to **19.8.4 Full Status Check**.

Figure 19.7 shows a EW mode 0 set/reset flowchart, **Figure 19.8** shows a EW mode 1 set/reset flow-chart.

19.5.2 Flash Memory Control Register 1 (FMR1)

•FMR11 Bit

EW mode 1 is entered by setting the FMR11 bit to 1 (EW mode 1). The FMR11 bit is valid only when the FMR01 bit is set to 1.

•FMR16 Bit

The combined setting of bits FMR02 and FMR16 enables program and erase in the user ROM area. To set the FMR16 bit to 1, first set it to 0 and then 1. The FMR16 bit is valid only when the FMR01 bit is set to 1 (CPU rewrite mode enable).

•FMR17 Bit

If the FMR17 bit is set to 1 (with wait state), 1 wait state is inserted when blocks A and B are accessed, regardless of the content of the PM17 bit in the PM1 register. The PM17 bit setting is reflected to access other blocks and internal RAM, regardless of the FMR17 bit setting.
Set the FMR17 bit to 1 (with wait state) to rewrite more than 100 times.

Table 19.4 Protection using FMR16 and FMR02

FMR16	FMR02	Block A, Block B	Block 0, Block 1	other user block
0	0	write enabled	write disabled	write disabled
0	1	write enabled	write disabled	write disabled
1	0	write enabled	write disabled	write enabled
1	1	write enabled	write enabled	write enabled

19.5.3 Flash Memory Control Register 4 (FMR4)

•FMR40 Bit

The erase-suspend function is enabled when the FMR40 bit is set to 1 (enabled).

•FMR41 Bit

When the FMR41 bit is set to 1 by program during auto-erasing in EW mode 0, erase-suspend mode is entered. In EW mode 1, the FMR41 bit is automatically set to 1 (suspend request) to enter erase-suspend mode when an enabled interrupt request is generated. Set the FMR41 bit to 0 (erase restart) to restart an auto-erasing operation.

•FMR46 Bit

The FMR46 bit is set to 0 during auto-erasing. It is set to 1 in erase-suspend mode.
Do not access to flash memory when the FMR46 bit is set to 0.

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

Symbol	Address	After Reset
FMR0	$01B7_{16}$	00000001_2

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0: Busy (during writing or erasing) 1: Ready	RO
FMR01	CPU rewrite mode select bit (1)	0: Disables CPU rewrite mode (Disables software command) 1: Enables CPU rewrite mode (Enables software commands)	RW
FMR02	Block 0, 1 rewrite enable bit (2)	Set write protection for user ROM area (see **Table 19.4**)	RW
FMSTP	Flash memory stop bit (3, 5)	0: Starts flash memory operation 1: Stops flash memory operation (Enters low-power consumption state and flash memory reset)	RW
($\overline{b5}$-$\overline{b4}$)	Reserved bit	Set to 0	RW
FMR06	Program status flag (4)	0: Successfully completed 1: Completion error	RO
FMR07	Erase status flag (4)	0: Successfully completed 1: Completion error	RO

NOTES:
1. Set the FMR01 bit to 1 immediately after setting it first to 0. Do not generate an interrupt or a DMA transfer between setting the bit to 0 and setting it to 1. Set this bit while the $P8_5$/$\overline{NMI}$/$\overline{SD}$ pin is held “H” when selecting the $\overline{NMI}$ function. Set by program in a space other than the flash memory in EW mode 0. Set this bit to read alley mode and 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1. Do not generate an interrupt or a DMA transfer between setting this bit to 0 and setting it to 1.
3. Set this bit in a space other than the flash memory by program. When this bit is set to 1, access to flash memory will be denied. To set this bit to 0 after setting it to 1, wait for 10 usec. or more after setting it to 1. To read data from flash memory after setting this bit to 0, maintain tps wait time before accessing flash memory.
4. This bit is set to 0 by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode). If the FMR01 bit is set to 0, this bit can be set to 1 by writing 1 to the FMR01 bit. However, the flash memory does not enter low-power consumption status and it is not initialized.

Flash Memory Control Register 1

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

Symbol	Address	After Reset
FMR1	$01B5_{16}$	$000XXX0X_2$

Bit Symbol	Bit Name	Function	RW
($\overline{b0}$)	Reserved bit	When read, the content is undefined	RO
FMR11	EW mode 1 select bit (1)	0: EW mode 0 1: EW mode 1	RW
($\overline{b3}$-$\overline{b2}$)	Reserved bit	When read, the content is undefined	RO
($\overline{b4}$)	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—
($\overline{b5}$)	Reserved bit	Set to 0	RW
FMR16	Block 0 to 5 rewrite enable bit (2)	Set write protection for user ROM space (see **Table 19.4**) 0: Disable 1: Enable	RW
FMR17	Block A, B access wait bit (3)	0: PM17 enabled 1: With wait state (1 wait)	RW

NOTES:
1. Set the FMR11 bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1. Do not generate an interrupt or a DMA transfer between setting the bit to 0 and setting it to 1. Set this bit while the $P8_5$/$\overline{NMI}$/$\overline{SD}$ pin is held "H" when the $\overline{NMI}$ function is selected. If the FMR01 bit is set to 0, bits FMR01 and FMR11 are both set to 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1. Do not generate an interrupt or a DMA transfer between setting this bit to 0 and setting it to 1.
3. When rewriting more than 100 times, set this bit to 1 (with wait state). When the FMR17 bit is set to1(with wait state), regardless of the PM17 bit setting, 1 wait state is inserted when accessing to blocks A and B. The PM17 bit setting is enabled, regardless of the FMR17 bit setting, as to the access to other block and the internal RAM.

Figure 19.5 FMR0 and FMR1 Registers

Flash Memory Control Register 4

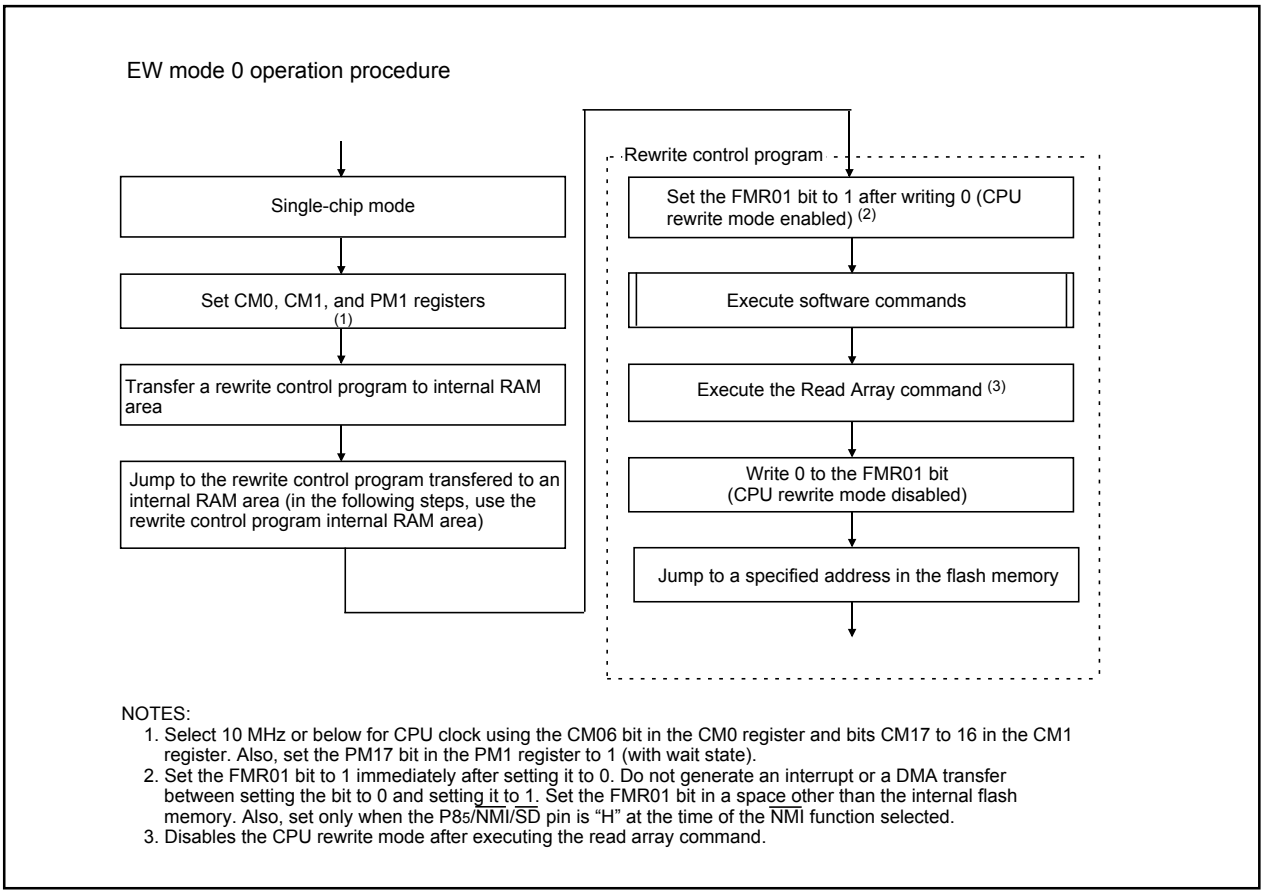
b7	b6	b5	b4	b3	b2	b1	b0
0		0	0	0	0		

Symbol: FMR4 | Address: $01B3_{16}$ | After Reset: 01000000_2

Bit Symbol	Bit Name	Function	RW
FMR40	Erase suspend function enable bit (1)	0: Disabled 1: Enabled	RW
FMR41	Erase suspend request bit (2)	0: Erase restart 1: Suspend request	RW
$\overline{\text{(b5-b2)}}$	Reserved bit	Set to 0	RO
FMR46	Erase status	0: During auto-erase operation 1: Auto-erase stop (erase suspend mode)	RO
$\overline{\text{(b7)}}$	Reserved bit	Set to 0	RW

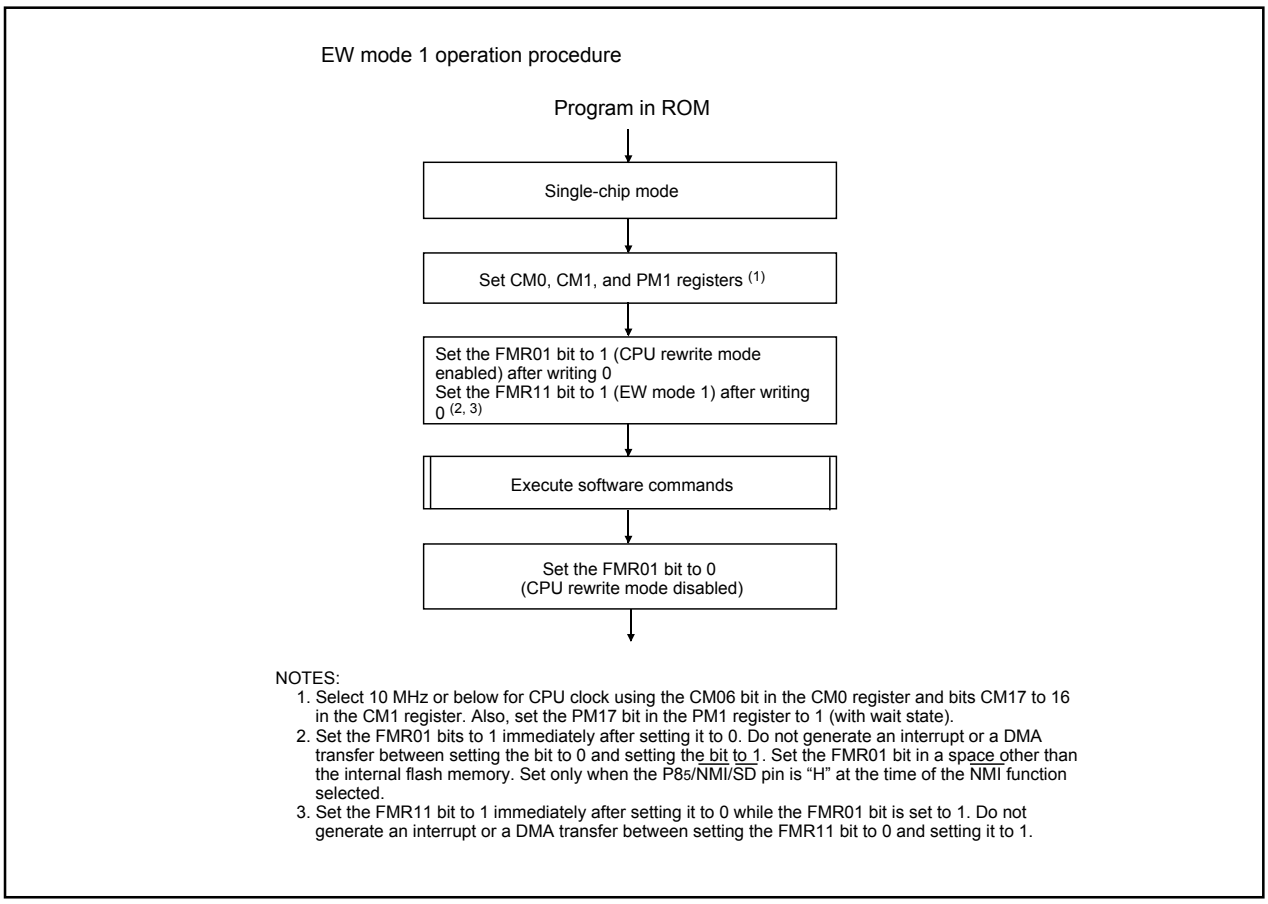
NOTES:
1. Set the FMR40 bit to 1 immediately after setting it first to 0. Do not generate an interrupt or a DMA transfer between setting the bit to 0 and setting it to 1. Set by program in a space other than the flash memory in EW mode 0.
2. The FMR41 bit is valid only when the FMR40 bit is set to 1. The FMR41 bit can be written only between executing an erase command and completing erase (this bit is set to 0 other than the above duration). The FMR41 bit can be set to 0 or 1 by program in EW mode 0. In EW mode 1, the FMR41 bit is automatically set to 1 when the FMR40 bit is 1 and a maskable interrupt is generated during erasing. The FMR41 bit cannot be set to 1 by program (it can be set to 0 by program).

Figure 19.6 FMR4 Register



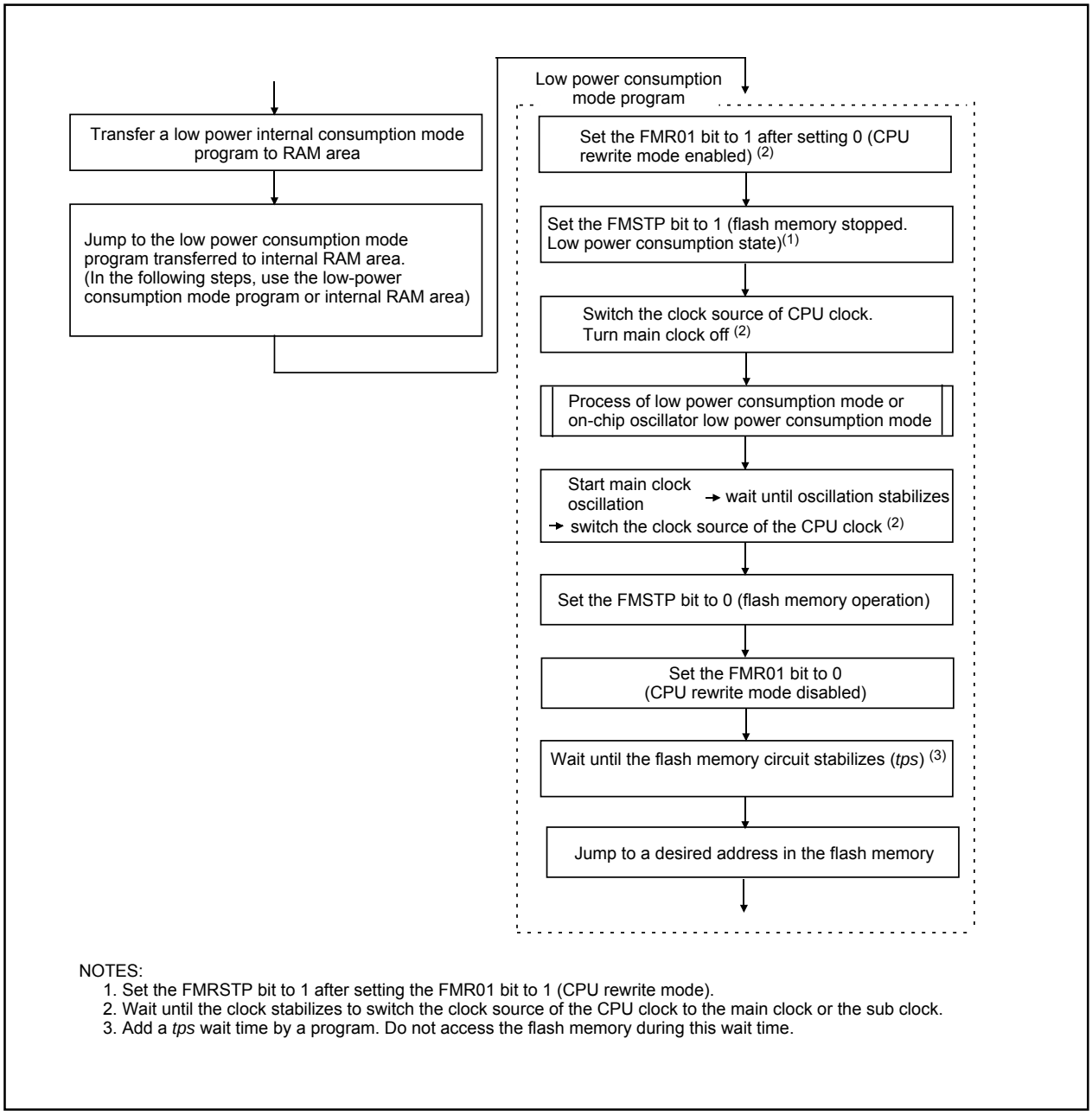
NOTES:
1. Select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and bits CM17 to 16 in the CM1 register. Also, set the PM17 bit in the PM1 register to 1 (with wait state).
2. Set the FMR01 bit to 1 immediately after setting it to 0. Do not generate an interrupt or a DMA transfer between setting the bit to 0 and setting it to 1. Set the FMR01 bit in a space other than the internal flash memory. Also, set only when the P8$_5$/$\overline{\text{NMI}}$/$\overline{\text{SD}}$ pin is "H" at the time of the $\overline{\text{NMI}}$ function selected.
3. Disables the CPU rewrite mode after executing the read array command.

Figure 19.7 Setting and Resetting of EW Mode 0



NOTES:
1. Select 10 MHz or below for CPU clock using the CM06 bit in the CM0 register and bits CM17 to 16 in the CM1 register. Also, set the PM17 bit in the PM1 register to 1 (with wait state).
2. Set the FMR01 bits to 1 immediately after setting it to 0. Do not generate an interrupt or a DMA transfer between setting the bit to 0 and setting the bit to 1. Set the FMR01 bit in a space other than the internal flash memory. Set only when the P8$_5$/$\overline{\text{NMI}}$/$\overline{\text{SD}}$ pin is "H" at the time of the $\overline{\text{NMI}}$ function selected.
3. Set the FMR11 bit to 1 immediately after setting it to 0 while the FMR01 bit is set to 1. Do not generate an interrupt or a DMA transfer between setting the FMR11 bit to 0 and setting it to 1.

Figure 19.8 Setting and Resetting of EW Mode 1



NOTES:
1. Set the FMRSTP bit to 1 after setting the FMR01 bit to 1 (CPU rewrite mode).
2. Wait until the clock stabilizes to switch the clock source of the CPU clock to the main clock or the sub clock.
3. Add a *tps* wait time by a program. Do not access the flash memory during this wait time.

Figure 19.9 Processing Before and After Low Power Dissipation Mode

19.6 Precautions in CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

19.6.1 Operation Speed

When the CPU clock source is the main clock, set the CPU clock frequency at 10 MHz or less with the CM06 bit in the CM0 register and bits CM17 and CM16 in the CM1 register, before entering CPU rewrite mode (EW mode 0 or EW mode 1). Also, when selecting f_3(ROC) of a on-chip oscillator as a CPU clock source, set bits ROCR3 and ROCR2 in the ROCR register to the CPU clock division rate at “divide-by-4” or “divide-by-8”, before entering CPU rewrite mode (EW mode 0 or EW mode 1).
In both cases, set the PM17 bit in the PM1 register to 1 (with wait state).

19.6.2 Prohibited Instructions

The following instructions cannot be used in EW mode 0 because the CPU tries to read data in the flash memory: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

19.6.3 Interrupts

EW Mode 0

- To use interrupts having vectors in a relocatable vector table, the vectors must be relocated to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available since registers FMR0 and FMR1 are forcibly reset when either interrupt occurs. However, the interrupt program, which allocates the jump addresses for each interrupt routine to the fixed vector table, is needed. Flash memory rewrite operation is aborted when the $\overline{\text{NMI}}$ or watchdog timer interrupt occurs. Set the FMR01 bit to 1 and execute the rewrite and erase program again after exiting the interrupt routine.
- The address match interrupt can not be used since the CPU tries to read data in the flash memory.

EW Mode 1

- Do not acknowledge any interrupts with vectors in the relocatable vector table or the address match interrupt during the auto program period or auto erase period with erase-suspend function disabled.

19.6.4 How to Access

To set bit FMR01, FMR02, FMR11 or FMR16 to 1, write 1 immediately after setting to 0. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to 0 and the instruction to set it to 1.
When the $\overline{\text{NMI}}$ function is selected, set the bit while an “H” signal is applied to the P8$_5$/$\overline{\text{NMI}}$/$\overline{\text{SD}}$ pin.

19.6.5 Writing in the User ROM Area

19.6.5.1 EW Mode 0

- If the supply voltage drops while rewriting the block where the rewrite control program is stored, the flash memory can not be rewritten, because the rewrite control program is not correctly rewritten. If this error occurs, rewrite the user ROM area in standard serial I/O mode or parallel I/O mode.

19.6.5.2 EW Mode 1

- Do not rewrite the block where the rewrite control program is stored.

19.6.6 DMA Transfer

In EW mode 1, do not generate a DMA transfer while the FMR00 bit in the FMR0 register is set to 0. (during the auto-programming or auto-erasing).

19.6.7 Writing Command and Data

Write the command codes and data to even addresses in the user ROM area.

19.6.8 Wait Mode

When entering wait mode, set the FMR01 bit to 0 (CPU rewrite mode disabled) before executing the WAIT instruction.

19.6.9 Stop Mode

When entering stop mode, the following settings are required:

- Set the FMR01 bit to 0 (CPU rewrite mode disabled) and disable the DMA transfer before setting the CM10 bit to 1 (stop mode).

19.6.10 Low Power Consumption Mode and On-Chip Oscillator-Low Power Consumption Mode

If the CM05 bit is set to 1 (main clock stopped), do not execute the following commands.

- Program
- Block erase

19.7 Software Commands

Read or write 16-bit commands and data from or to even addresses in the user ROM area. When writing a command code, 8 high-order bits (D_{15}–D_8) are ignored.

Table 19.5 Software Commands

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_{15} to D_0)	Mode	Address	Data (D_{15} to D_0)
Read array	Write	X	xxFF16			
Read status register	Write	X	xx7016	Read	X	SRD
Clear status register	Write	X	xx5016			
Program	Write	WA	xx4016	Write	WA	WD
Block erase	Write	X	xx2016	Write	BA	xxD016

SRD: Status register data (D_7 to D_0)
WA : Write address (However,even address)
WD : Write data (16 bits)
BA : Highest-order block address (However,even address)
X : Any even address in the user ROM area
xx : 8 high-order bits of command code (ignored)

19.7.1 Read Array Command (FF16)

The read array command reads the flash memory.

Read array mode is entered by writing command code xxFF16 in the first bus cycle. Content of a specified address can be read in 16-bit unit after the next bus cycle. The MCU remains in read array mode until an another command is written. Therefore, contents of multiple addresses can be read consecutively.

19.7.2 Read Status Register Command (7016)

The read status register command reads the status register.

By writing command code xx7016 in the first bus cycle, the status register can be read in the second bus cycle (Refer to **19.8 Status Register**). Read an even address in the user ROM area. Do not execute this command in EW mode 1.

19.7.3 Clear Status Register Command (5016)

The clear status register command clears the status register to 0.

By writing xx5016 in the first bus cycle, and bits FMR06 to FMR07 in the FMR0 register and bits SR4 to SR5 in the status register are set to 0.

19.7.4 Program Command (40$_{16}$)

The program command writes 2-byte data to the flash memory.

Auto program operation (data program and verify) start by writing xx40$_{16}$ in the first bus cycle and data to the write address specified in the second bus cycle. The address value specified in the first bus cycle must be the same even address as the write address secified in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether an auto-programming operation has been completed. The FMR00 bit is set to 0 during the auto-program and 1 when the auto-program operation is completed.

After the completion of auto-program operation, the FMR06 bit in the FMR0 register indicates whether or not the auto-program operation has been successfully completed. (Refer to **19.8.4 Full Status Check**).

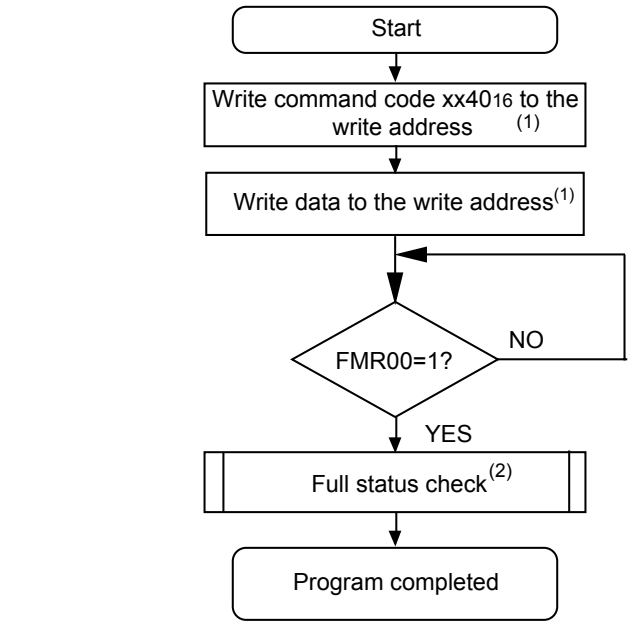
Also, each block can disable programming command (Refer to **Table 19.4**).

An address that is already written cannot be altered or rewritten.

When commands other than the program command are executed immediately after executing the program command, set the same address as the write address specified in the second bus cycle of the program command, to the specified address value in the first bus cycle of the following command.

In EW mode 1, do not execute this command on the blocks where the rewrite control program is allocated.

In EW mode 0, the MCU enters read status register mode as soon as the auto-program operation starts and the status register can be read. The SR7 bit in the status register is set to 0 as soon as the auto-program operation starts. This bit is set to 1 when the auto-program operation is completed. The MCU remains in read status register mode until the read array command is written. After completion of the auto-program operation, the status register indicates whether or not the auto-program operation has been successfully completed.



NOTES:
1. Write the command code and data at even address.
2. Refer to **Figure 19.13**.

Figure 19.10 Flow Chart of Program Command

19.7.5 Block Erase

Auto erase operation (erase and verify) start in the specified block by writing xx20$_{16}$ in the first bus cycle and xxD0$_{16}$ to the highest-order even addresse of a block in the second bus cycle.

The FMR00 bit in the FMR0 register indicates whether the auto-erase operation has been completed. The FMR00 bit is set to 0 (busy) during the auto-erase and 1 (ready) when the auto-erase operation is completed.

When using the erase-suspend function in EW mode 0, verify whether a flash memory has entered erase suspend mode, by the FMR46 bit in the FMR4 register. The FMR46 bit is set to 0 during auto-erase operation and 1 when the auto-erase operation is completed (entering erase-suspend).

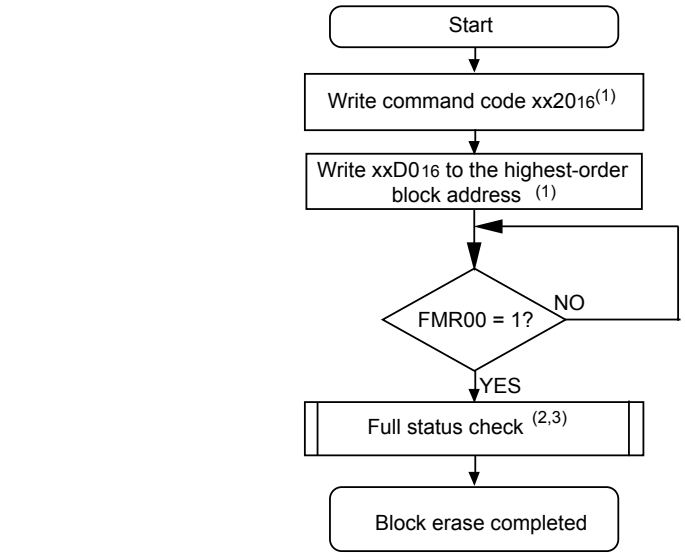
After the completion of an auto-erase operation, the FMR07 bit in the FMR0 register indicates whether or not the auto-erase operation has been successfully completed. (Refer to **19.8.4 Full Status Check**). Also, each block can disable erasing. (Refer to **Table 19.4**).

Figure 19.11 shows a flow chart of the block erase command programming when not using the erase-suspend function. **Figure 19.12** shows a flow chart of the block erase command programming when using an erase-suspend function.

In EW mode 1, do not execute this command on the block where the rewrite control program is allocated.

In EW mode 0, the MCU enters read status register mode as soon as the auto-erase operation starts and the status register can be read. The SR7 bit in the status register is set to 0 at the same time the auto-erase operation starts. This bit is set to 1 when the auto-erase operation is completed. The MCU remains in read status register mode until the read array command is written.

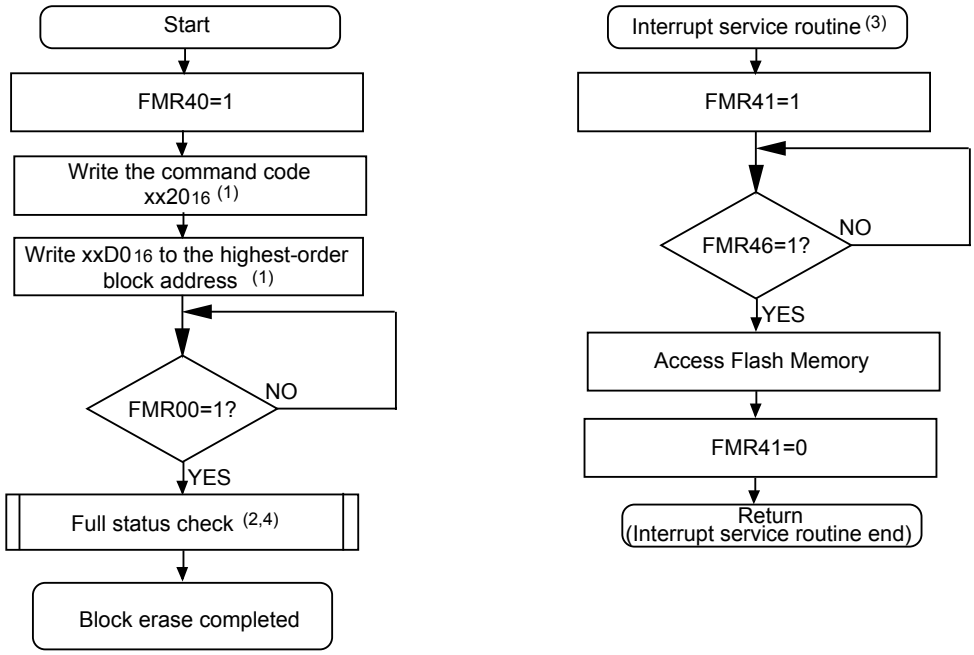
When the erase error occurs, execute the clear status register command and block erase command at leaset three times until an erase error does not occur.



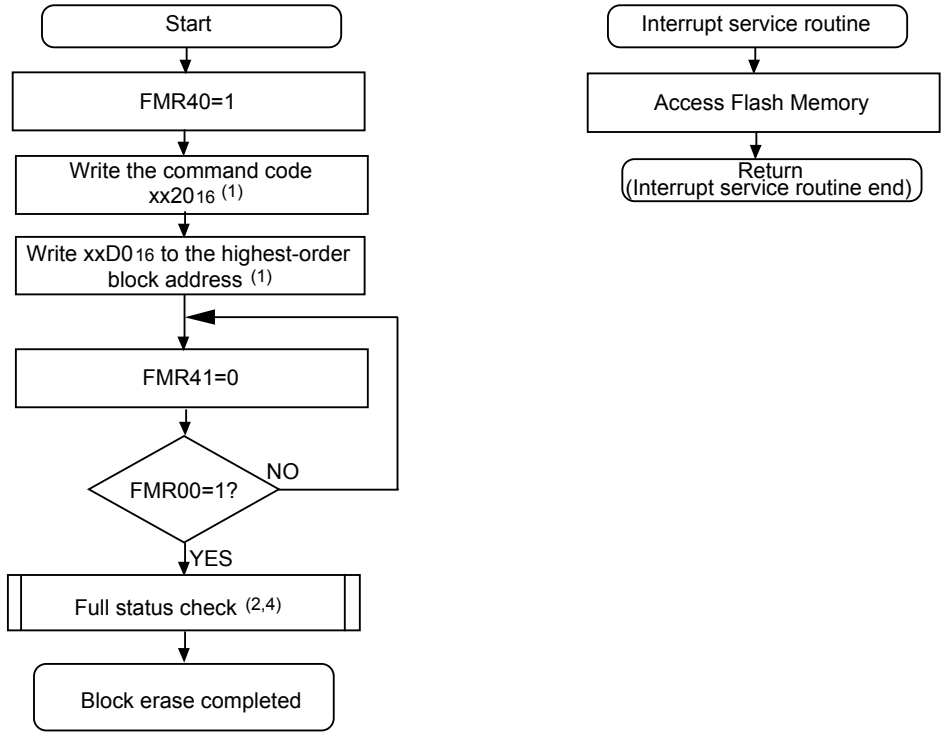
NOTES:
1. Write the command code and data at even address.
2. Refer to **Figure 19.13**.
3. Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is generated.

Figure 19.11 Flow Chart of Block Erase Command (when not using erase suspend function)

(EW mode 0)



(EW mode 1)



NOTES:
1. Write the command code and data to even address.
2. Execute the clear status register command and block erase command at least 3 times until an erase error is not generated when an erase error is occured.
3. In EW mode 0, allocate an interrupt vector table of an interrupt, to be used, to the RAM area.
4. Refer to **Figure 19.13**.

Figure 19.12 Block Erase Command (at use erase suspend)

19.8 Status Register

The status register indicates the operating status of the flash memory and whether or not erase or program operation is successfully completed. Bits FMR00, FMR06, and FMR07 in the FMR0 register indicate the status of the status register.

Table 19.6 lists the status register.

In EW mode 0, the status register can be read in the following cases:

(1) Any even address in the user ROM area is read after writing the read status register command

(2) Any even address in the user ROM area is read from when the program or block erase command is executed until when the read array command is executed.

19.8.1 Sequence Status (SR7 and FMR00 Bits)

The sequence status indicates the flash memory operating status. It is set to 0 (busy) while the auto-program and auto-erase operation is being executed and 1 (ready) as soon as these operations are completed. This bit indicates 0 (busy) in erase-suspend mode.

19.8.2 Erase Status (SR5 and FMR07 Bits)

Refer to **19.8.4 Full Status Check**.

19.8.3 Program Status (SR4 and FMR06 Bits)

Refer to **19.8.4 Full Status Check**.

Table 19.6 Status Register

Bits in the SRD Register	Bits in the FMR0 Register	Status Name	Contents		Value After Reset
			0	1	
SR7 (D_7)	FMR00	Sequence status	Busy	Ready	1
SR6 (D_6)	———	Reserved	-	-	———
SR5 (D_5)	FMR07	Erase status	Completed normally	Terminated by error	0
SR4 (D_4)	FMR06	Program status	Completed normally	Terminated by error	0
SR3 (D_3)	———	Reserved	-	-	———
SR2 (D_2)	———	Reserved	-	-	———
SR1 (D_1)	———	Reserved	-	-	———
SR0 (D_0)	———	Reserved	-	-	———

- D_7 to D_0: Indicates the data bus which is read out when executing the read status register command.
- The FMR07 bit (SR5) and FMR06 bit (SR4) are set to 0 by executing the clear status register command.
- When the FMR07 bit (SR5) or FMR06 bit (SR4) is set to 1, the program and block erase command are not accepted.

19.8.4 Full Status Check

If an error occurs, bits FMR06 to FMR07 in the FMR0 register are set to 1, indicating a specific error. Therefore, execution results can be comfirmed by verifying these status bits (full status check).

Table 19.7 lists errors and FMR0 register state. **Figure 19.13** shows a flow chart of the full status check and handling procedure for each error.

Table 19.7 Errors and FMR0 Register Status

FMR0 register (SRD register) status		Error	Error occurrence condition
FMR07 (SR5)	FMR06 (SR4)		
1	1	Command sequence error	• An incorrect commands is written • A value other than xxD0$_{16}$ or xxFF$_{16}$ is written in the second bus cycle of the block erase command (1) • When the block erase command is executed on an protected block • When the program command is executed on protected blocks
1	0	Erase error	• The block erase command is executed on an unprotected block but the program operation is not successfully completed
0	1	Program error	• The program command is executed on an unprotected block but the program operation is not successfully completed

NOTE:

1. The flash memory enters read array mode by writing command code xxFF$_{16}$ in the second bus cycle of these commands. The command code written in the first bus cycle becomes invalid.

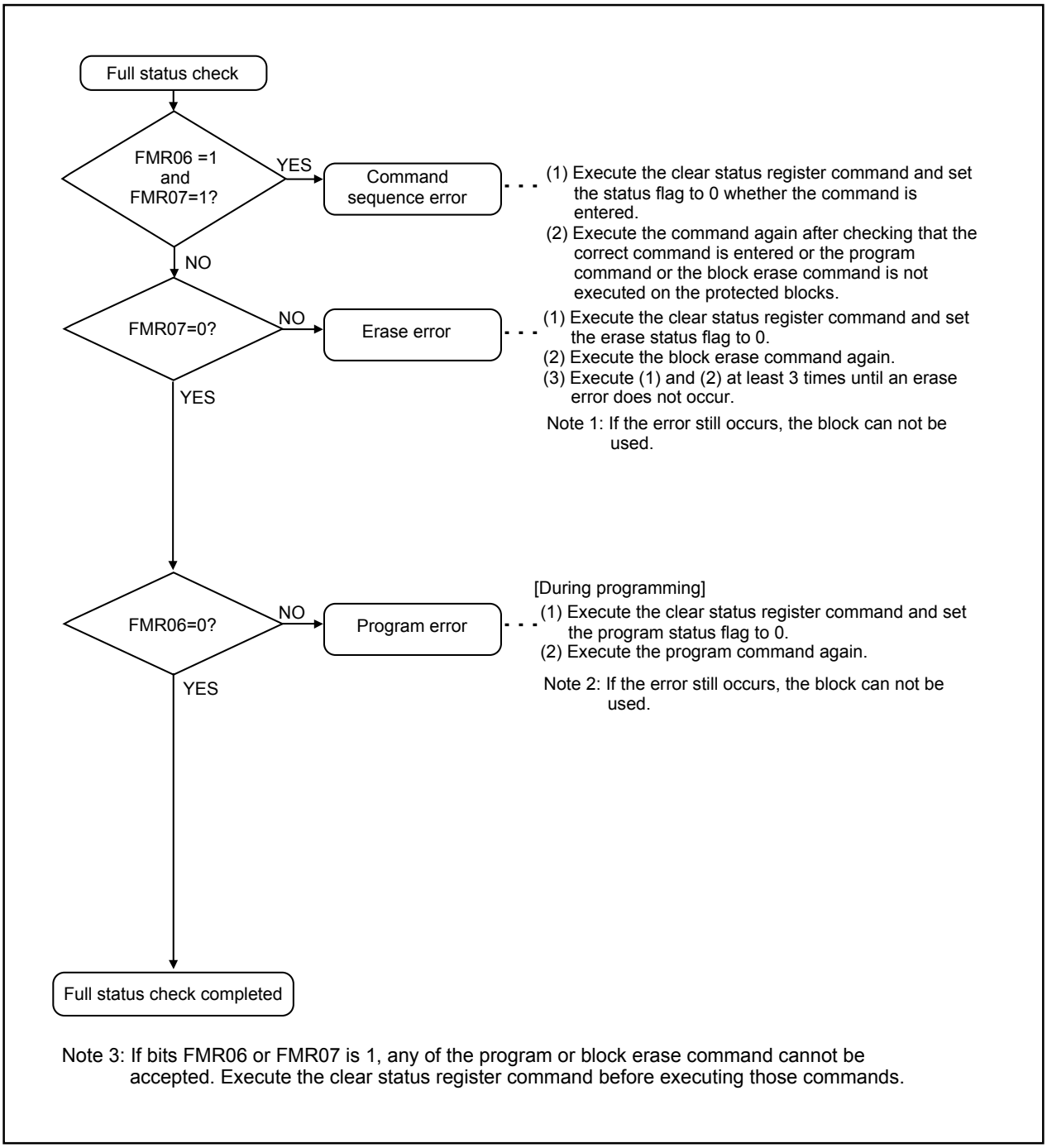


Figure 19.13 Full Status Check and Handling Procedure for Each Error

19.9 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the M16C/28 group (T-ver./V-ver.) can be used to rewrite the flash memory user ROM area, while the MCU is mounted on a board. For more information about the serial programmer, contact your serial programmer manufacturer. Refer to the user's manual included with your serial programmer for instruction.

Table 19.8 lists pin description (flash memory standard serial input/output mode). **Figures 19.14** and **19.15** show pin connections for standard serial input/output mode.

19.9.1 ID Code Check Function

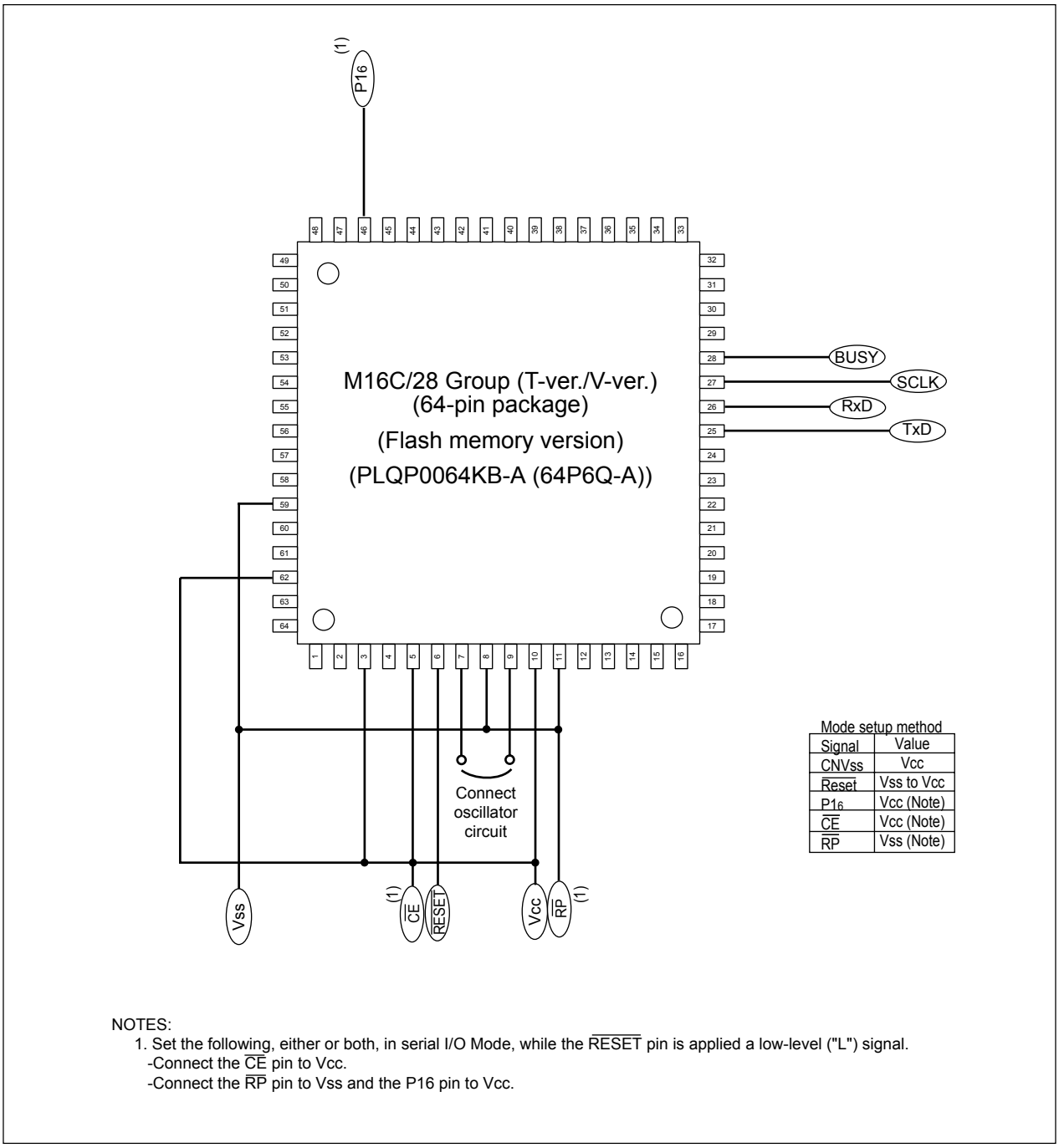
The ID code check function determines whether or not the ID codes sent from the serial programmer matches those written in the flash memory. (Refer to **19.3 Functions To Prevent Flash Memory from Rewriting**.)

Table 19.8 Pin Descriptions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Descriptio n
VCC,VSS	Power input		Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin.
CNVSS	CNVS S	I	Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is "L" level, wait for td(ROC).
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
AVCC, AVSS	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD conversion.
P00 to P07	Input port P0	I	Input "H" or "L" level signal or leave open.
P10 to P15, P17	Input port P1	I	Input "H" or "L" level signal or leave open.
P16	Input port P1	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ pin is “L”. (Note 2)
P20 to P27	Input port P2	I	Input "H" or "L" level signal or leave open.
P30 to P37	Input port P3	I	Input "H" or "L" level signal or leave open.
P60 to P63	Input port P6	I	Input "H" or "L" level signal or leave open.
P64	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitor signal output pin for boot program operation check
P65	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input "L".
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin (Note 1)
P70 to P77	Input port P7	I	Input "H" or "L" level signal or leave open.
P80 to P84, P87	Input port P8	I	Input "H" or "L" level signal or leave open.
P85	$\overline{\text{RP}}$ input	I	Connect this pin to Vss while $\overline{\text{RESET}}$ pin is “L”. (Note 2)
P86	$\overline{\text{CE}}$ input	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ pin is “L”. (Note 2)
P90 to P93, P95 to P97	Input port P9	I	Input "H" or "L" level signal or leave open.
P100 to P107	Input port P10	I	Input "H" or "L" level signal or leave open.

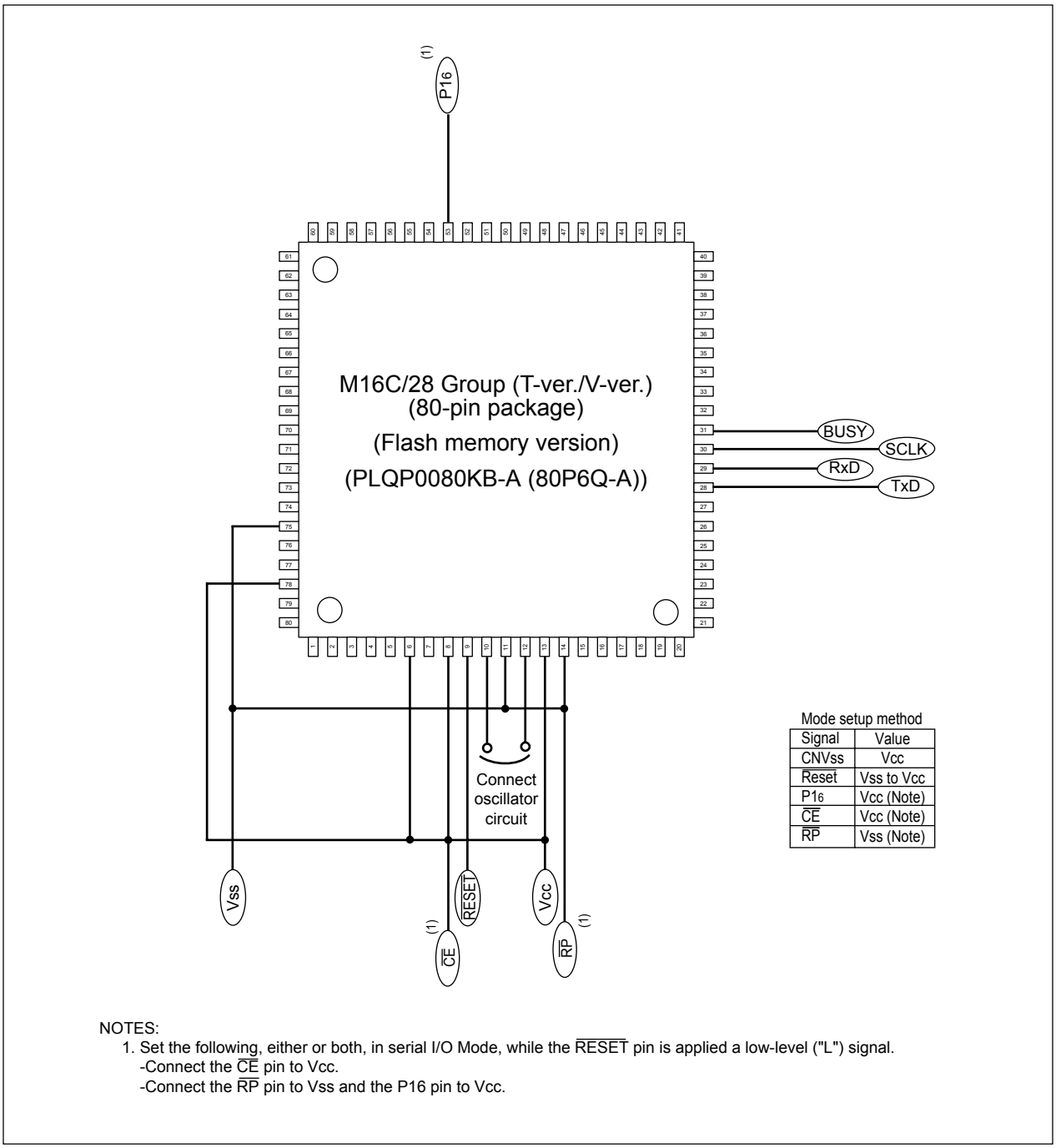
NOTES:

1. When using standard serial I/O mode 1, to input “H” to the TxD pin is necessary while the $\overline{\text{RESET}}$ pin is held “L”. Therefore, connect this pin to VCC via a resistor. Adjust the pull-up resistor value on a system not to affect a data transfer after reset, because this pin changes to a data-output pin
2. Set the following, either or both.
   -Connect the $\overline{\text{CE}}$ pin to VCC.
   -Connect the $\overline{\text{RP}}$ pin to VSS and P16 pin to VCC.



NOTES:
1. Set the following, either or both, in serial I/O Mode, while the $\overline{\text{RESET}}$ pin is applied a low-level ("L") signal.
   -Connect the $\overline{\text{CE}}$ pin to Vcc.
   -Connect the $\overline{\text{RP}}$ pin to Vss and the P16 pin to Vcc.

Figure 19.14 Pin Connections for Serial I/O Mode (1)

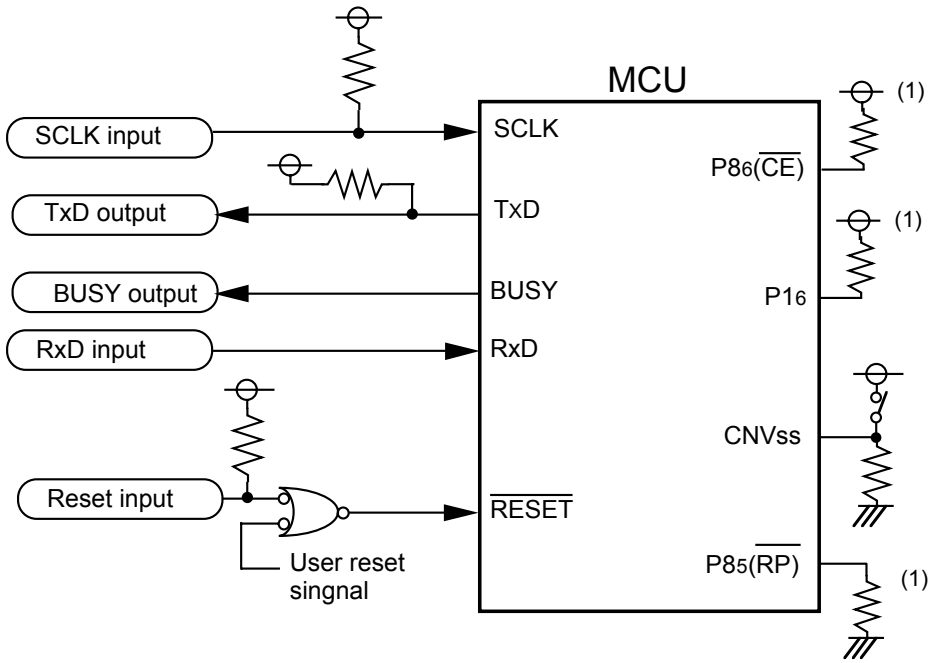


NOTES:
1. Set the following, either or both, in serial I/O Mode, while the RESET pin is applied a low-level ("L") signal.
   -Connect the CE pin to Vcc.
   -Connect the RP pin to Vss and the P16 pin to Vcc.

Figure 19.15 Pin Connections for Serial I/O Mode (2)

19.9.2 Example of Circuit Application in Standard Serial I/O Mode

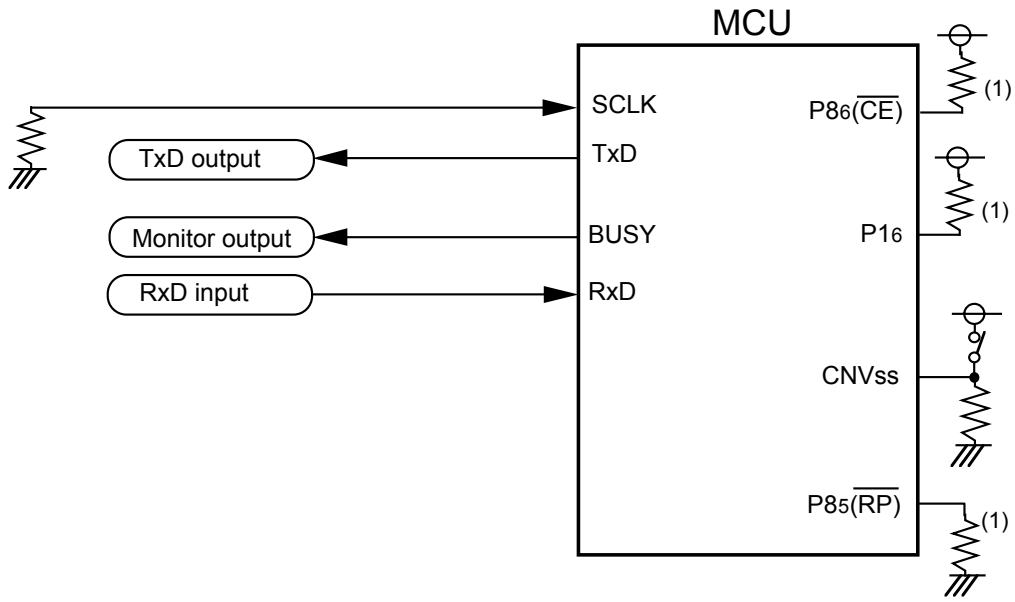
Figure 19.16 shows an example of a circuit application in standard serial I/O mode 1 and **Figure 19.17** shows an example of a circuit application in standard serial I/O mode 2. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.



(1) Controlling pins and external circuits vary with the serial programmer. For more information, refer to the user's manual included with the serial programmer.
(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) In standard serial input/output mode 1, if the user reset signal becomes “L” while the MCU is communicating with the serial programmer, break the connection between the user reset signal and the $\overline{\text{RESET}}$ pin using a jumper switch.

NOTE:
1. Set the following, either or both.
   - Connect the $\overline{\text{CE}}$ pin to Vcc
   - Connect the $\overline{\text{RP}}$ pin to Vss and the P1 6 pin to Vcc

Figure 19.16 Circuit Application in Standard Serial I/O Mode 1



(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.

NOTE:
1. Set the following, either or both.
   - Connect the $\overline{CE}$ pin to Vcc
   - Connect the $\overline{RP}$ pin to Vss and the P1 6 pin to Vcc

Figure 19.17 Circuit Application in Standard Serial I/O Mode 2

19.10 Parallel I/O Mode

In parallel input/output mode, the user ROM can be rewritten by a parallel programmer supporting the M16C/28 group (T-ver./V-ver.). Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

19.10.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read or rewritten. (Refer to **19.3 Functions To Prevent Flash Memory from Rewriting**).

20. Electrical Characteristics

20.1 T version

Table 20.1 Absolute Maximum Ratings

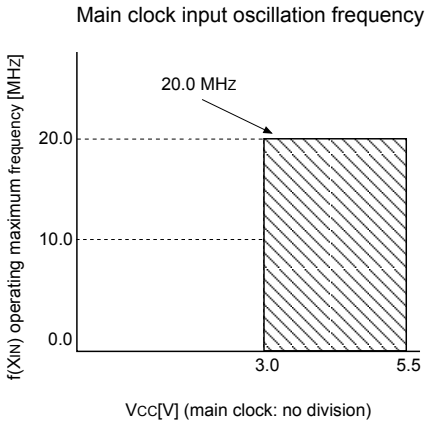
Symbol	Parameter			Condition	Value	Unit
V_{CC}	Supply Voltage			V_{CC}=AV_{CC}	-0.3 to 6.5	V
AV_{CC}	Analog Supply Voltage			V_{CC}=AV_{CC}	-0.3 to 6.5	V
V_I	Input Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, V_{REF}, $\overline{RESET}$, CNV_{SS}			-0.3 to V_{CC}+0.3	V
V_O	Output Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{OUT}			-0.3 to V_{CC}+0.3	V
Pd	Power Dissipation			-40≤Topr≤85° C	300	mW
Topr	Operating Ambient Temperature	during CPU operation			-40 to 85	° C
		during flash memory program and erase operation	Program Space (Block 0 to Block 4)		0 to 60	° C
			Data Space (Block A, Block B)		-40 to 85	° C
Tstg	Storage Temperature				-65 to 150	° C

Table 20.2 Recommended Operating Conditions (1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply Voltage		3.0		5.5	V
AV_{CC}	Analog Supply Voltage			V_{CC}		V
V_{SS}	Supply Voltage			0		V
AV_{SS}	Analog Supply Voltage			0		V
V_{IH}	Input High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$	0.7V_{CC}		V_{CC}	V
		XIN, ~~RESET~~, CNVSS	0.8V_{CC}		V_{CC}	V
		SDA_{MM}, SCL_{MM} When I^2C bus input level is selected	0.7V_{CC}		V_{CC}	V
		SDA_{MM}, SCL_{MM} When SMBUS input level is selected	1.4		V_{CC}	V
V_{IL}	Input Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$	0		0.3V_{CC}	V
		XIN, ~~RESET~~, CNVSS	0		0.2V_{CC}	V
		SDA_{MM}, SCL_{MM} When I^2C bus input level is selected	0		0.3V_{CC}	V
		SDA_{MM}, SCL_{MM} When SMBUS input level is selected	0		0.6	V
$I_{OH(peak)}$	Peak Output High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			-10.0	mA
$I_{OH(avg)}$	Average Output High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			-5.0	mA
$I_{OL(peak)}$	Peak Output Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			10.0	mA
$I_{OL(avg)}$	Average Output Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			5.0	mA
$f(X_{IN})$	Main Clock Input Frequency(4)		0		20	MHz
$f(X_{CIN})$	Sub Clock Frequency			32.768	50	kHz
f_1(ROC)	On-chip Oscillator Frequency 1		0.5	1	2	MHz
f2(ROC)	On-chip Oscillator Frequency 2		1	2	4	MHz
f3(ROC)	On-chip Oscillator Frequency 3		8	16	26	MHz
f(PLL)	PLL Clock Frequency(4)		10		20	MHz
f(BCLK)	CPU Operation Clock Frequency		0		20	ms
t_{SU}(PLL)	Wait Time to Stabilize PLL Frequency Synthesizer	V_{CC}=5.0V			20	ms
		V_{CC}=3.0V			50	ms

NOTES:

1. Referenced to V_{CC} = 3.0 to 5.5V at Topr = -40 to 85 ° C unless otherwise specified.
2. The mean output current is the mean value within 100ms.
3. The total $I_{OL(peak)}$ for all ports must be 80mA or less. The total $I_{OH(peak)}$ for all ports must be -80mA or less.
4. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.



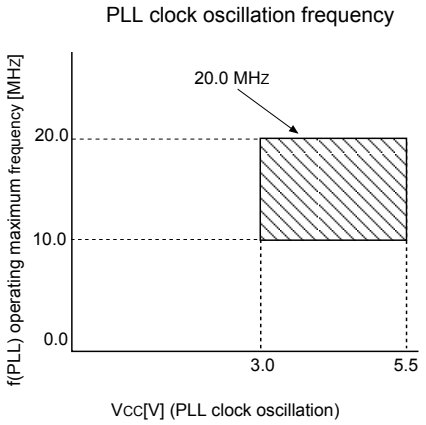


Table 20.3 A/D Conversion Characteristics (1)

Symbol	Parameter		Measurement Condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		V_{REF} = V_{CC}			10	Bits
INL	Integral Nonlinearity Error	10 bit	V_{REF} = V_{CC} = 5 V			±3	LSB
			V_{REF} = V_{CC} = 3.3 V			±5	LSB
		8 bit	V_{REF} = V_{CC} = 3.3 V			±2	LSB
-	Absolute Accuracy	10 bit	V_{REF} = V_{CC} = 5 V			±3	LSB
			V_{REF} = V_{CC} = 3.3 V			±5	LSB
		8 bit	V_{REF} = V_{CC} = 3.3 V			±2	LSB
DNL	Differential Nonlinearity Error					±1	LSB
-	Offset Error					±3	LSB
-	Gain Error					±3	LSB
R_{LADDER}	Resistor Ladder		V_{REF} = V_{CC}	10		40	kΩ
t_{CONV}	10-bit Conversion Time Sample & Hold Function Available		V_{REF} = V_{CC} = 5 V, øAD = 10 MHz	3.3			µs
t_{CONV}	8-bit Conversion Time Sample & Hold Function Available		V_{REF} = V_{CC} = 5 V, øAD = 10 MHz	2.8			µs
V_{REF}	Reference Voltage			2.0		V_{CC}	V
V_{IA}	Analog Input Voltage			0		V_{REF}	V

NOTES:
1. Referenced to V_{CC} = AV_{CC} = V_{REF} = 3.3 to 5.5 V, V_{SS} = AV_{SS} = 0 V at Topr = -40 to 85° C unless otherwise specified.
2. Keep ϕAD frequency at 10 MHz or less. Additionally, divide the f_{AD} if V_{CC} is less than 4.2V, and make ϕAD frequency equal to or lower than f_{AD}/2.
3. When sample & hold function is disabled, keep ϕAD frequency at 250kHz or more in addition to the limitation in Note 2. When sample & hold function is enabled, keep ϕAD frequency at 1MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is 3/ ϕAD frequency. When sample & hold function is disabled, sampling time is 2/ ϕAD frequency.

Table 20.4 Flash Memory Version Electrical Characteristics (1) for 100/1000 E/W cycle products [Program Space and Data Space in U3; Program Space in U7]

Symbol	Parameter		Standard			Unit
			Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3)		100/1000(4, 11)			cycles
-	Word Program Time (Vcc=5.0V, Topr=25° C)			75	600	µs
-	Block Erase Time (Vcc=5.0V, Topr=25° C)	2-Kbyte Block		0.2	9	s
		8-Kbyte Block		0.4	9	s
		16-Kbyte Block		0.7	9	s
		32-Kbyte Block		1.2	9	s
td(SR-ES)	Duration between Suspend Request and Erase Suspend				8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit				15	µs
-	Data Hold Time (5)		20			years

Table 20.5 Flash Memory Version Electrical Characteristics (6) for 10000 E/W cycle products

[Data Space in U7(7)]

Symbol	Parameter	Standard			Unit
		Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3, 8, 9)	10000(4, 10)			cycles
-	Word Program Time (Vcc=5.0V, Topr=25° C)		100		µs
-	Block Erase Time (Vcc=5.0V, Topr=25° C) (2-Kbyte block)		0.3		s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit			15	µs
-	Data Hold Time (5)	20			years

NOTES:
1. Referenced to VCC=3.0 to 5.5V at Topr=0 to 60° C(program space)/ Topr=-40 to 85° C(data space), unless otherwise specified.
2. VCC=5V; TOPR=25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.
   If program and erase endurance is *n* cycle (*n*=100, 1000, 10000), each block can be erased and programmed *n* cycles.
   For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guranteed (1 to minimum value are guaranteed).
5. Topr=55° C
6. Referenced to VCC=3.0 to 5.5V at Topr=-40 to 85° C unless otherwise specified.
7. **Table 20.5** applies for data space in B7 and U7 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 20.4**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If erase errors are occured during block erase, perform clear status register command and clock erase command in sequential order, at least three times, until erase errors disappear.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3; 1,000 cycles for program space in U7.
12. Customers desiring E/W failure rate information should contact their Renesas technical support representative.

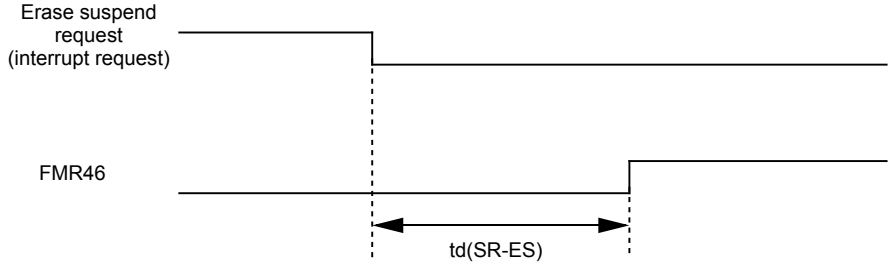
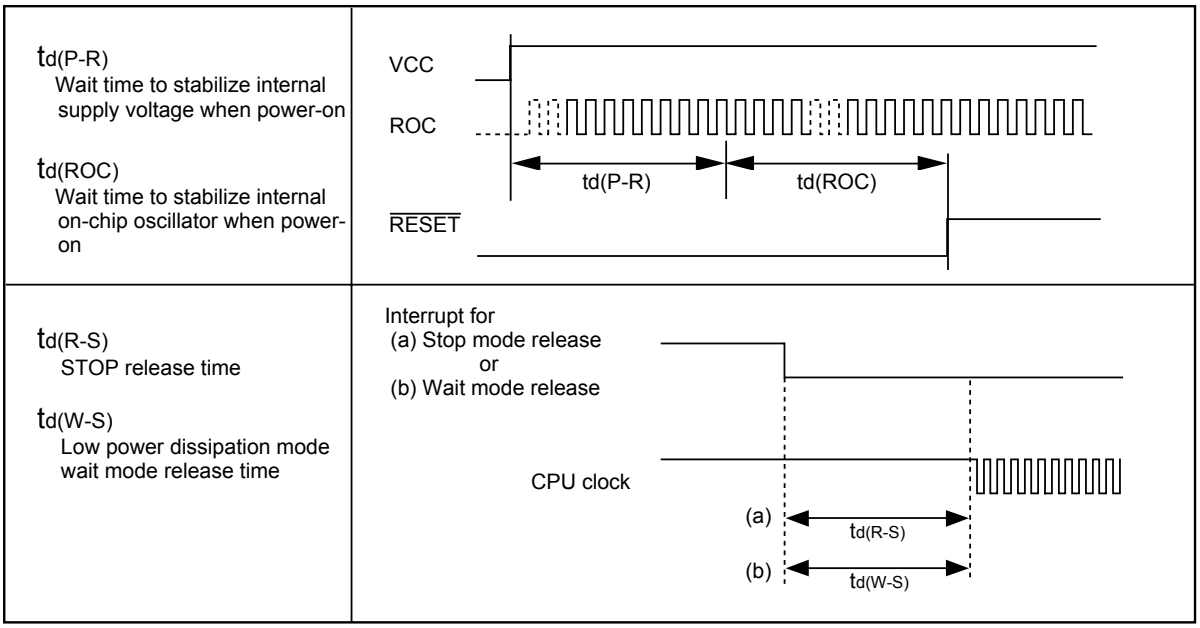


Table 20.6 Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait Time to Stabilize Internal Supply Voltage when Power-on	Vcc=3.0 to 5.5V			2	ms
td(ROC)	Wait Time to Stabilize Internal On-chip Oscillator when Power-on				40	µs
td(R-S)	STOP Release Time				150	µs
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				150	µs



V_{CC} = 5V

Table 20.7 Electrical Characteristics (1)

Symbol	Parameter			Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OH}=-5mA	V_{CC}-2.0		V_{CC}	V
V_{OH}	Output High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OH}=-200µA	V_{CC}-0.3		V_{CC}	V
V_{OH}	Output High ("H") Voltage	X_{OUT}	High Power	I_{OH}=-1mA	V_{CC}-2.0		V_{CC}	V
			Low Power	I_{OH}=-0.5mA	V_{CC}-2.0		V_{CC}	
	Output High ("H") Voltage	X_{COUT}	High Power	No load applied		2.5		V
			Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OL}=5mA			2.0	V
V_{OL}	Output Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OL}=200µA			0.45	V
V_{OL}	Output Low ("L") Voltage	X_{OUT}	High Power	I_{OL}=1mA			2.0	V
			Low Power	I_{OL}=0.5mA			2.0	
	Output Low ("L") Voltage	X_{COUT}	High Power	No load applied		0		V
			Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, $\overline{INT0}$-$\overline{INT5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$-$\overline{CTS2}$, SCL, SDA, $CLK0$-$CLK2$, $TA2_{OUT}$-$TA4_{OUT}$, $\overline{KI0}$-$\overline{KI3}$, $RXD0$-$RXD2$, $SIN3$, $SIN4$			0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$			0.2		2.5	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}			0.2		0.8	V
I_{IH}	Input High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=5V			5.0	µA
I_{IL}	Input Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=0V			-5.0	µA
R_{PULLUP}	Pull-up Resistance	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		V_I=0V	30	50	170	kΩ
Rf_{XIN}	Feedback Resistance	X_{IN}				1.5		MΩ
Rf_{XCIN}	Feedback Resistance	X_{CIN}				15		MΩ
V_{RAM}	RAM Standby Voltage			In stop mode	2.0			V

NOTE:

1. Referenced to V_{CC}=4.2 to 5.5V, V_{SS}=0V at Topr=-40 to 85 ° C, f(BCLK)=20MHz unless otherwise specified.

V_{CC} = 5V

Table 20.8 Electrical Characteristics (2) (1)

Symbol	Parameter	Measurement Condition			Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC}=4.2 to 5.5V)	Output pins are left open and other pins are connected to V_{SS}	Mask ROM	f(BCLK) = 20 MHz, main clock, no division		18	25	mA
				On-chip oscillation $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		2		mA
			Flash memory	f(BCLK) = 20 MHz, main clock, no division		18	25	mA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		2		mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Mask ROM	f(X_{CIN}) = 32 kHz, In low-power consumption mode, Program running on ROM(3)		25		µA
				On-chip oscillation $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM(3)		25		µA
				f(BCLK) = 32kHz, In low-power consumption mode, Program running on flash memory(3)		450		µA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity high		8.5		µA
				f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity low		3		µA
				While clock stops, Topr = 25° C		0.8	3	µA

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.9 External Clock Input (X_{IN} input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	50		ns
$t_{w(H)}$	External clock input HIGH pulse width	20		ns
$t_{w(L)}$	External clock input LOW pulse width	20		ns
t_r	External clock rise time		9	ns
t_f	External clock fall time		9	ns

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.10 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	100		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	40		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	40		ns

Table 20.11 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	400		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	200		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	200		ns

Table 20.12 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	200		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	100		ns

Table 20.13 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	100		ns

Table 20.14 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	400		ns

Table 20.15 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	800		ns
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	200		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	200		ns

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.16 Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	80		ns

Table 20.17 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 20.18 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 20.19 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	125		ns

Table 20.20 Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	70		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 20.21 External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	250		ns

V_{CC} = 5V

Timing Requirements

(V_{CC} = 5V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.22 Multi-master I^2C bus Line

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
tBUF	Bus free time	4.7		1.3		µs
tHD;STA	The hold time in start condition	4.0		0.6		µs
tLOW	The hold time in SCL clock "0" status	4.7		1.3		µs
tR	SCL, SDA signals' rising time		1000	20+0.1Cb	300	ns
tHD;DAT	Data hold time	0		0	0.9	µs
tHIGH	The hold time in SCL clock "1" status	4.0		0.6		µs
tF	SCL, SDA signals' falling time		300	20+0.1Cb	300	ns
tsu;DAT	Data setup time	250		100		ns
tsu;STA	The setup time in restart condition	4.7		0.6		µs
tsu;STO	Stop condition setup time	4.0		0.6		µs

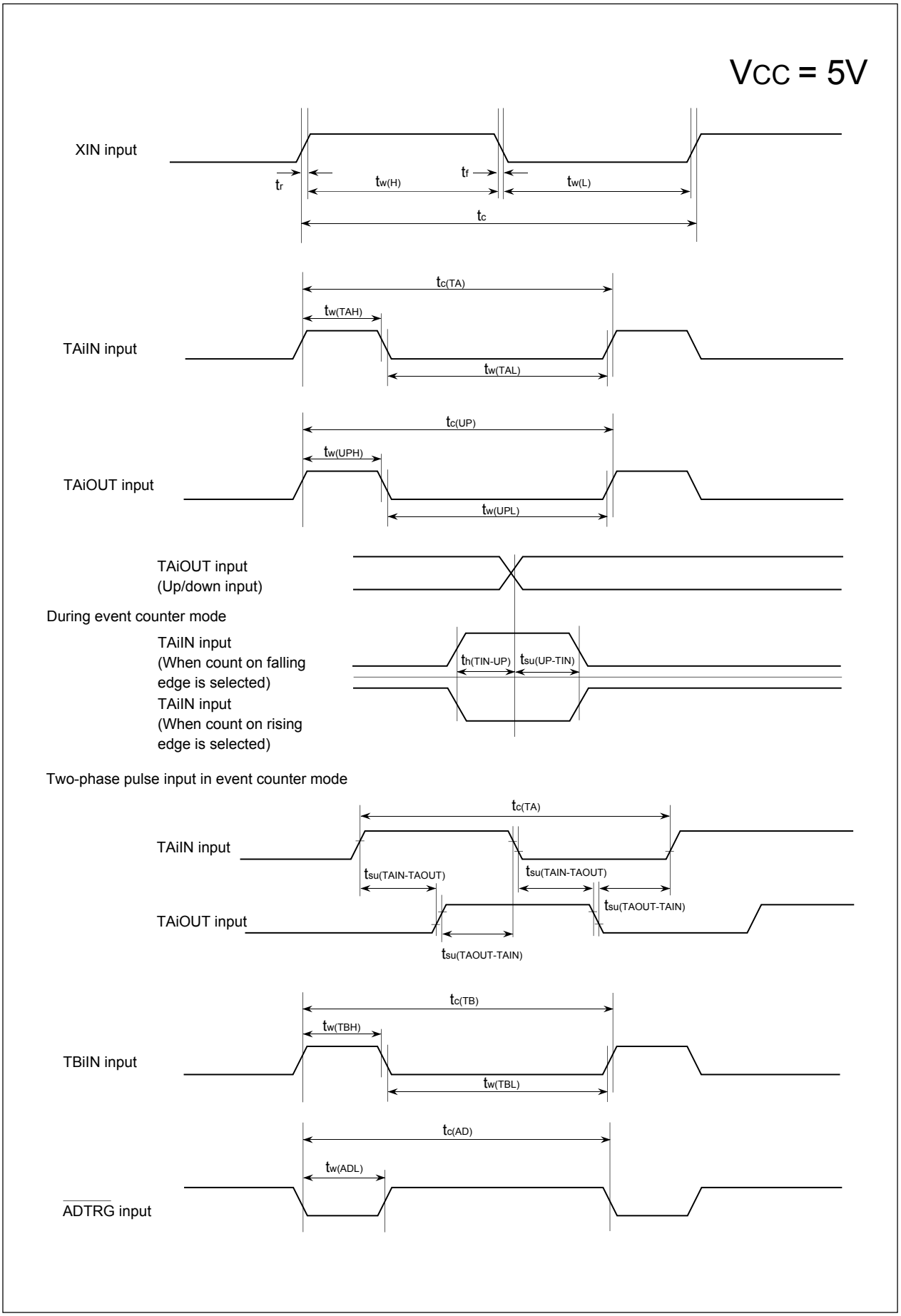


Figure 20.1 Timing Diagram (1)

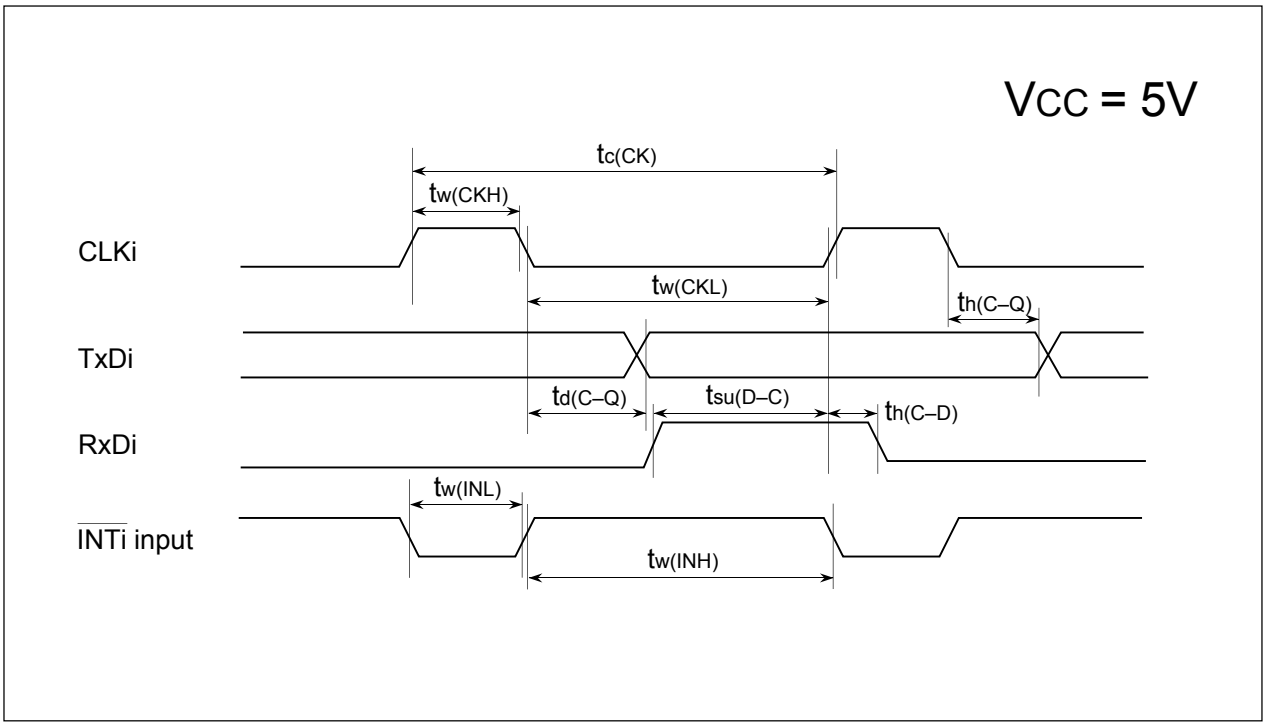


Figure 20.2 Timing Diagram (2)

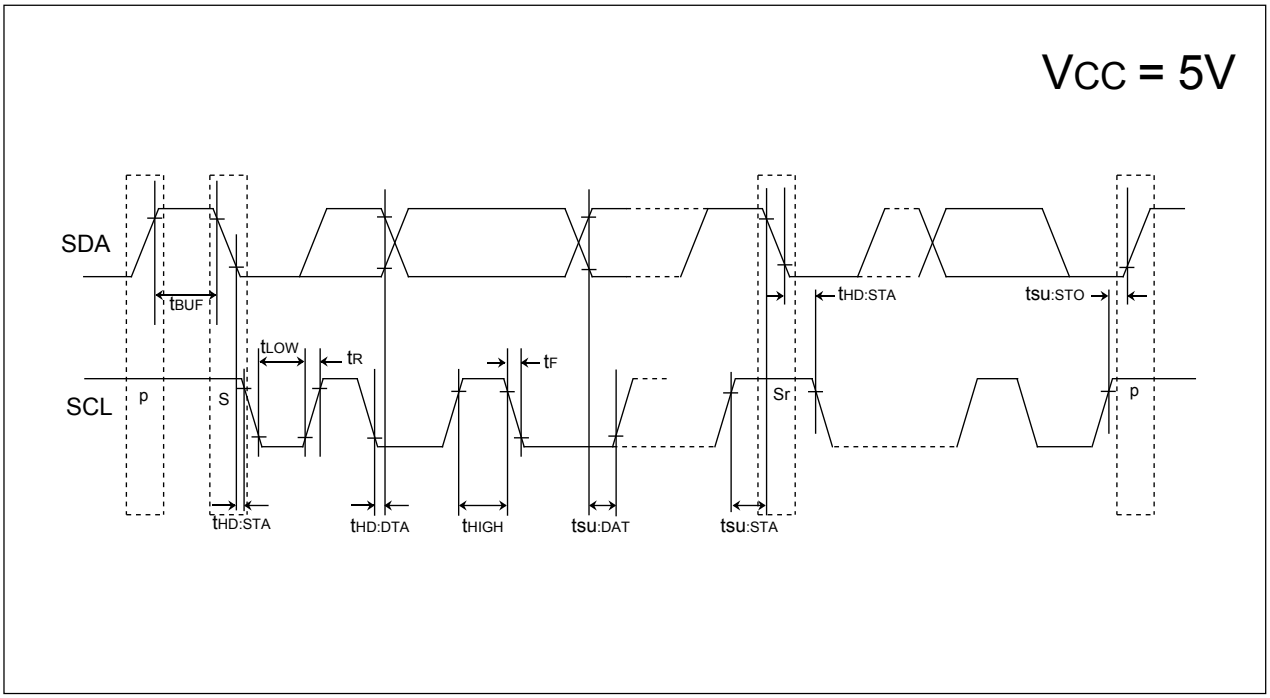


Figure 20.3 Timing Diagram (3)

Table 20.23 Electrical Characteristics (1)

V_{CC} = 3V

Symbol	Parameter				Condition	Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			I_{OH}=-1mA	V_{CC}-0.5		V_{CC}	V
V_{OH}	Output High ("H") Voltage		X_{OUT}	High Power	I_{OH}=-0.1mA	V_{CC}-0.5		V_{CC}	V
				Low Power	I_{OH}=-50µA	V_{CC}-0.5		V_{CC}	
	Output High ("H") Voltage		X_{COUT}	High Power	No load applied		2.5		V
				Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			I_{OL}=1mA			0.5	V
V_{OL}	Output Low ("L") Voltage		X_{OUT}	High Power	I_{OL}=0.1mA			0.5	V
				Low Power	I_{OL}=50µA			0.5	
	Output Low ("L") Voltage		X_{COUT}	High Power	No load applied		0		V
				Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, $\overline{INT0}$-$\overline{INT5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$-$\overline{CTS2}$, SCL, SDA, $CLK0$-$CLK2$, $TA2_{OUT}$-$TA4_{OUT}$, $\overline{KI0}$-$\overline{KI3}$, $RXD0$-$RXD2$, $SIN3$, $SIN4$						0.8	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$						1.8	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}						0.8	V
I_{IH}	Input High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}			V_I=3V			4.0	µA
I_{IL}	Input Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}			V_I=0V			-4.0	µA
R_{PULLUP}	Pull-up Resistance	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			V_I=0V	50	100	500	kΩ
Rf_{XIN}	Feedback Resistance		X_{IN}				3.0		MΩ
Rf_{XCIN}	Feedback Resistance		X_{CIN}				25		MΩ
V_{RAM}	RAM Standby Voltage				In stop mode	2.0			V

NOTE:

1. Referenced to V_{CC}=3.0 to 3.6V, V_{SS}=0V at Topr= -40 to 85 ° C, f(BCLK)=20MHz unless otherwise specified.

V_{CC} = 3V

Table 20.24 Electrical Characteristics (2) (1)

Symbol	Parameter	Measurement Condition			Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC}=3.0 to 3.6V)	Output pins are left open and other pins are connected to V_{SS}	Mask ROM	f(BCLK) = 10 MHz, PLL operated, no division		8	13	mA
				On-chip oscillation, f2(ROC) selected, f(BCLK) = 1 MHz		1		mA
			Flash memory	f(BCLK) = 10 MHz, No division		8	13	mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 3.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, Vcc = 3.0 V		11		mA
			Mask ROM	f(X_{CIN}) = 32 kHz, In low-power consumption mode, ROM (3)		20		µA
				On-chip oscillation, f2(ROC) selected, f(BCLK) = 1MHz, In wait mode		25		µA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM(3)		20		µA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory(3)		450		µA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		45		µA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity high		6.6		µA
				f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity low		2.2		µA
				While clock stops, Topr = 25° C		0.7	3	µA

NOTES:

1. Referenced to V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V at Topr = -40 to 85 ° C, f(BCLK) = 20 MHz unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.

V_{CC} = 3V

Timing Requirements

(V_{CC} = 3V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.25 External Clock Input (X_{IN} input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

V_{CC} = 3V

Timing Requirements

(V_{CC} = 3V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.26 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	60		ns

Table 20.27 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	300		ns

Table 20.28 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 20.29 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 20.30 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

Table 20.31 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	2		µs
$t_{su(TAIN-TAOUT)}$	TAiOUT input setup time	500		ns
$t_{su(TAOUT-TAIN)}$	TAiIN input setup time	500		ns

VCC = 3V

Timing Requirements

(VCC = 3V, VSS = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.32 Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	120		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	120		ns

Table 20.33 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table 20.34 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table 20.35 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input LOW pulse width	200		ns

Table 20.36 Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	100		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 20.37 External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	380		ns

V_{CC} = 3V

Timing Requirements

(V_{CC} = 3V, V_{SS} = 0V, at Topr = – 40 to 85°C unless otherwise specified)

Table 20.38 Multi-master I^2C bus Line

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
tBUF	Bus free time	4.7		1.3		µs
tHD;STA	The hold time in start condition	4.0		0.6		µs
tLOW	The hold time in SCL clock "0" status	4.7		1.3		µs
tR	SCL, SDA signals' rising time		1000	20+0.1Cb	300	ns
tHD;DAT	Data hold time	0		0	0.9	µs
tHIGH	The hold time in SCL clock "1" status	4.0		0.6		µs
tF	SCL, SDA signals' falling time		300	20+0.1Cb	300	ns
tSU;DAT	Data setup time	250		100		ns
tSU;STA	The setup time in restart condition	4.7		0.6		µs
tSU;STO	Stop condition setup time	4.0		0.6		µs

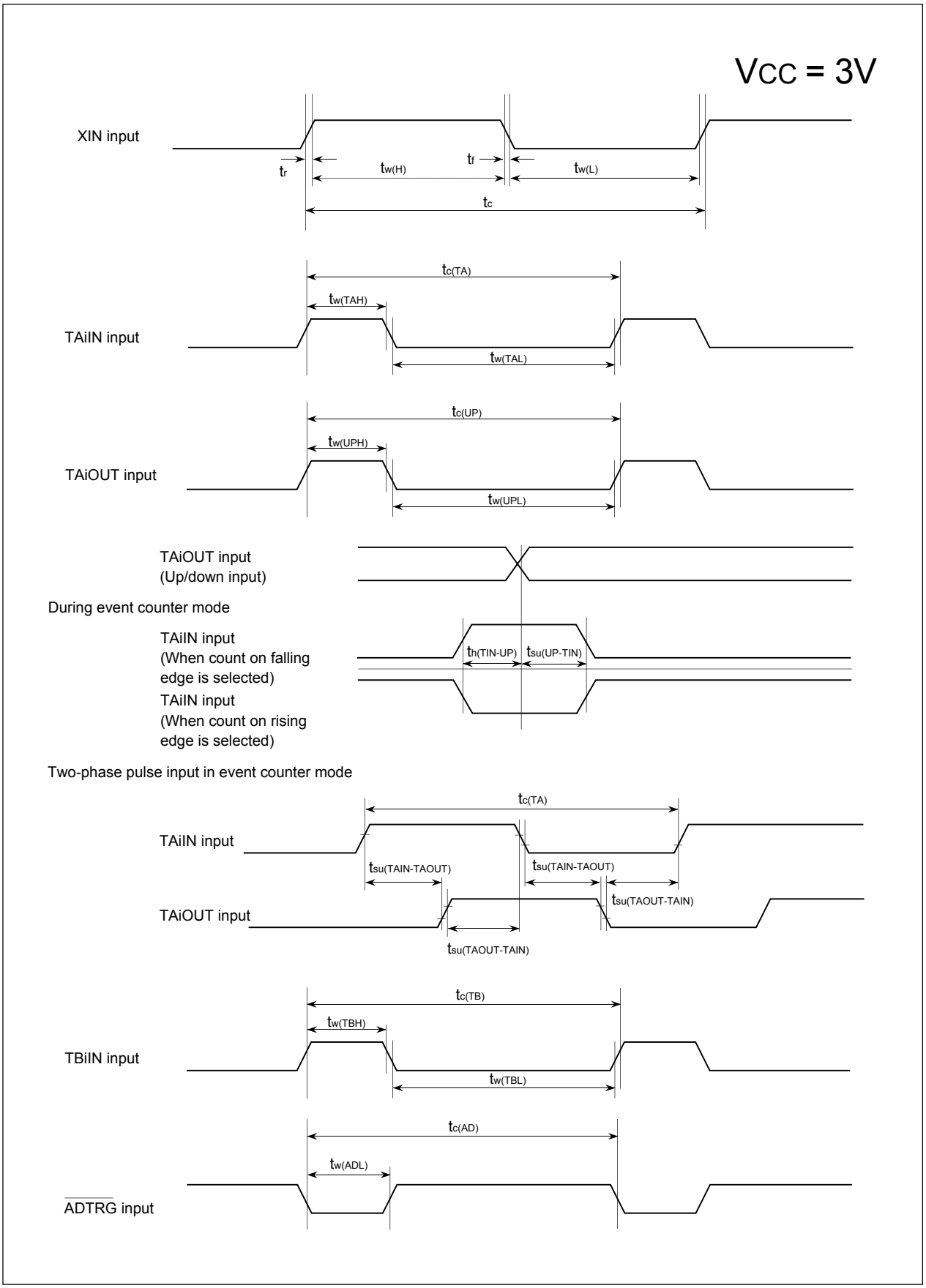


Figure 20.4 Timing Diagram (1)

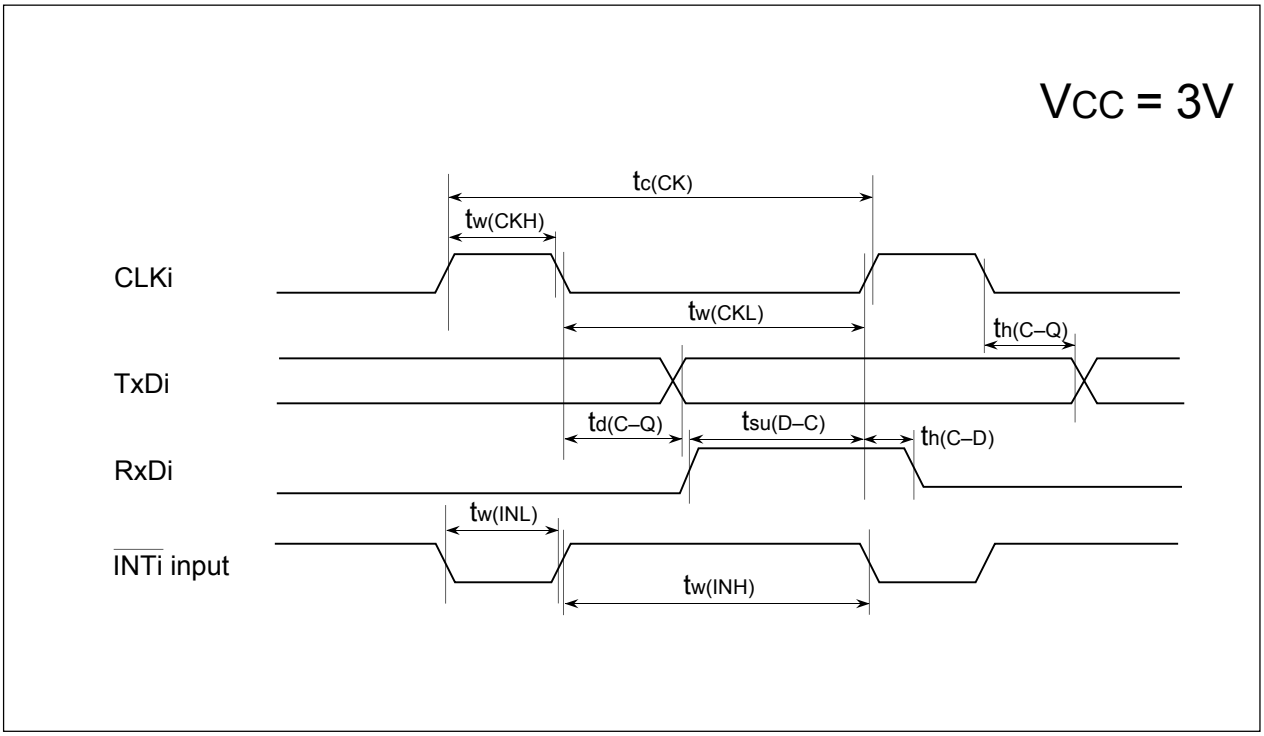


Figure 20.5 Timing Diagram (2)

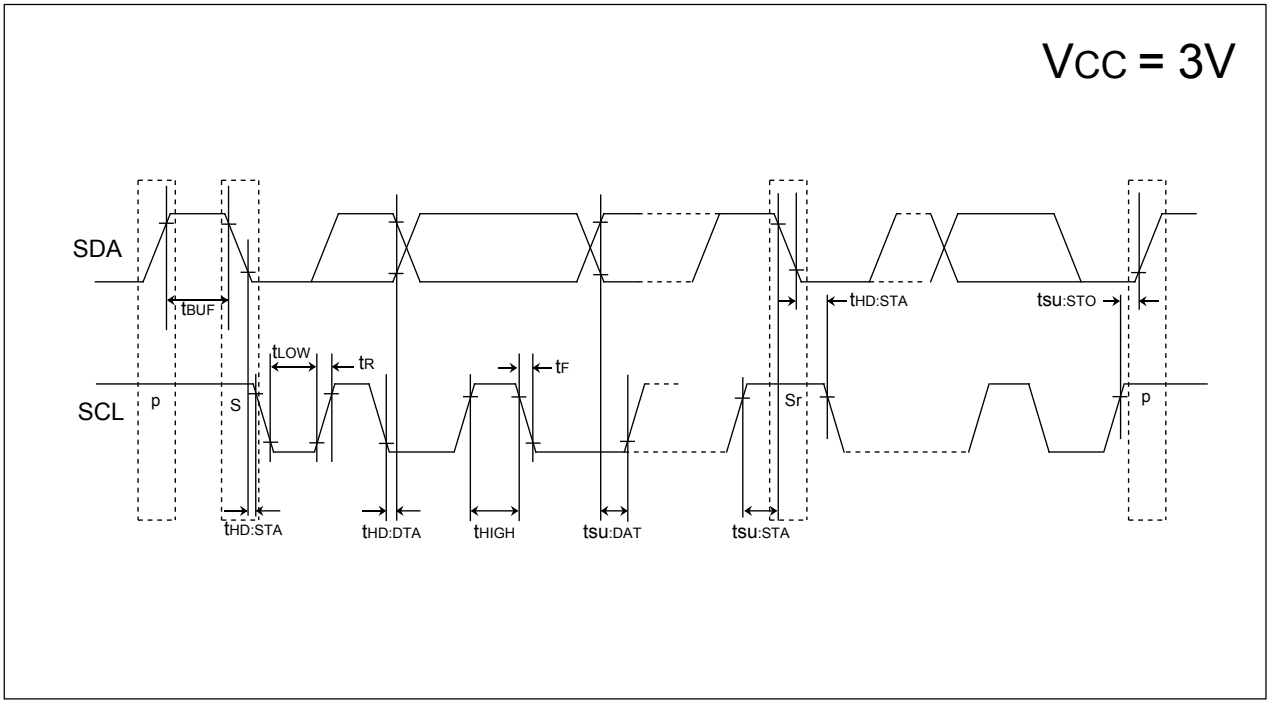


Figure 20.6 Timing Diagram (3)

20.2 V Version

Table 20.39 Absolute Maximum Ratings

Symbol	Parameter			Condition	Value	Unit
V_{CC}	Supply Voltage			V_{CC}=AV$_{CC}$	-0.3 to 6.5	V
AV$_{CC}$	Analog Supply Voltage			V_{CC}=AV$_{CC}$	-0.3 to 6.5	V
V_I	Input Voltage	P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, X_{IN}, V_{REF}, $\overline{\text{RESET}}$, CNV_{SS}			-0.3 to V_{CC}+0.3	V
V_O	Output Voltage	P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P6$_0$ to P6$_7$, P7$_0$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$ to P9$_3$, P9$_5$ to P9$_7$, P10$_0$ to P10$_7$, X_{OUT}			-0.3 to V_{CC}+0.3	V
Pd	Power Dissipation			-40≤Topr≤85° C	300	mW
				85≤Topr≤125° C	200	mW
Topr	Operating Ambient Temperature	during CPU operation			-40 to 125	° C
		during flash memory program and erase operation	Program Space (Block 0 to Block 4)		0 to 60	° C
			Data Space (Block A, Block B)		-40 to 125	° C
Tstg	Storage Temperature				-65 to 150	° C

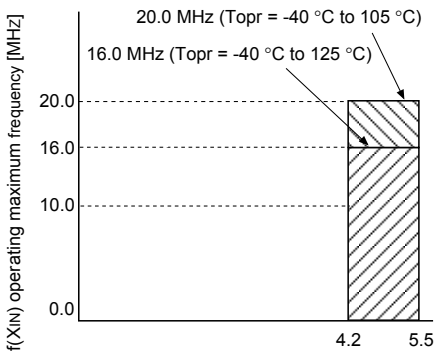
Table 20.40 Recommended Operating Conditions (1)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply Voltage		4.2		5.5	V
AVcc	Analog Supply Voltage			Vcc		V
V_{SS}	Supply Voltage			0		V
AVss	Analog Supply Voltage			0		V
V_{IH}	Input High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$	0.7Vcc		Vcc	V
		XIN, $\overline{RESET}$, CNVSS	0.8Vcc		Vcc	V
V_{IL}	Input Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$	0		0.3Vcc	V
		XIN, $\overline{RESET}$, CNVSS	0		0.2Vcc	V
$I_{OH(peak)}$	Peak Output High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			-10.0	mA
$I_{OH(avg)}$	Average Output High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			-5.0	mA
$I_{OL(peak)}$	Peak Output Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			10.0	mA
$I_{OL(avg)}$	Average Output Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			5.0	mA
f(X_{IN})	Main Clock Input Frequency(3)	Topr=-40 to 105° C	0		20	MHz
		Topr=-40 to 125° C	0		16	MHz
f(X_{CIN})	Sub Clock Frequency			32.768	50	kHz
f_1(ROC)	On-chip Oscillator Frequency 1		0.5	1	2	MHz
f2(ROC)	On-chip Oscillator Frequency 2		1	2	4	MHz
f3(ROC)	On-chip Oscillator Frequency 3		8	16	26	MHz
f(PLL)	PLL Clock Frequency(3)	Topr=-40 to 105° C	10		20	MHz
		Topr=-40 to 125° C	10		16	MHz
f(BCLK)	CPU Operation Clock Frequency	Topr=-40 to 105° C	0		20	ms
		Topr=-40 to 125° C	0		16	ms
t_{SU}(PLL)	Wait Time to Stabilize PLL Frequency Synthesizer	Vcc=5.0V			20	ms

NOTES:

1. Referenced to Vcc = 4.2 to 5.5V at Topr = -40 to 125 ° C unless otherwise specified.
2. The mean output current is the mean value within 100ms.
3. Relationship among main clock oscillation frequency, PLL clock oscillation frequency and supply voltage.
4. The total $I_{OL(peak)}$ for all ports must be 80mA or less. The total $I_{OH(peak)}$ for all ports must be -80mA or less.

Main clock input oscillation frequency



PLL clock oscillation frequency

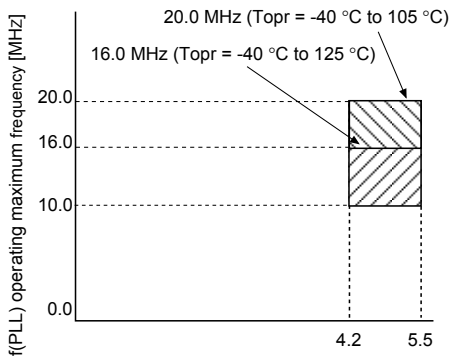


Table 20.41 A/D Conversion Characteristics (1)

Symbol	Parameter		Measurement Condition	Standard			Unit
				Min.	Typ.	Max.	
-	Resolution		V_{REF} = V_{CC}			10	Bits
INL	Integral Nonlinearity Error	10 bit	V_{REF} = V_{CC} = 5 V			±3	LSB
		8 bit	V_{REF} = V_{CC} = 5 V			±2	LSB
-	Absolute Accuracy	10 bit	V_{REF} = V_{CC} = 5 V			±3	LSB
		8 bit	V_{REF} = V_{CC} = 5 V			±2	LSB
DNL	Differential Nonlinearity Error					±1	LSB
-	Offset Error					±3	LSB
-	Gain Error					±3	LSB
R_{LADDER}	Resistor Ladder		V_{REF} = V_{CC}	10		40	kΩ
t_{CONV}	10-bit Conversion Time Sample & Hold Function Available		V_{REF} = V_{CC} = 5 V, øAD = 10 MHz	3.3			µs
t_{CONV}	8-bit Conversion Time Sample & Hold Function Available		V_{REF} = V_{CC} = 5 V, øAD = 10 MHz	2.8			µs
V_{REF}	Reference Voltage			2.0		V_{CC}	V
V_{IA}	Analog Input Voltage			0		V_{REF}	V

NOTES:
1. Referenced to V_{CC} = AV_{CC} = V_{REF} = 4.2 to 5.5 V, V_{SS} = AV_{SS} = 0 V at Topr = -40 to 125 ° C unless otherwise specified.
2. Keep ϕAD frequency at 10 MHz or less.
3. When sample & hold function is disabled, keep ϕAD frequency at 250kHz or more in addition to the limitation in Note 2.
   When sample & hold function is enabled, keep ϕAD frequency at 1MHz or more in addition to the limitation in Note 2.
4. When sample & hold function is enabled, sampling time is 3/ ϕAD frequency.
   When sample & hold function is disabled, sampling time is 2/ ϕAD frequency.

Table 20.42 Flash Memory Version Electrical Characteristics (1) for 100/1000 E/W cycle products

[Program Space and Data Space in U3; Program Space in U7]

Symbol	Parameter		Standard			Unit
			Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3)		100/1000(4, 11)			cycles
-	Word Program Time (Vcc=5.0V, Topr=25° C)			75	600	µs
-	Block Erase Time (Vcc=5.0V, Topr=25° C)	2-Kbyte Block		0.2	9	s
		8-Kbyte Block		0.4	9	s
		16-Kbyte Block		0.7	9	s
		32-Kbyte Block		1.2	9	s
td(SR-ES)	Duration between Suspend Request and Erase Suspend				8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit				15	µs
-	Data Hold Time (5)		20			years

Table 20.43 Flash Memory Version Electrical Characteristics (6) for 10000 E/W cycle products

[Data Space in U7 (7)]

Symbol	Parameter	Standard			Unit
		Min.	Typ.(2)	Max.	
-	Program and Erase Endurance(3, 8, 9)	10000(4, 10)			cycles
-	Word Program Time (Vcc=5.0V, Topr=25° C)		100		µs
-	Block Erase Time (Vcc=5.0V, Topr=25° C) (2-Kbyte block)		0.3		s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
tPS	Wait Time to Stabilize Flash Memory Circuit			15	µs
-	Data Hold Time (5)	20			years

NOTES:

1. Referenced to VCC=4.2 to 5.5V at Topr=0 to 60° C(program space)/ Topr=-40 to 125° C(data space), unless otherwise specified.
2. VCC=5V; TOPR=25° C
3. Program and erase endurance is defined as number of program-erase cycles per block.
   If program and erase endurance is *n* cycle (*n*=100, 1000, 10000), each block can be erased and programmed *n* cycles.
   For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guranteed (1 to minimum value are guranteed).
5. Topr=55° C
6. Referenced to VCC=4.2 to 5.5V at Topr=-40 to 125° C unless otherwise specified.
7. **Table 20.42** applies for data space in B7 and U7 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 20.43**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If erase errors are occured during block erase, perform clear status register command and clock erase command in sequential order, at least three times, until erase errors disappear.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3; 1,000 cycles for program space in U7.
12. Customers desiring E/W failure rate information should contact their Renesas technical support representative.

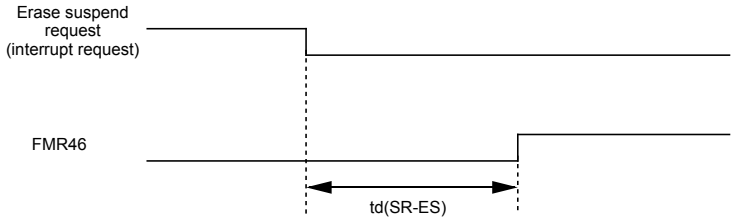
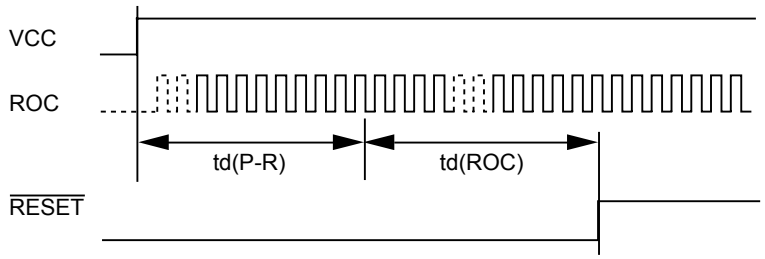
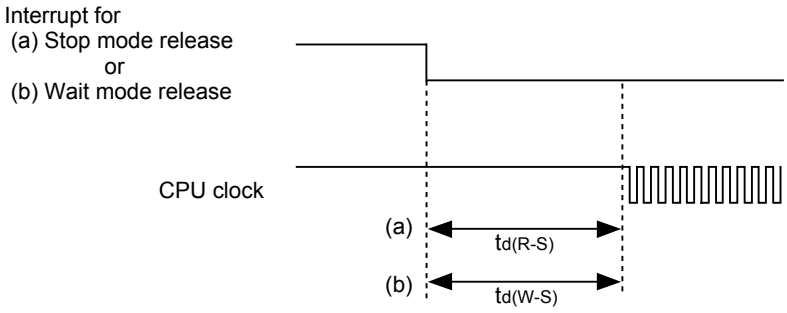


Table 20.44 Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measurement Condition	Standard			Unit
			Min.	Typ.	Max.	
td(P-R)	Wait Time to Stabilize Internal Supply Voltage when Power-on	Vcc=4.2 to 5.5V			2	ms
td(ROC)	Wait Time to Stabilize Internal On-chip Oscillator when Power-on				40	µs
td(S-R)	STOP Release Time				150	µs
td(W-S)	Low Power Dissipation Mode Wait Mode Release Time				150	µs

<table>
<tr>
<td>td(P-R)
Wait time to stabilize internal supply voltage when power-on

td(ROC)
Wait time to stabilize internal on-chip oscillator when power-on</td>
<td> 
</td>
</tr>
<tr>
<td>td(R-S)
STOP release time

td(W-S)
Low power dissipation mode wait mode release time</td>
<td> 
</td>
</tr>
</table>

V_{CC} = 5V

Table 20.45 Electrical Characteristics (1)

Symbol	Parameter				Condition	Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			I_{OH}=-5mA	V_{CC}-2.0		V_{CC}	V
V_{OH}	Output High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			I_{OH}=-200µA	V_{CC}-0.3		V_{CC}	V
V_{OH}	Output High ("H") Voltage		X_{OUT}	High Power	I_{OH}=-1mA	V_{CC}-2.0		V_{CC}	V
				Low Power	I_{OH}=-0.5mA	V_{CC}-2.0		V_{CC}	
	Output High ("H") Voltage		X_{COUT}	High Power	No load applied		2.5		V
				Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			I_{OL}=5mA			2.0	V
V_{OL}	Output Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			I_{OL}=200µA			0.45	V
V_{OL}	Output Low ("L") Voltage		X_{OUT}	High Power	I_{OL}=1mA			2.0	V
				Low Power	I_{OL}=0.5mA			2.0	
	Output Low ("L") Voltage		X_{COUT}	High Power	No load applied		0		V
				Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, $\overline{INT_0}$-$\overline{INT_5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS_0}$-$\overline{CTS_2}$, SCL, SDA, CLK_0-CLK_2, $TA2_{OUT}$-$TA4_{OUT}$, $\overline{KI_0}$-$\overline{KI_3}$, RXD_0-RXD_2, SIN_3, SIN_4				0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$				0.2		2.5	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}				0.2		0.8	V
I_{IH}	Input High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}			V_I=5V			5.0	µA
I_{IL}	Input Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}			V_I=0V			-5.0	µA
R_{PULLUP}	Pull-up Resistance	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$			V_I=0V	30	50	170	kΩ
Rf_{XIN}	Feedback Resistance		X_{IN}				1.5		MΩ
Rf_{XCIN}	Feedback Resistance		X_{CIN}				15		MΩ
V_{RAM}	RAM Standby Voltage				In stop mode	2.0			V

NOTES:

1. Referenced to V_{CC}=4.2 to 5.5V, V_{SS}=0V at Topr=-40 to 105 ° C, f(BCLK)=20MHz / V_{CC}=4.2 to 5.5V, V_{SS}=0V at Topr= -40 to 125 ° C, f(BCLK)=16MHz, unless otherwise specified.

Vcc = 5V

Table 20.46 Electrical Characteristics (2) (1)

Symbol	Parameter	Measurement Condition			Standard			Unit
					Min.	Typ.	Max.	
I_{CC}	Power Supply Current (V_{CC} = 4.2 to 5.5 V)	Output pins are left open and other pins are connected to V_{SS}	Mask ROM	f(BCLK) = 20 MHz, main clock, no division		18	25	mA
				On-chip oscillation $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		2		mA
			Flash memory	f(BCLK) = 20 MHz, main clock, no division		18	25	mA
				f(BCLK) = 16 MHz, main clock, no division		14	20	mA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz		2		mA
			Flash memory program	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Flash memory erase	f(BCLK) = 10 MHz, Vcc = 5.0 V		11		mA
			Mask ROM	f(X_{CIN}) = 32 kHz, In low-power consumption mode, Program running on ROM(3)		25		µA
				On-chip oscillation $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
			Flash memory	f(BCLK) = 32 kHz, In low-power consumption mode, Program running on RAM(3)		25		µA
				f(BCLK) = 32 kHz, In low-power consumption mode, Program running on flash memory(3)		450		µA
				On-chip oscillation, $f_{2(ROC)}$ selected, f(BCLK) = 1 MHz, In wait mode		50		µA
			Mask ROM, Flash memory	f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity high		8.5		µA
				f(BCLK) = 32 kHz, In wait mode(2), Oscillation capacity low		3		µA
				While clock stops, Topr = 25° C		0.8	3	µA

NOTES:

1. Referenced to V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -40 to 105 ° C, f(BCLK) = 20 MHz / V_{CC} = 4.2 to 5.5 V, V_{SS} = 0 V at Topr = -40 to 125 ° C, f(BCLK) = 16 MHz, unless otherwise specified.
2. With one timer operates, using f_{C32}.
3. This indicates the memory in which the program to be executed exists.

V_{CC} = 5V

Timing Requirements

(Vcc=5V, Vss=0V, at Topr=-40 to 125°C unless otherwise specified)

Table 20.47 External Clock Input (XIN input)

Symbol	Parameter		Standard		Unit
			Min.	Max.	
tc	External Clock Input Cycle Time	Topr=-40° C to 105° C	50		ns
		Topr=-40° C to 125° C	62.5		ns
$tw_{(H)}$	External Clock Input High ("H") Width	Topr=-40° C to 105° C	20		ns
		Topr=-40° C to 125° C	25		ns
$tw_{(L)}$	External Clock Input Low ("L") Width	Topr=-40° C to 105° C	20		ns
		Topr=-40° C to 125° C	25		ns
tr	External Clock Rise Time	Topr=-40° C to 105° C		9	ns
		Topr=-40° C to 125° C		15	ns
tf	External Clock Fall Time	Topr=-40° C to 105° C		9	ns
		Topr=-40° C to 125° C		15	ns

Vcc = 5V

Timing Requirements

(Vcc=5V, Vss=0V, at Topr=-40 to 125°C unless otherwise specified)

Table 20.48 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} Input Cycle Time	100		ns
$t_{w(TAH)}$	TAi_{IN} Input High ("H") Width	40		ns
$t_{w(TAL)}$	TAi_{IN} Input Low ("L") Width	40		ns

Table 20.49 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} Input Cycle Time	400		ns
$t_{w(TAH)}$	TAi_{IN} Input High ("H") Width	200		ns
$t_{w(TAL)}$	TAi_{IN} Input Low ("L") Width	200		ns

Table 20.50 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} Input Cycle Time	200		ns
$t_{w(TAH)}$	TAi_{IN} Input High ("H") Width	100		ns
$t_{w(TAL)}$	TAi_{IN} Input Low ("L") Width	100		ns

Table 20.51 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAi_{IN} Input High ("H") Width	100		ns
$t_{w(TAL)}$	TAi_{IN} Input Low ("L") Width	100		ns

Table 20.52 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAi_{OUT} Input Cycle Time	2000		ns
$t_{w(UPH)}$	TAi_{OUT} Input High ("H") Width	1000		ns
$t_{w(UPL)}$	TAi_{OUT} Input Low ("L") Width	1000		ns
$t_{su(UP-TIN)}$	TAi_{OUT} Input Setup Time	400		ns
$t_{h(TIN-UP)}$	TAi_{OUT} Input Hold Time	400		ns

Table 20.53 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} Input Cycle Time	800		ns
$t_{su(TAIN-TAOUT)}$	TAi_{OUT} Input Setup Time	200		ns
$t_{su(TAOUT-TAIN)}$	TAi_{IN} Input Setup Time	200		ns

VCC = 5V

Timing Requirements

(VCC=5V, VSS=0V, at Topr=-40 to 125°C unless otherwise specified)

Table 20.54 Timer B Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN Input High ("H") Width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN Input Low ("L") Width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN Input Cycle Time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN Input High ("H") Width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN Input Low ("L") Width (counted on both edges)	80		ns

Table 20.55 Timer B Input (Pulse Period Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input High ("H") Width	200		ns
$t_{w(TBL)}$	TBiIN Input Low ("L") Width	200		ns

Table 20.56 Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN Input Cycle Time	400		ns
$t_{w(TBH)}$	TBiIN Input High ("H") Width	200		ns
$t_{w(TBL)}$	TBiIN Input Low ("L") Width	200		ns

Table 20.57 A/D Trigger Input

Symbol	Parameter	Standard		Unit
		Min.	Max	
$t_{c(AD)}$	$\overline{ADTRG}$ Input Cycle Time (required for trigger)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ Input Low ("L") Width	125		ns

Table 20.58 Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi Input Cycle Time	200		ns
$t_{w(CKH)}$	CLKi Input High ("H") Width	100		ns
$t_{w(CKL)}$	CLKi Input Low ("L") Width	100		ns
$t_{d(C-Q)}$	TxDi Output Delay Time		80	ns
$t_{h(C-Q)}$	TxDi Hold Time	0		ns
$t_{su(D-C)}$	RxDi Input Setup Time	70		ns
$t_{h(C-Q)}$	RxDi Input Hold Time	90		ns

Table 20.59 External Interrupt $\overline{INTi}$ Input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ Input High ("H") Width	250		ns
$t_{w(INL)}$	$\overline{INTi}$ Input Low ("L") Width	250		ns

Vcc = 5V

Timing Requirements

(Vcc=5V, Vss=0V, at Topr=-40 to 125°C unless otherwise specified)

Table 20.60 Multi-master I²C Bus Line

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
tBUF	Bus free time	4.7		1.3		µs
tHD;STA	The hold time in start condition	4.0		0.6		µs
tLOW	The hold time in SCL clock "0" status	4.7		1.3		µs
tR	SCL, SDA signals' rising time		1000	20+0.1Cb	300	ns
tHD;DAT	Data hold time	0		0	0.9	µs
tHIGH	The hold time in SCL clock "1" status	4.0		0.6		µs
tF	SCL, SDA signals' falling time		300	20+0.1Cb	300	ns
tsu;DAT	Data setup time	250		100		ns
tsu;STA	The setup time in restart condition	4.7		0.6		µs
tsu;STO	Stop condition setup time	4.0		0.6		µs

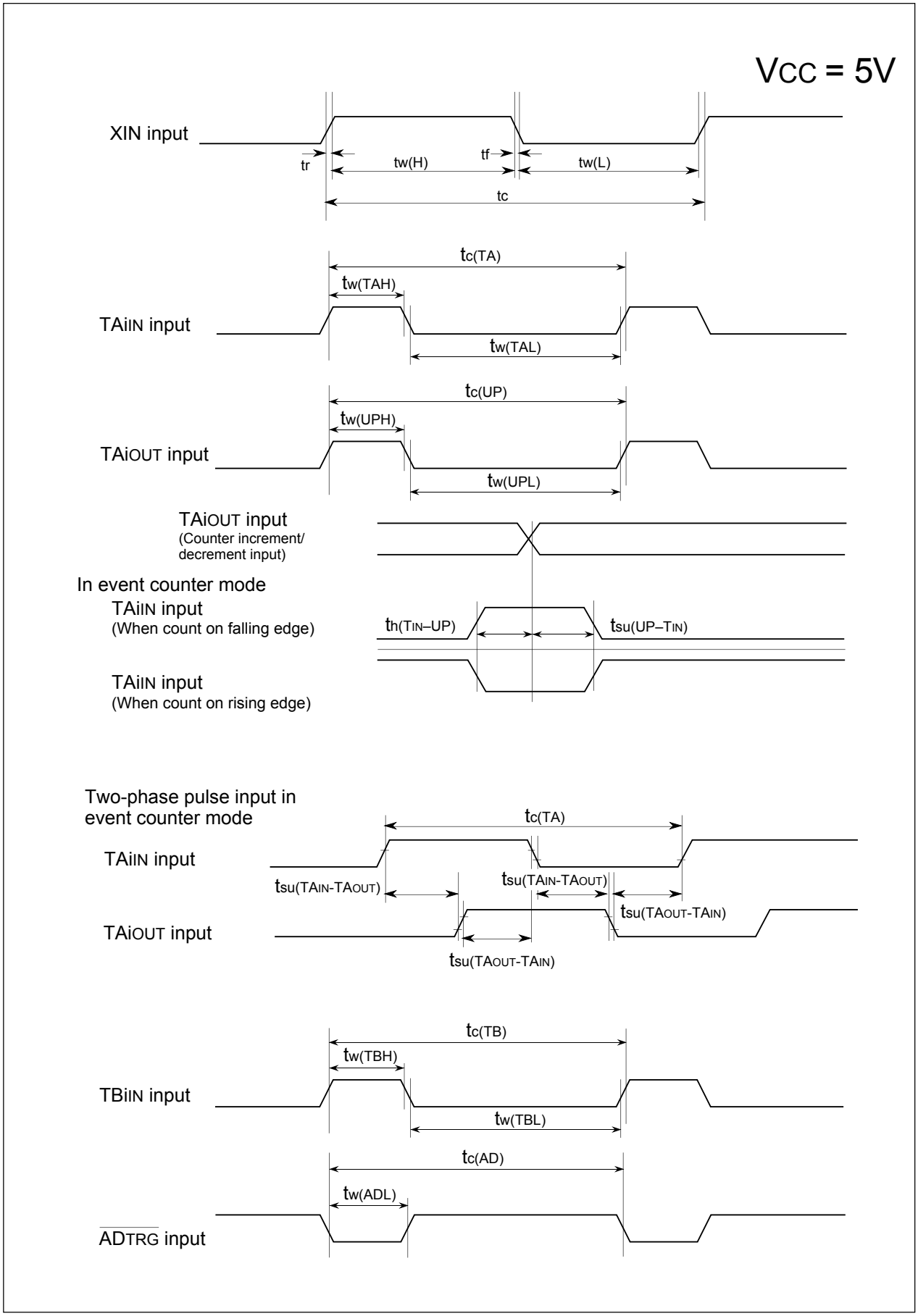


Figure 20.7 Timing Diagram (1)

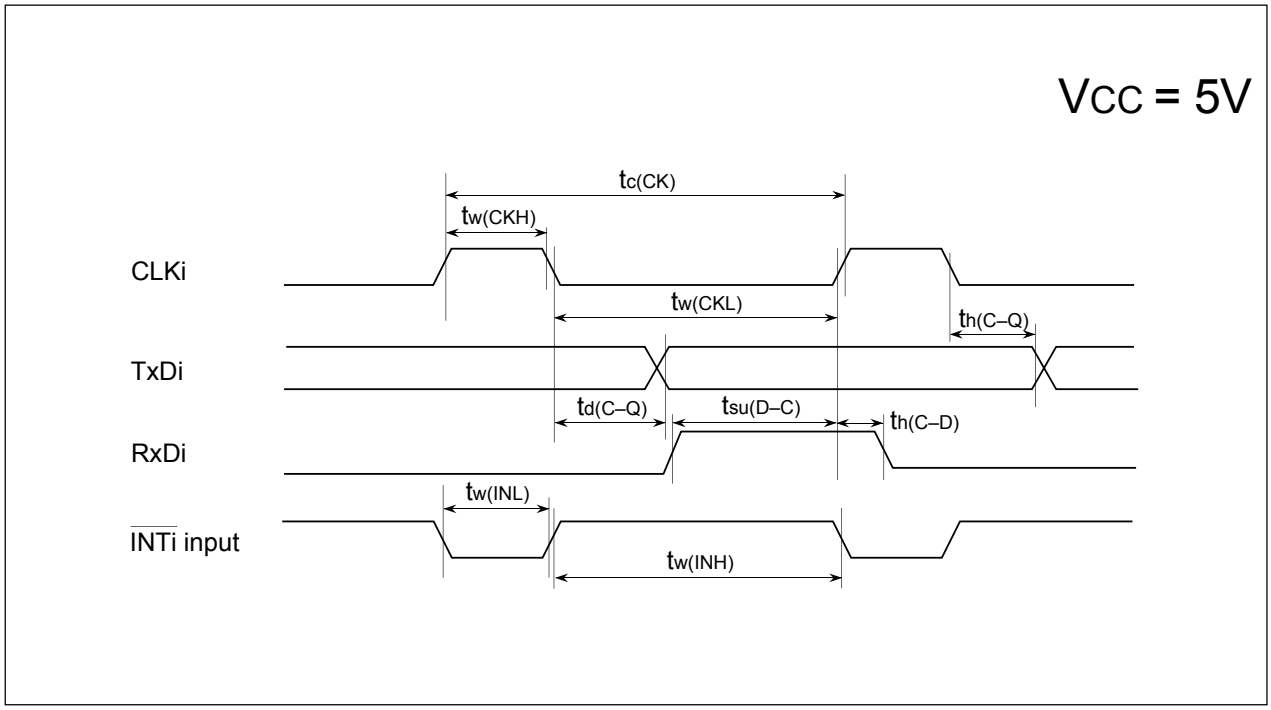


Figure 20.8 Timing Diagram (2)

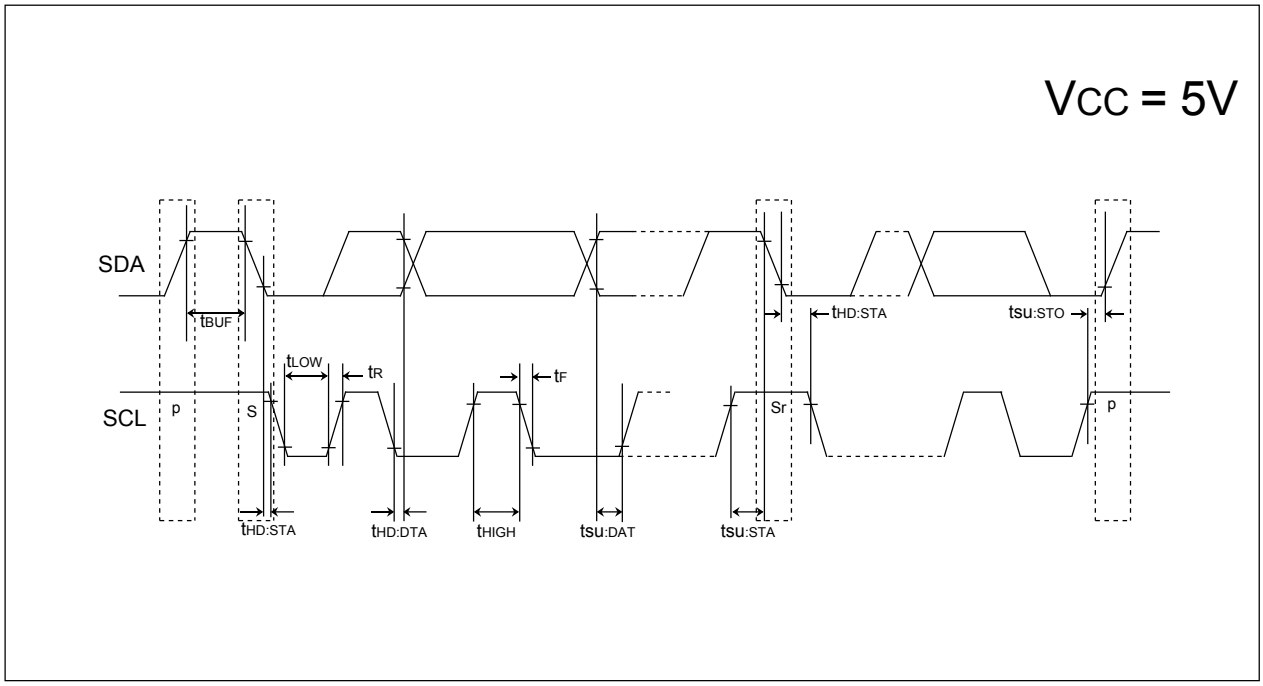


Figure 20.9 Timing Diagram (3)

21. Precautions

21.1 SFR

21.1.1 For 80-Pin Package

Set the IFSR20 bit in the IFSR2A register to 1 after reset and set bits PACR2 to PACR0 in the PACR register to 011_2.

21.1.2 For 64-Pin Package

Set the IFSR20bit in the IFSR2A register to 1 after reset and set bits PACR2 to PACR0 in the PACR register to 010_2.

21.1.3 Register Setting

Immediate values should be set in the registers containing write-only bits. When establishing a new value by modifying a previous value, write the previous value into RAM as well as the register. Change the contents of the RAM and then transfer the new value to the register.

21.2 Clock Generation Circuit

21.2.1 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met.

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
$f_{(ripple)}$	Power supply ripple allowable frequency(V_{CC})				10	kHz
$V_{p-p(ripple)}$	Power supply ripple allowabled amplitude voltage	(V_{CC}=5V)			0.5	V
		(V_{CC}=3V)			0.3	V
$V_{CC(\|DV/DT\|)}$	Power supply ripple rising/falling gradient	(V_{CC}=5V)			0.3	V/ms
		(V_{CC}=3V)			0.3	V/ms

$f_{(ripple)}$
Power supply ripple allowable frequency (V_{CC})

$V_{p-p(ripple)}$
Power supply ripple allowable amplitude voltage

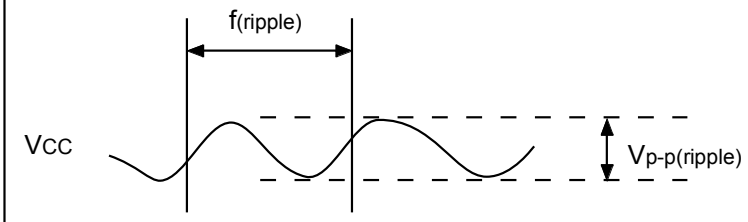


Figure 21.1 Voltage Fluctuation Timing

21.2.2 Power Control

1. When exiting stop mode by hardware reset, the device will startup using the on-chip oscillator.

2. Set the MR0 bit in the TAiMR register(i=0 to 4) to 0 (pulse is not output) to use the timer A to exit stop mode.

3. When entering wait mode, insert a JMP.B instruction before a WAIT instruction. Do not excute any instructions which can generate a write to RAM between the JMP.B and WAIT instructions. Disable the DMA transfers, if a DMA transfer may occur between the JMP.B and WAIT instructions. After the WAIT instruction, insert at least 4 NOP instructions. When entering wait mode, the instruction queue reads ahead the instructions following WAIT, and depending on timing, some of these may execute before the MCU enters wait mode.

Program example when entering wait mode

```
Program Example:    JMP.B   L1   ; Insert JMP.B instruction before WAIT instruction
                L1:
                    FSET    I    ;
                    WAIT         ; Enter wait mode
                    NOP          ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```

4. When entering stop mode, insert a JMP.B instruction immediately after executing an instruction which sets the CM10 bit in the CM1 register to 1, and then insert at least 4 NOP instructions. When entering stop mode, the instruction queue reads ahead the instructions following the instruction which sets the CM10 bit to 1 (all clock stops), and, some of these may execute before the MCU enters stop mode or before the interrupt routine for returning from stop mode.

Program example when entering stop mode

```
Program Example:    FSET    I
                    BSET    CM10    ; Enter stop mode
                    JMP.B   L2      ; Insert JMP.B instruction
                L1:
                    NOP             ; More than 4 NOP instructions
                    NOP
                    NOP
                    NOP
```

5. Wait until the main clock oscillation stabilization time, before switching the CPU clock source to the main clock.
Similarly, wait until the sub clock oscillates stably before switching the CPU clock source to the sub clock.

6. Suggestions to reduce power consumption

(a) Ports

The processor retains the state of each I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A dash current may flow through the input ports in high impedance state, if the input is floating. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

(b) A/D converter

When A/D conversion is not performed, set the VCUT bit in ADiCON1 register to 0 (no Vref connection). When A/D conversion is performed, start the A/D conversion at least 1 μs or longer after setting the VCUT bit to 1 (Vref connection).

(c) Stopping peripheral functions

Use the CM0 register CM02 bit to stop the unnecessary peripheral functions during wait mode.
However, because the peripheral function clock (f_{C32}) generated from the sub-clock does not stop, this measure is not conducive to reducing the power consumption of the chip. If low speed mode or low power dissipation mode is to be changed to wait mode, set the CM02 bit to 0 (do not peripheral function clock stopped when in wait mode), before changing wait mode.

(d) Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

(e)Low Power Consumption Control Register

Follow the procedure below to set the LPCC0 and LPCC1 registers in order to reduce power consumtion.

1) Set the LPCC0 register to 0021_{16}
2) Set the PRC0 bit in the PRCR register to 1
3) Set the LPCC13 bit in the LPCC1 register to 1
4) Set the PRC0 bit to 0

```
Example:  MOV.B    #00100001b, LPCC0    ;
          BSET     PRC0                 ; Write enabled
          MOV.B    #00001000b, LPCC1    ;
          BCLR     PRC0                 ; Write disabled
```

Low-Power Consumption Control Register 0

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

Symbol: LPCC0 | Address: 021016 | After Reset: X00000002

Bit Symbol	Bit Name	Function	RW
LPCC00	Reserved bit	Set to 1	RW
$\overline{(b4 - b1)}$	Reserved bit	Set to 0	RW
LPCC05	Reserved bit	Set to 1	RW
$\overline{(b7 - b6)}$	Reserved bit	Set to 0	RW

Low-Power Consumption Control Register 1(1)

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

Symbol: LPCC1 | Address: 025F16 | After Reset: 0016

Bit Symbol	Bit Name	Function	RW
$\overline{(b2 - b0)}$	Reserved bit	Set to 0	RW
LPCC13	Reserved bit	Set to 1	RW
$\overline{(b7 - b4)}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. Rewrite the LPCC1 register after setting the PRC0 bit in the PRCR register to 1 (wirte enabled).

Figure 21.2 LPCC0 Register and LPCC1 Register

21.3 Protection

Set the PRC2 bit to 1 (write enabled) and then write to any address, and the PRC2 bit will be cleared to 0 (write protected). The registers protected by the PRC2 bit should be changed in the next instruction after setting the PRC2 bit to 1. Make sure no interrupts or DMA transfers will occur between the instruction in which the PRC2 bit is set to 1 and the next instruction.

21.4 Interrupts

21.4.1 Reading Address 00000$_{16}$

Do not read the address 00000$_{16}$ in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000$_{16}$ during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to 0.
If the address 00000$_{16}$ is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to 0. This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

21.4.2 Setting the SP

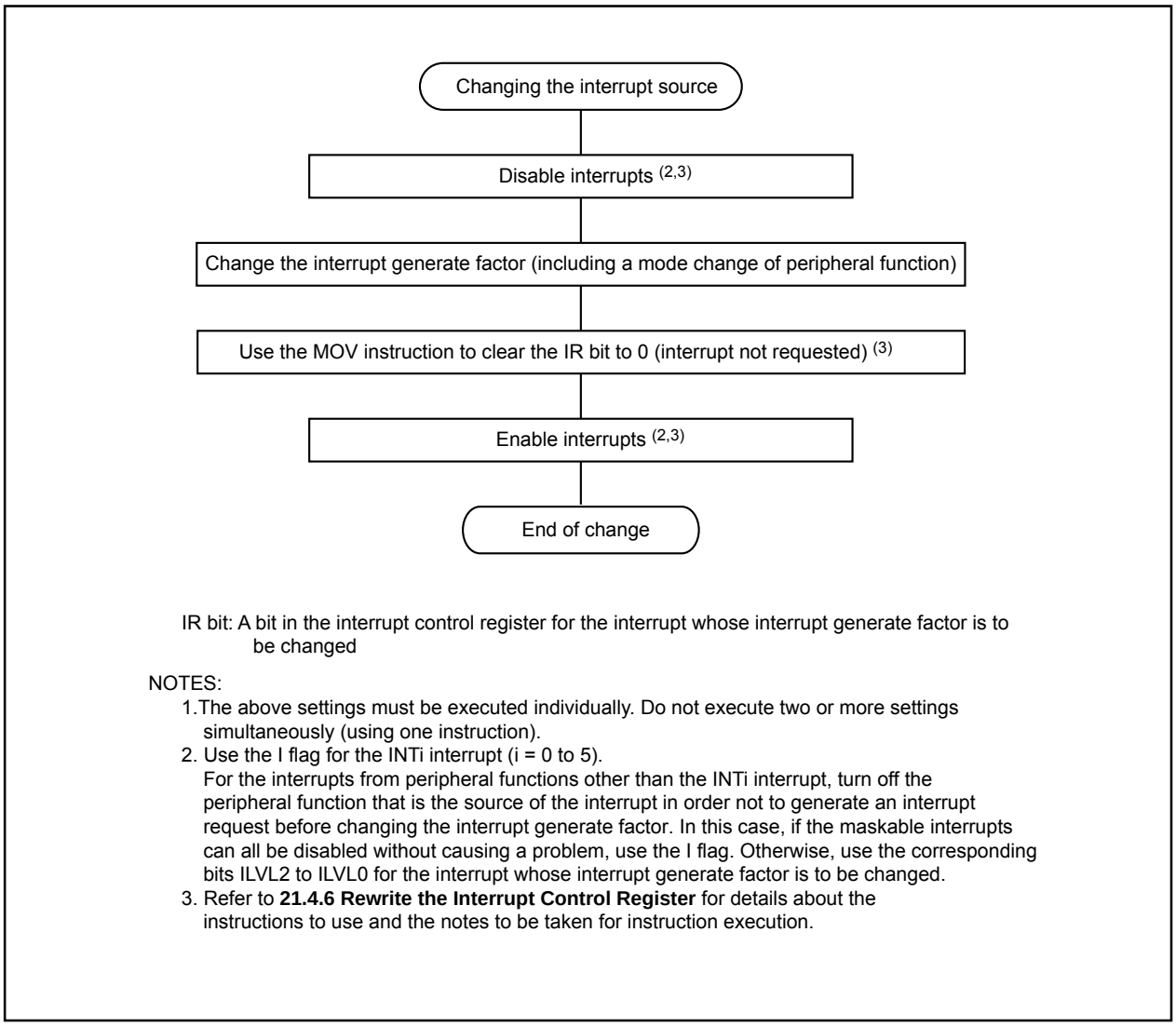
Set any value in the SP(USP, ISP) before accepting an interrupt. The SP(USP, ISP) is cleared to 0000$_{16}$ after reset. Therefore, if an interrupt is accepted before setting any value in the SP(USP, ISP), the program may go out of control.

21.4.3 $\overline{\text{NMI}}$ Interrupt

1. The $\overline{\text{NMI}}$ interrupt is invalid after reset. The $\overline{\text{NMI}}$ interrupt becomes effective by setting the PM24 bit in the PM2 register to “1”. Set the PM24 bit to "1" when a high-level signal ("H") is applied to the $\overline{\text{NMI}}$ pin. If the PM24 bit is set to "1" when a low-level signal ("L") is applied, $\overline{\text{NMI}}$ interrupt is generated. Once $\overline{\text{NMI}}$ interrupt is enabled, it will not be disabled unless a reset is applied.
2. The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8_5 bit in the P8 register.
3. When selecting $\overline{\text{NMI}}$ function, stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM1 register’s CM10 bit is fixed to 0.
4. When selecting $\overline{\text{NMI}}$ function, do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
5. When selecting $\overline{\text{NMI}}$ function, the low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.
6. When using the $\overline{\text{NMI}}$ interrupt for exiting stop mode, set the NDDR register to FF$_{16}$ (disable digital debounce filter) before entering stop mode.

21.4.4 Changing the Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to 0 (interrupt not requested).
“Changing the interrupt generate factor” referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to 0 (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.
Figure 21.3 shows the procedure for changing the interrupt generate factor.



IR bit: A bit in the interrupt control register for the interrupt whose interrupt generate factor is to be changed

NOTES:
1. The above settings must be executed individually. Do not execute two or more settings simultaneously (using one instruction).
2. Use the I flag for the INTi interrupt (i = 0 to 5).
   For the interrupts from peripheral functions other than the INTi interrupt, turn off the peripheral function that is the source of the interrupt in order not to generate an interrupt request before changing the interrupt generate factor. In this case, if the maskable interrupts can all be disabled without causing a problem, use the I flag. Otherwise, use the corresponding bits ILVL2 to ILVL0 for the interrupt whose interrupt generate factor is to be changed.
3. Refer to **21.4.6 Rewrite the Interrupt Control Register** for details about the instructions to use and the notes to be taken for instruction execution.

Figure 21.3 Procedure for Changing the Interrupt Generate Factor

21.4.5 $\overline{\text{INT}}$ Interrupt

1. Either an “L” level of at least $t_{W(INH)}$ or an “H” level of at least $t_{W(INL)}$ width is necessary for the signal input to pins INT_0 through INT_5 regardless of the CPU operation clock.
2. If the POL bit in registers INT0IC to INT5IC or bits IFSR7 to IFSR0 in the IFSR register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to clear the IR bit to 0 (interrupt not requested) after changing any of those register bits.
3. When using the INT_5 interrupt for exiting stop mode, set the P17DDR register to FF_{16} (disable digital debounce filter) before entering stop mode.

21.4.6 Rewrite the Interrupt Control Register

(1) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may occur. Otherwise, disable the interrupt before rewriting the interrupt control register.

(2) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

Changing any bit other than the IR bit

If while executing an instruction, a request for an interrupt controlled by the register being modified occurs, the IR bit in the register may not be set to 1 (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

Changing the IR bit

Depending on the instruction used, the IR bit may not always be cleared to 0 (interrupt not requested). Therefore, be sure to use the MOV instruction to clear the IR bit.

(3) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (2) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to 1 (interrupts enabled) before the interrupt control register is rewrited, due to the internal bus and the instruction queue buffer.

Example 1: Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
      FCLR    I              ; Disable interrupts
      AND.B   #00h, 0055h    ;Set the TA0IC register to 00₁₆
      NOP                    ;
      NOP
      FSET    I              ; Enable interrupts
```

The number of NOP instruction is as follows.
PM20 = 1 (1 wait) : 2, PM20 = 0 (2 waits): 3

Example 2:Using the dummy read to keep the FSET instruction waiting

```
INT_SWITCH2:
      FCLR    I              ; Disable interrupts
      AND.B   #00h, 0055h    ; Set the TA0IC register to 00₁₆
      MOV.W   MEM, R0        ; Dummy read
      FSET    I              ; Enable interrupts
```

Example 3:Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
      PUSHC   FLG
      FCLR    I              ; Disable interrupts
      AND.B   #00h, 0055h    ; Set the TA0IC register to 00₁₆
      POPC    FLG            ; Enable interrupts
```

21.4.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

21.5 DMAC

21.5.1 Write to DMAE Bit in DMiCON Register

When both of the conditions below are met, follow the steps below.

(a) Conditions
- The DMAE bit is set to 1 again while it remains set (DMAi is in an active state).
- A DMA request may occur simultaneously when the DMAE bit is being written.

(b) Procedure
(1) Write 1 to the DMAE bit and DMAS bit in DMiCON register simultaneously(1).
(2) Make sure that the DMAi is in an initial state(2) in a program.

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

NOTES:
1. The DMAS bit remains unchanged even if 1 is written. However, if 0 is written to this bit, it is set to 0 (DMA not requested). In order to prevent the DMAS bit from being modified to 0, 1 should be written to the DMAS bit when 1 is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
   Similarly, when writing to the DMAE bit with a read-modify-write instruction, 1 should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.

2. Read the TCRi register to verify whether the DMAi is in an initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in an initial state. (If a DMA request occurs after writing to the DMAE bit, the value written to the TCRi register is 1.) If the read value is a value in the middle of transfer, the DMAi is not in an initial state.

21.6 Timer

21.6.1 Timer A

21.6.1.1 Timer A (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register and the TAi register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure the TAiMR register is modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always FFFF16. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

21.6.1.2 Timer A (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure bits TAZIE, TA0TGL, and TA0TGH in the TAiMR register, the UDF register, the ONSF register, and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always FFFF16 when the timer counter underflows and 000016 when the timer counter overflows. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

21.6.1.3 Timer A (One-shot Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, bits TA0TGL and TA0TGH in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure bits TA0TGL and TA0TGH in the TAiMR register, the ONSF register, and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. When setting TAiS bit to 0 (count stop), the followings occur:
   • A counter stops counting and a content of reload register is reloaded.
   • TAiOUT pin outputs "L".
   • After one cycle of the CPU clock, the IR bit in TAiIC register is set to 1 (interrupt request).

3. Output in one-shot timer mode synchronizes with a count source internally generated. When the external trigger has been selected, a maximun delay of one cycle of the count source occurs between the trigger input to TAiIN pin and output in one-shot timer mode.

4. The IR bit is set to 1 when timer operation mode is set with any of the following procedures:
   • Select one-shot timer mode after reset.
   • Change an operation mode from timer mode to one-shot timer mode.
   • Change an operation mode from event counter mode to one-shot timer mode.
   To use the timer Ai interrupt (the IR bit), set the IR bit to 0 after the changes listed above have been made.

5. When a trigger occurs while the timer is counting, the counter reloads the reload register value, and continues counting after a second trigger is generated and the counter is decremented once. To generate a trigger while counting, space more than one cycle of the timer count source from the first trigger and generate again.

6. When selecting the external trigger for the count start conditions in timer A one-shot timer mode, do not generate an external trigger 300ns before the count value of timer A is set to 0000_{16}. The one-shot timer may stop counting.

7. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

21.6.1.4 Timer A (Pulse Width Modulation Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using bits TA0TGL and TA0TGH in the TAiMR (i = 0 to 4) register, the TAi register, the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure bits TA0TGL and TA0TGH in the TAiMR register, the ONSF register and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. The IR bit is set to 1 when setting a timer operation mode with any of the following procedures:
   • Select the PWM mode after reset.
   • Change an operation mode from timer mode to PWM mode.
   • Change an operation mode from event counter mode to PWM mode.
   To use the timer Ai interrupt (interrupt request bit), set the IR bit to 0 by program after the above listed changes have been made.

3. When setting TAiS register to 0 (count stop) during PWM pulse output, the following action occurs:
   • Stop counting.
   • When TAiOUT pin is output “H”, output level is set to “L” and the IR bit is set to 1.
   • When TAiOUT pin is output “L”, both output level and the IR bit remains unchanged.

4. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

21.6.2 Timer B

21.6.2.1 Timer B (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR register to 1 (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always FFFF16. If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.

21.6.2.2 Timer B (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TBiMR (i = 0 to 2) register and TBi register before setting the TBiS bit in the TABSR register to 1 (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not.

2. The counter value can be read out at any time by reading the TBi register. However, if this register is read at the same time the counter is reloaded, the read value is always FFFF16. If the TBi register is read after setting a value in it but before the counter starts counting, the read value is the one that has been set in the register.

21.6.2.3 Timer B (Pulse Period/pulse Width Measurement Mode)

1. The timer remains idle after reset. Set the mode, count source, etc. using the TBiMR (i = 0 to 2) register before setting the TBiS bit in the TABSR or the TBSR register to 1 (count starts).
   Always make sure the TBiMR register is modified while the TBiS bit remains 0 (count stops) regardless whether after reset or not. To clear the MR3 bit to 0 by writing to the TBiMR register while the TBiS bit is set to 1 (count starts), be sure to write the same value as previously written to bits TM0D0, TM0D1, MR0, MR1, TCK0, and TCK1 and a 0 to the MR2 bit.

2. The IR bit in TBiIC register (i=0 to 2) goes to 1 (interrupt request), when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the MR3 bit in TBiMR register within the interrupt routine.

3. If the source of interrupt cannot be identified by the MR3 bit such as when the measurement pulse input and a timer overflow occur at the same time, use another timer to count the number of times timer B has overflowed.

4. To set the MR3 bit to 0 (no overflow), set TBiMR register with setting the TBiS bit to 1 and counting the next count source after setting the MR3 bit to 1 (overflow).

5. Use the IR bit in TBiIC register to detect only overflows. Use the MR3 bit only to determine the interrupt factor within the interrupt routine.

6. When a count is started and the first effective edge is input, an undefined value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

7. A value of the counter is undefined at the beginning of a count. MR3 may be set to 1 and timer Bi interrupt request may be generated between a count start and an effective edge input.

8. For pulse width measurement, pulse widths are successively measured. Use program to check whether the measurement result is an “H” level width or an “L” level width.

21.6.3 Three-phase Motor Control Timer Function

When the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forced cutoff by SD pin input (high-impedance) enabled), the INV03 bit in the INVC0 register is set to 1 (three-phase motor control timer output enabled), and a low-level ("L") signal is applied to the $\overline{SD}$ pin while a three-phase PWM signal is output, the MCU is forced to cutoff and pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ are placed in a high-impedance state and the INV03 bit is set to 0 (three-phase motor control timer output disabled).
To resume the three-phase PWM signal output from pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$, set the INV03 bit to 1 and the IVPCR1 bit to 0 (three-phase output forced cutoff disabled) after the $\overline{SD}$ pin level becomes "H". Then set the IVPCR1 bit to 1 (three-phase output forced cutoff enabled) in order to enable the three-phase output forced cutoff function by input to the SD pin again.
The INV03 bit cannot be set to 1 while an "L" signal is input to the $\overline{SD}$ pin. To set the INV03 bit to 1 after forcible cutoff, write 1 to the INV03 bit and read the bit to ensure that it is set to 1 by program. Then set the IVPCR1 bit to 1 after setting it to 0.

21.7 Timer S

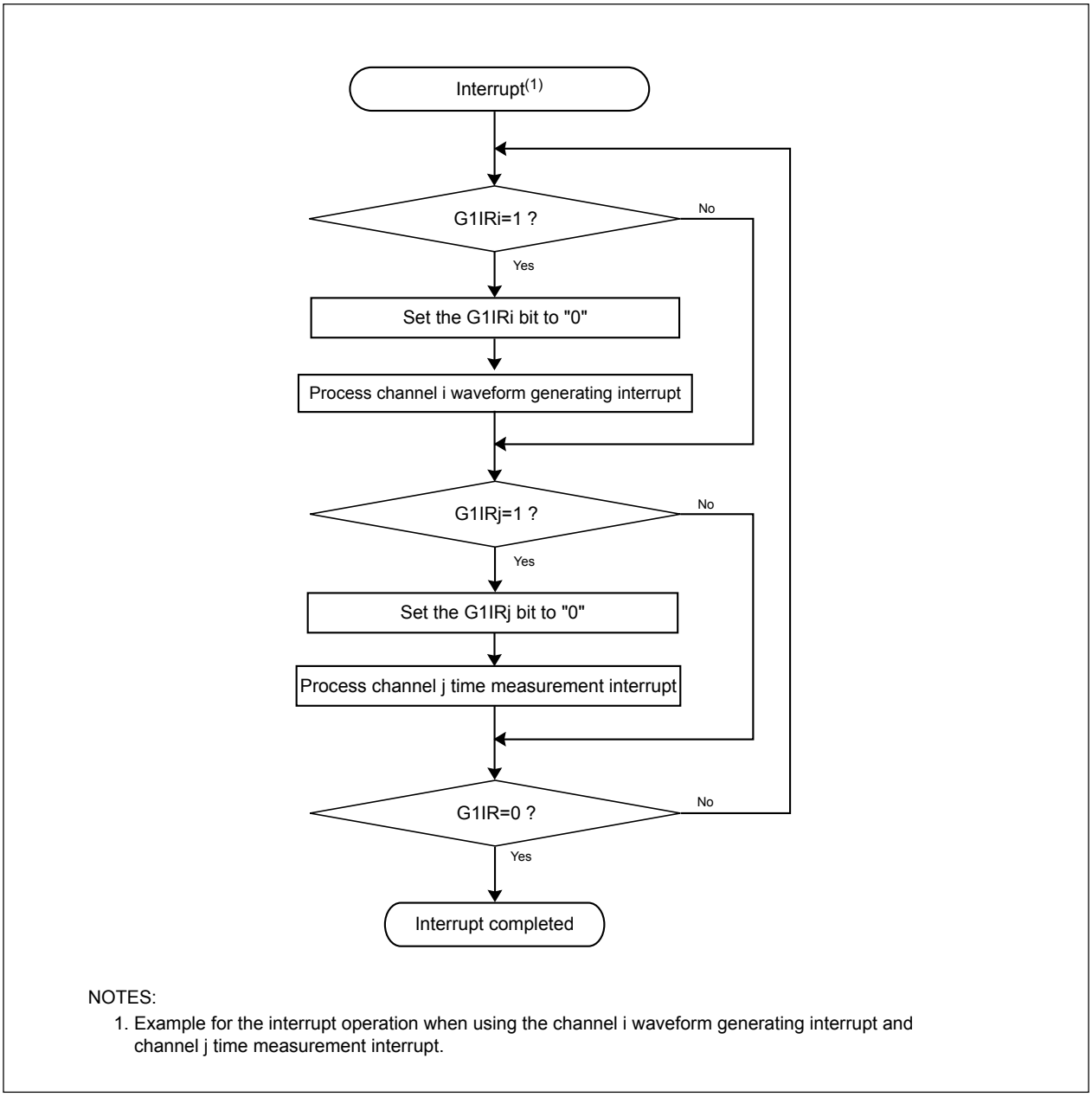
21.7.1 Rewrite the G1IR Register

Bits in the G1IR register are not automatically set to 0 (no interrupt requested) even if a requested interrupt is acknowledged. Set each bit to 0 by program after the interrupt requests are verified.

The IC/OC interrupt is generated when any bit in the G1IR register is set to 1 (interrupt requested) after all the bits are set to 0. If conditions to generate an interrupt are met when the G1IR register holds the value other than 00_{16}, the IC/OC interrupt request will not be generated. In order to enable an IC/OC interrupt request again, clear the G1IR register to 00_{16}. Use the following instructions to set each bit in the G1IR register to 0.

Subject instructions: AND, BCLR

Figure 21.4 shows an example of IC/OC interrupt i processing.



NOTES:
1. Example for the interrupt operation when using the channel i waveform generating interrupt and channel j time measurement interrupt.

Figure 21.4 IC/OC Interrupt i Flow Chart

21.7.2 Rewrite the ICOCiIC Register

When the interrupt request to the ICOCiIC register is generated during the instruction process, the IR bit may not be set to 1 (interrupt requested) and the interrupt request may not be acknowledged. At that time, when the bit in the G1IR register is held to 1 (interrupt requested), the following IC/OC interrupt request will not be generated. When changing the ICOCiIC register settiing, use the following instruction.

Subject instructions: AND, OR, BCLR, BSET

When initializing Timer S, change the ICOCiIC register setting with the request again after setting the IOCiIC and G1IR registers to 00_{16}.

21.7.3 Waveform Generating Function

1. If the BTS bit in the G1BCR1 register is set to 0 (base timer is reset) when the waveform is generating and the base timer is stopped counting, the waveform output pin keeps the same output level. The output level will be changed when the base timer and the G1POj register match the setting value next time after the base timer starts counting again.
2. If the G1POCRj register is set when the waveform is generated, the same setting value of the IVL bit is applied to the waveform generating pin. Do not set the G1POCRj register when the waveform is generating.
3. When the RST1 bit in the G1BCR1 register is set to 1 (the base timer is reset by matching the G1PO0 register), the base timer is reset after two clock cycles of f_{BT1} when the base timer value matches the G1PO0 register value. A high-level ("H") signal is applied to the OUTC10 pin between the base timer value match to the base timer reset.

21.7.4 IC/OC Base Timer Interrupt

If the MCU is operated in the combination selected from Tabl e 1 for use when the RST4 bit in the G1BCR0 register is set to 1 (reset the base timer that matches the G1BTRR register) to reset the base timer, an IC/OC base timer interrupt request is generated twice.

Table 21.1 Uses of IT Bit in the G1BCR0 Register and G1BTRR Register

IT Bit in the G1BCR0 Register	G1BTRR Register
0 (bit 15 in the base timer overflows)	$07FFF_{16}$ to $0FFFE_{16}$
1 (bit 14 in the base timer overflows)	$03FFF_{16}$ to $0FFFE_{16}$ or $0BFFF_{16}$ to $0FFFE_{16}$

The second IC/OC base timer interrupt request is generated because the base timer overflow request is generated after one fBT1 clock cycle as soon as the base timer is reset.

One of the following conditions must be met in order not to generate the IC/OC base timer interrupt request twice:

1) When the RST4 bit is set to 1, set the G1BTRR register with a combination other than what is listed in **Table 21.1**.
2) Do not reset the base timer by matching the G1BTRR register. Reset the base timer by matching the G1P00 register. In other words, do not set the RST4 bit to 1 to reset the base timer. Set the RST1 bit in the G1BCR1 register to 1 (reset the base timer that matches the G1P00 register).

21.8 Serial I/O

21.8.1 Clock-Synchronous Serial I/O

21.8.1.1 Transmission/reception

1. With an external clock selected, and choosing the $\overline{RTS}$ function, the output level of the $\overline{RTSi}$ pin goes to “L” when the data-receivable status becomes ready, which informs the transmission side that the reception has become ready. The output level of the $\overline{RTSi}$ pin goes to “H” when reception starts. So if the $\overline{RTSi}$ pin is connected to the $\overline{CTSi}$ pin on the transmission side, the circuit can transmission and reception data with consistent timing. With the internal clock, the $\overline{RTS}$ function has no effect.

2. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the $P7_3/RTS_2/TxD1$(when the U1MAP bit in PACR register is 1) and CLK_2 pins go to a high-impedance state.

21.8.1.2 Transmission

When an external clock is selected, the conditions must be met while if the CKPOL bit in the UiC0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock), the external clock is in the high state; if the CKPOL bit in the UiC0 register is set to 1 (transmit data output at the rising edge and the receive data taken in at the falling edge of the transfer clock), the external clock is in the low state.

- The TE bit in UiC1 register is set to 1 (transmission enabled)
- The TI bit in UiC1 register is set to 0 (data present in UiTB register)
- If $\overline{CTS}$ function is selected, input on the $\overline{CTSi}$ pin is set to “L”

21.8.1.3 Reception

1. In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxDi pin when receiving data.

2. When an internal clock is selected, set the TE bit in the UiC1 register (i = 0 to 2) to 1 (transmission enabled) and write dummy data to the UiTB register, and the shift clock will thereby be generated. When an external clock is selected, set the TE bit in the UiC1 register (i = 0 to 2) to 1 and write dummy data to the UiTB register, and the shift clock will be generated when the external clock is fed to the CLKi input pin.

3. When successively receiving data, if all bits of the next receive data are prepared in the UARTi receive register while the RE bit in the UiC1 register (i = 0 to 2) is set to 1 (data present in the UiRB register), an overrun error occurs and the UiRB register OER bit is set to 1 (overrun error occurred). In this case, because the content of the UiRB register is undefined, a corrective measure must be taken by programs on the transmit and receive sides so that the valid data before the overrun error occurred will be retransmitted. Note that when an overrun error occurred, the SiRIC register IR bit does not change state.

4. To receive data in succession, set dummy data in the lower-order byte of the UiTB register every time reception is made.

5. When an external clock is selected, make sure the external clock is in high state if the CKPOL bit is set to 0, and in low state if the CKPOL bit is set to 1 before the following conditions are met:
   - The RE bit in the UiC1 register is set to 1 (reception enabled)
   - The TE bit in the UiC1 register is set to 1 (transmission enabled)
   - The TI bit in the UiC1 register= 0 (data present in the UiTB register)

21.8.2 UART Mode

21.8.2.1 Special Mode 1 (I^2C bus Mode)

When generating start, stop and restart conditions, set the STSPSEL bit in the U2SMR4 register to 0 and wait for more than half cycle of the transfer clock before setting each condition generate bit (STAREQ, RSTAREQ and STPREQ) from 0 to 1.

21.8.2.2 Special Mode 2

If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the RTS_2 and CLK_2 pins go to a high-impedance state.

21.8.2.3 Special Mode 4 (SIM Mode)

A transmit interrupt request is generated by setting the U2C1 register U2IRS bit to 1 (transmission complete) and U2ERE bit to 1 (error signal output) after reset. Therefore, when using SIM mode, be sure to clear the IR bit to 0 (no interrupt request) after setting these bits.

21.8.3 SI/O3, SI/O4

The SOUTi default value which is set to the SOUTi pin by the SMi7 bit approximately 10ns may be output when changing the SMi3 bit from 0 (I/O port) to 1 (SOUTi output and CLKfunction) while the SMi2 bit in the SiC (i=3 and 4) to 0 (SOUTi output) and the SMi6 bit is set to 1 (internal clock). And then the SOUTi pin is held high-impedance.

If the level which is output from the SOUTi pin is a problem when changing the SMi3 bit from 0 to 1, set the default value of the SOUTi pin by the SMi7 bit.

21.9 A/D Converter

1. Set registers ADCON0 (except bit 6), ADCON1, ADCON2 and ADTRGCON when A/D conversion is stopped (before a trigger occurs).

2. When the VCUT bit in ADCON1 register is changed from 0 (Vref not connected) to 1 (Vref connected), start A/D conversion after passing 1 µs or longer.

3. To prevent noise-induced device malfunction or latchup, as well as to reduce conversion errors, insert capacitors between the AVCC, VREF, and analog input pins (ANi, AN0i, AN2i(i=0 to 7), and AN3i(i=0 to 2)) each and the AVSS pin. Similarly, insert a capacitor between the VCC1 pin and the VSS pin. **Figure 21.5** is an example connection of each pin.

4. Make sure the port direction bits for those pins that are used as analog inputs are set to 0 (input mode). Also, if the TGR bit in the ADCON0 register is set to 1 (external trigger), make sure the port direction bit for the $\overline{\text{ADTRG}}$ pin is set to 0 (input mode).

5. When using key input interrupts, do not use any of the four AN4 to AN7 pins as analog inputs. (A key input interrupt request is generated when the A/D input voltage goes low.)

6. The ϕAD frequency must be 10 MHz or less. Without sample-and-hold function, limit the ϕAD frequency to 250kHz or more. With the sample and hold function, limit the ϕAD frequency to 1MHz or more.

7. When changing an A/D operation mode, select analog input pin again in bits CH2 to CH0 in the ADCON0 register and bits SCAN1 to SCAN0 in the ADCON1 register.

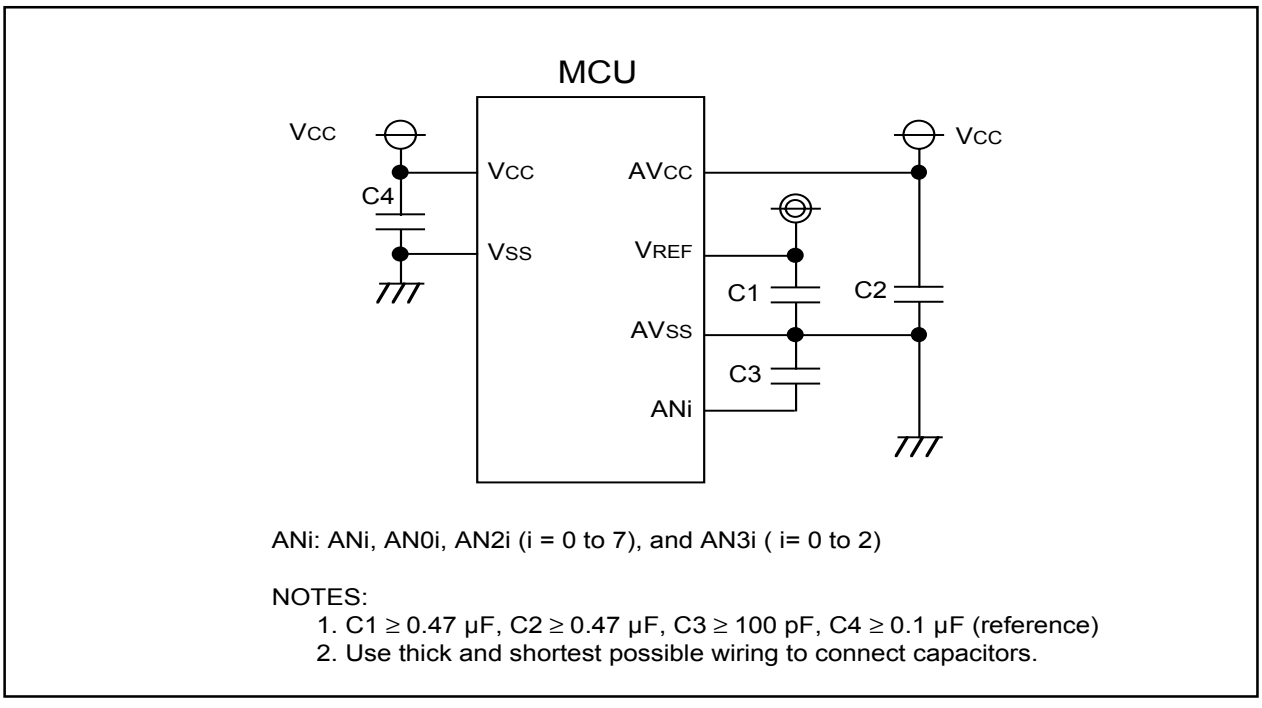


Figure 21.5 Use of capacitors to reduce noise

8. If the CPU reads the ADi register (i = 0 to 7) at the same time the conversion result is stored in the ADi register after completion of A/D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.
   - When operating in one-shot, single-sweep mode, simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1:
     Check to see that A/D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register to see if A/D conversion is completed.)
   - When operating in repeat mode or repeat sweep mode 0 or 1:
     Use the main clock for CPU clock directly without dividing it.

9. If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to 0 (A/D conversion halted), the conversion result of the A/D converter is undefined. The contents of ADi registers irrelevant to A/D conversion may also become undefined. If while A/D conversion is underway the ADST bit is cleared to 0 in a program, ignore the values of all ADi registers.

10. When setting the ADST bit in the ADCON register to 0 and terminating forcefully by a program in single sweep conversion mode, A/D delayed trigger mode 0 and A/D delayed trigger mode 1 during A/D converting operation, the A/D interrupt request may be generated. If this causes a problem, set the ADST bit to 0 after an interrupt is disabled.

21.10 Multi-master I^2C bus Interface

21.10.1 Writing to the S00 Register

When the start condition is not generated, the SCL pin may output the short low-signal ("L") by setting the S00 register. Set the register when the SCL pin outputs an "L" signal.

21.10.2 AL Flag

When the arbitration lost is generated and the AL flag in the S10 register is set to 1 (detected), the AL flag can be cleared to 0 (not detected) by writing a transmit data to the S00 register. The AL flag should be cleared at the timing when master geneates the start condition to start a new transfer.

21.11 Programmable I/O Ports

1. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the $P7_2$ to $P7_5$, $P8_0$ and $P8_1$ pins go to a high-impedance state.

2. The input threshold voltage of pins differs between programmable input/output ports and peripheral functions.
   Therefore, if any pin is shared by a programmable input/output port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions V_{IH} and V_{IL} (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable input/output port or the peripheral function—is currently selected.

3. When the SM32 bit in the S3C register is set to 1, the $P3_2$ pin goes to high-impedance state. When the SM42 bit in the S4C register is set to 1, the $P9_6$ pin goes to high-imepdance state.

4. When the INV03 bit in the INVC0 register is 1(three-phase motor control timer output enabled), an "L" input on the $P8_5$ /$\overline{NMI}$/$\overline{SD}$ pin, has the following effect.

   - When the TB2SC register IVPCR1 bit is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a high-impedance state.
   - When the TB2SC register IVPCR1 bit is set to 0 (three-phase output forcible cutoff by input on $\overline{SD}$ pin disabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a normal port.

   Therefore, the $P8_5$ pin can not be used as programmable I/O port when the INV03 bit is set to 1.
   When the $\overline{SD}$ function isn't used, set to 0 (Input) in $PD8_5$ and pullup to H in the $P8_5$ /$\overline{NMI}$/$\overline{SD}$ pin from outside.

21.12 Electric Characteristic Differences Between Mask ROM and Flash Memory Version

Flash memory version and mask ROM version may have different characteristics, operating margin, noise tolerated dose, noise width dose in electrical characteristics due to internal ROM, different layout pattern, etc. When switching to the mask ROM version, conduct equivalent tests as system evaluation tests conducted in the flush memory version.

21.13 Mask ROM Version

21.13.1 Internal ROM Area

In the masked ROM version, do not write to internal ROM area. Writing to the area may increase power consumption.

21.13.2 Reserved Bit

The b3 to b0 in addresses $0FFFFF_{16}$ are reserved bits. Set these bits to 1111_2.

21.14 Flash Memory Version

21.14.1 Functions to Inhibit Rewriting Flash Memory Rewrite

ID codes are stored in addresses 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, and 0FFFFB16. If wrong data are written to theses addresses, the flash memory cannot be read or written in standard serial I/O mode.

The ROMCP register is mapped in address 0FFFFF16. If wrong data is written to this address, the flash memory cannot be read or written in parallel I/O mode.

In the flash memory version of MCU, these addresses are allocated to the vector addresses ("H") of fixed vectors. The b3 to b0 in address 0FFFFF16 are reserved bits. Set these bits to 11112.

21.14.2 Stop Mode

When the MCU enters stop mode, execute the instruction which sets the CM10 bit to 1 (stop mode) after setting the FMR01 bit to 0 (CPU rewrite mode disabled) and disabling the DMA transfer.

21.14.3 Wait Mode

When the MCU enters wait mode, excute the WAIT instruction after setting the FMR01 bit to 0 (CPU rewrite mode disabled).

21.14.4 Low Power Dissipation Mode, On-Chip Oscillator Low Power Dissipation Mode

If the CM05 bit is set to 1 (main clock stop), the following commands must not be executed.

- Program
- Block erase

21.14.5 Writing Command and Data

Write the command code and data at even addresses.

21.14.6 Program Command

Write xx4016 in the first bus cycle and write data to the write address in the second bus cycle, and an auto program operation (data program and verify) will start. Make sure the address value specified in the first bus cycle is the same even address as the write address specified in the second bus cycle.

21.14.7 Operation Speed

When CPU clock source is main clock, before entering CPU rewrite mode (EW mode 0 or 1), select 10 MHz or less for BCLK using the CM06 bit in the CM0 register and bits CM17 to CM16 in the CM1 register.

Also, when CPU clock is f3(ROC) on-chip oscillator clock, before entering CPU rewrite mode (EW mode 0 or 1), set the ROCR3 to ROCR2 bits in the ROCR register to “divied by 4” or “divide by 8”.

On both cases, set the PM17 bit in the PM1 register to 1 (with wait state).

21.14.8 Instructions Inhibited Against Use

The following instructions cannot be used in EW mode 0 because the flash memory’s internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

21.14.9 Interrupts

EW Mode 0

- Any interrupt which has a vector in the variable vector table can be used providing that its vector is transferred into the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.
  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

EW Mode 1

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program period or auto erase period with erase-suspend function disabled.
- The $\overline{\text{NMI}}$ interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.
  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

21.14.10 How to Access

To set the FMR01, FMR02, FMR11 or FMR16 bit to 1, set the subject bit to 1 immediately after setting to 0. Do not generate an interrupt or a DMA transfer between the instruction to set the bit to 0 and the instruction to set the bit to 1. Set the bit when the PM24 bit is set to 1 ($\overline{\text{NMI}}$ funciton) and a high-level ("H") signal is applied to the $\overline{\text{NMI}}$ pin.

21.14.11 Writing in the User ROM Area

EW Mode 0

- If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

EW Mode 1

- Avoid rewriting any block in which the rewrite control program is stored.

21.14.12 DMA Transfer

In EW mode 1, make sure that no DMA transfers will occur while the FMR00 bit in the FMR0 register is set to 0(during the auto program or auto erase period).

21.14.13 Regarding Programming/Erasure Times and Execution Time

As the number of programming/erasure times increases, so does the execution time for software commands (program command and block erase command).
The software commands are aborted by hardware reset 1, $\overline{\text{NMI}}$ interrupt, and watchdog timer interrupt. If a software command is aborted by such reset or interrupt, the affected block must be erased before reexecuting the aborted command.

21.14.14 Definition of Programming/Erasure Times

"Number of programs and erasure" refers to the number of erasure per block.
If the number of program and erasure is n (n=100 1,000 10,000) each block can be erased n times.
For example, if a 2K byte block A is erased after writing 1 word data 1024 times, each to a different address, this is counted as one program and erasure. However, data cannot be written to the same adrress more than once without erasing the block. (Rewrite prohibited)

21.14.15 Flash Memory Version Electrical Characteristics 10,000 E/W cycle product (U7)

When Block A or B E/W cycles exceed 100, set the FMR17 bit in the FMR1 register to 1 (1 wait) to select one wait state per block access for U7. When FMR17 is set to 1, one wait state is inserted per access to Block A or B - regardless of the value of PM17. Wait state insertion during access to all other blocks, as well as to internal RAM, is controlled by PM17 - regardless of the FMR17 bit setting.
To use the limited number of erasure efficiently, write to unused address within the block instead of rewite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times before erase becomes necessary.
Maintaining an equal number of erasure between Block A and B will also improve efficiency.
We recommend keeping track of the number of times erasure is used.

21.14.16 Boot Mode

An undefined value is sometimes output in the I/O port until the internal power supply becomes stable when "H" is applied to the CNVSS pin and "L" is applied to the $\overline{\text{RESET}}$ pin.
When setting the CNVSS pin to "H", the following procedure is required:

(1) Apply an "L" signal to the $\overline{\text{RESET}}$ pin and the CNVSS pin.
(2) Bring VCC to more than 2.7V, and wait at least 2msec. (Internal power supply stable waiting time)
(3) Apply an "H" signal to the CNVSS pin.
(4) Apply an "H" signal to the $\overline{\text{RESET}}$ pin.

When the CNVSS pin is “H” and RESET pin is “L”, P67 pin is connected to the pull-up resister.

21.15 Noise

Connect a bypass capacitor (approximately 0.1μF) across the VCC and VSS pins using the shortest and thicker possible wiring. **Figure 21.6** shows the bypass capacitor connection.

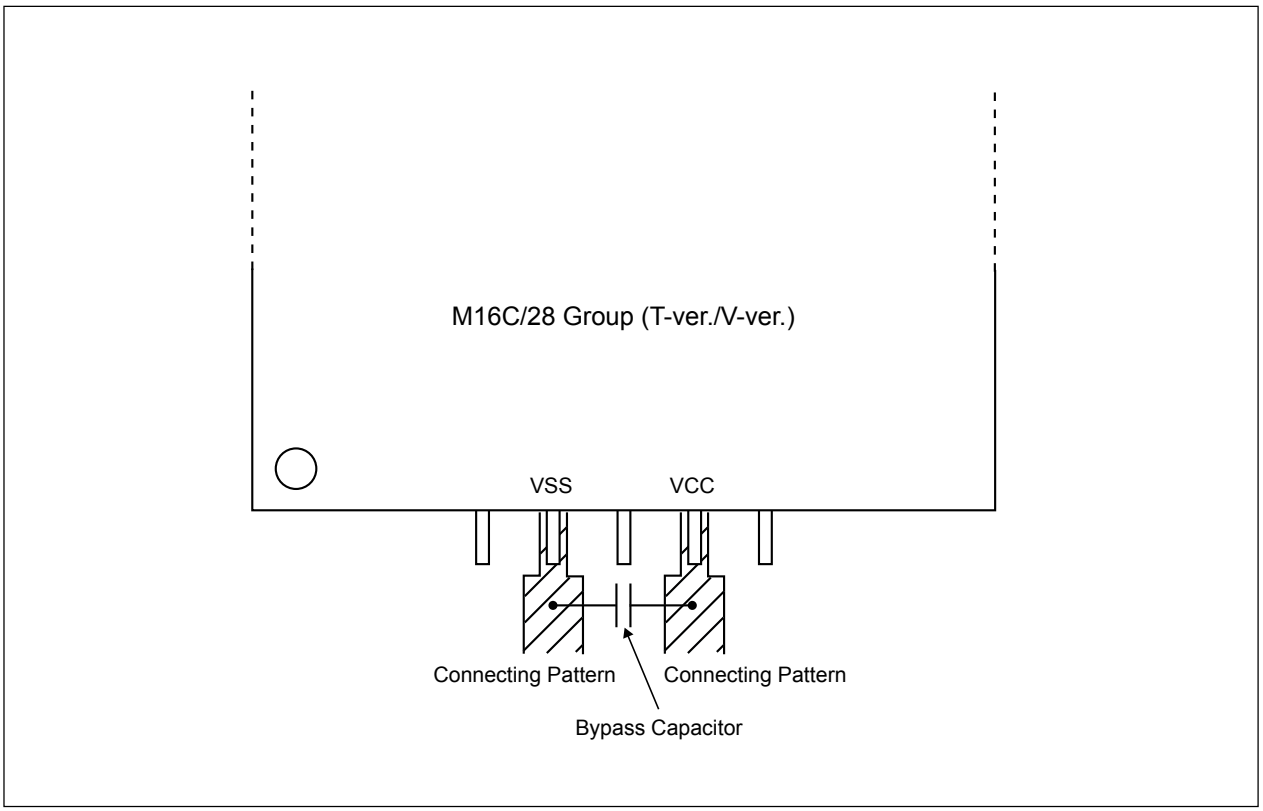


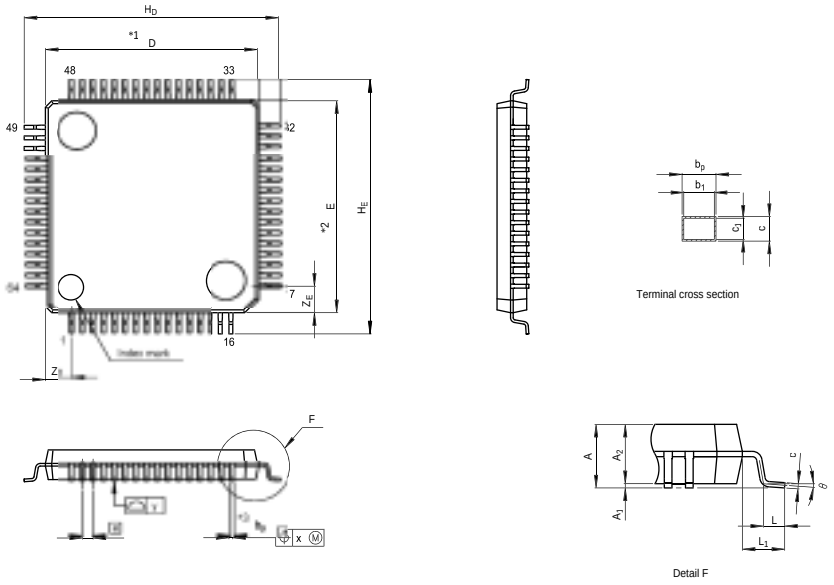
Figure 21.6 Bypass Capacitor Connection

21.16 Instruction for a Device Use

When handling a device, extra attention is necessary to prevent it from crashing during the electrostatic discharge period.

Appendix 1. Package Dimensions

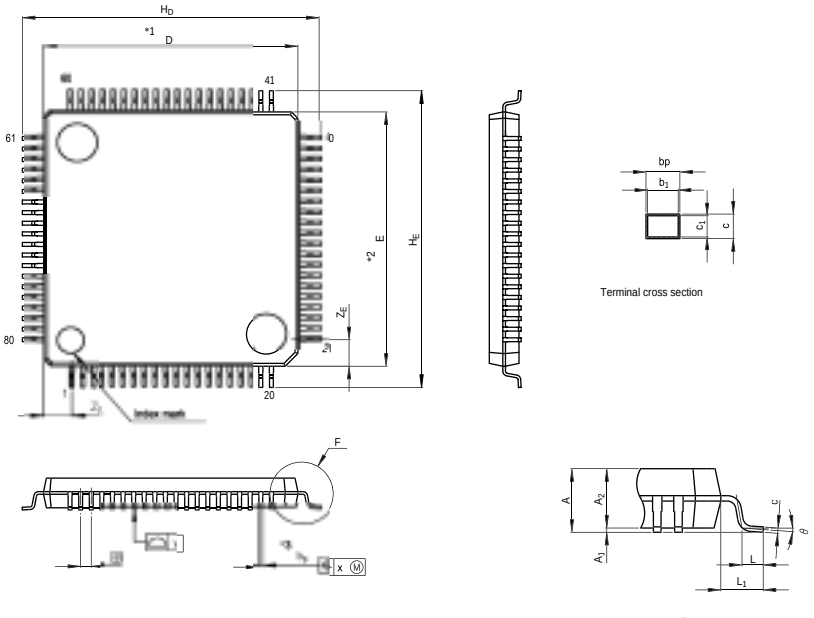
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP64-10x10-0.50	PLQP0064KB-A	64P6Q-A / FP-64K / FP-64KV	0.3g



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	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
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	—	1.25	—
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-LQFP80-12x12-0.50	PLQP0080KB-A	80P6Q-A	0.5g



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	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0	0.1	0.2
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°	—	10°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.3	0.5	0.7
L_1	—	1.0	—

Appendix 2. Functional Comparison

Appendix 2.1 Difference between M16C/28 Group Normal-ver. and M16C/28 Group T-/V-ver. (1)

Item	Description	M16C/28(Normal-ver.)	M16C/28(T-ver./V-ver.)
Clock Generation Circuit	Clock output function (function of b1 to b0 bits in the CM0 register)	Not available (reserved bit)	Available (clock output function select bit)
Reset	Low Voltage Detect Circuit (function of 0019_{16}, $001A_{16}$, $001F_{16}$)	Available (voltage detect register 1, voltage detect register 2, low voltage detect interrupt register)	Not available (reserved register)
Three-phase Motor Control Timer	Three-phase port switching function (function of 0358_{16})	Not available (reserved register)	Available (port function select register)
A/D	Number of A/D input pin	24 channels (excluding $AN3_0$ to $AN3_2$)	27 channels (including $AN3_0$ to $AN3_2$)
	Delayed trigger mode 0	Not available in the 1st chip version and chip version A	Available
	Delayed trigger mode 1	Not available in the 1st chip version and chip version A	Available
CRC Calculation	Available (compatible to CRC-CCITT and CRC-16 methods)	Not available (all related registers are reserved registers)	Available (1 circuit)
Pin Function	3 pins (80-pin/85-pin package), 64 pins (64-pin package.)	$P9_2$/$TB2_{IN}$	$P9_2$/$AN3_2$/$TB2_{IN}$
	4 pins (80-pin package), 1 pin (64-pin package)	$P9_1$/$TB1_{IN}$	$P9_1$/$AN3_1$/$TB1_{IN}$
	5 pins (80-pin package), 2 pins (64-pin package)	$P9_0$/$TB0_{IN}$	$P9_0$/$AN3_0$/$TB0_{IN}$/CLK_{OUT}
Flash Memory	$P9_3$ in standard serial I/O mode	I (other than 128 Kbyte version) I/O (128 Kbyte version)	I/O

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

NOTE:
1. Since the M16C/28 group uses the common emulator used in the M16C/29 group, all the functions are available for M16C/28. When evaluating M16C/28 group, do not access to the SFR which is not built-in the M16C/28 gorup. Refere to hardware manual for details and electrical characteristics.

Appendix 2.2 Difference between M16C/28 Group T-/V-ver. and M16C/29 Group T-/V-ver. (1)

Item	Description	M16C/28(T-ver./V-ver.)	M16C/29(T-ver./V-ver.)
Protection	Function of the PRC0 bit	Enable to set the CM0, CM1, CM2, POCR, PLC0 and PCLKR registers	Enable to set the CM0, CM1, CM2, POCR, PLC0, PCLKR and CCLKR registers
Interrupt	The IFSR20 bit setting in the IFSR2A register	Set to 1	Set to 0
	The b1 bit in the IFSR2A register	Not available (reserved bit)	Interrupt cause switching bit (0: A/D conversion, 1:key input)
	The b2 bit in the IFSR2A register	Not available (reserved bit)	Interrupt cause switching bit (0: CAN0 wake-up/ error)
	Interrupt cause in the Interrupt number 13	Key input interrupt	CAN0 error
	Interrupt cause in the Interrupt number 14	Key input interrupt	A/D, key input interrupt
CAN module	compatible to 2.0B	Not available (all related registers are reserved registers)	Available (1 channel)
Pin Function	2 pins (80-pin/85-pin package), 62 pins (64-pin package)	$P9_3$/$AN2_4$	$P9_3$/$AN2_4$/CTX
	3 pins (80-pin/85-pin package), 64 pins (64-pin package)	$P9_2$/$TB2_{IN}$	$P9_2$/$AN3_2$/$TB2_{IN}$/CRX
Flash Memory	$P9_3$ in standard serial I/O mode	I/O	CTX output

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

NOTE:
1. Since the M16C/28 group uses the common emulator used in the M16C/29 group, all the functions are available for M16C/28. When evaluating M16C/28 group, do not access to the SFR which is not built-in the M16C/28 gorup. Refere to hardware manual for details and electrical characteristics.

Register Index

REVISION HISTORY	M16C/28 Group(T-ver./V-ver.) Hardware Manual

Rev.	Date	Description	
		Page	Summary
1.00	**Dec. 05**		New Document
1.10	**Mar.30,07**	All pages	Word standardized: MCU, CPU clock
			Overview
		1	• **1.1 Features** description modified
		2, 3	• **Tables 1.1 and 1.2 Performance Overviews** note on trademark modified
		4, 5	• **Figures 1.1 and 1.2 M16C/28 Group (T-ver./ V-ver.) Block Diagrams** Notes are added
		6	• **Tables 1.3 and 1.4 Product Lists** updated
		7	• **Figure 1.3 Product Numbering System** updated
		8	• **Tables 1.5 and 1.6 Product Code** modified
			• **Tables 1.7 and 1.8 Product Code** mask version newly added
		10	• **Figure 1.6 Marking Diagram of Mask ROM version** is newly added
		12, 13	• **Table 1.9 Pin Characteristics for 80-pin Package** is newly added
		15, 16	• **Table 1.10 Pin Characteristics for 64-pin Package** is newly added
		17 to 19	• **Table 1.11 to 1.12 Pin Description** is modified
			Memory
		22	• **Figure 3.1 Memory Map** Internal RAM memory size is modified
			SFR
		25	• **Table 4.3 SFR Information (3)** Registers LPCC0 and LPCC1 are added, values after reset for ROCR register and FMR4 register are modified
		27	• **Table 4.5 SFR Information (5)** Value after reset for IFSR2A register is modified
			Reset
		32	• **Figure 5.2 Reset Sequence** Vcc and ROC timing lines modified
			Processor Mode
		33	• Description added
			• **Figure 6.1 Bus Block Diagram** added
		34	• **Figure 6.2 PM2 Register** added
		35	• **Figure 6.3 Bus Block Diagram** and **Table 6.1 Accessible area and bus cycles** newly added
			Clock Generation Circuits
		36	• **Table 7.1 Clock Generation Circuit Specifications** "Oscillation stop, restart function" is modified
		37	• **Figure 7.1 Clock Generation Circuits** Partially modified
		41	• **Figure 7.6 PM2 Register** Notes 4 to 5 partially modified
		48	• **7.6.1 Normal Operation Mode** Description is modified
		52	• **Figure 7.11 State Transition to Stop Mode and Wait Mode** Figure partially modified
		53	• **Figure 7.12 State Transition in Normal Mode** Figure partially modified
		54	• **Table 7.7 Allowed Transition and Setting** Table contents partially modified

Rev.	Date	Description	
		Page	Summary
		55	• **7.8 Osillation Stop and Re-Oscillation Detect Function** Description modified
			Protection
		58	• Description partially modified, LPCC1 register added
			• **Figure 8.1 PRCR Register** PRC0 bit is modified
			Interrupts
		76	•**Table 9.6 Value of the PC that is saved to the stack area when an address match interrupt request is accepted** modified, note added
			Watchdog Timer
		78	• Description partially added
			• **Figure 10.1 Watchdog Timer Block Diagram** Partially modified
		79	• **Figure 10.2 WDTS Register** note deleted
			Timer
		106	• **12.2 Timer B** Description of A/D trigger mode modified
			• **Figure 12.15 Timer B Block Diagram** A/D trigger mode added
		112	• **12.2.4 A/D Trigger Mode** Description modified
		118	• **Figure 12.28 ICTB2 Register** bits 7 and 6 modified
		120	• **Figure 12.30 TB2SC Register** Note 4 is added, contents modified
		123	• **Figure 12.33 Triangular Wave Modulation Operation** Description modified
		124	• **Figure 12.34 Sawtooth Wave Modulation Operation** Description modified
		128	• **Figure 12.38 TPRC Register** bit map modified
			Timer S
		131	• **Figure 13.2 G1BT Register** Description patially modified, **G1BCR0 Register** Bit name partially modified
		144	• **Figure 13.15 Base Timer Reset Operation by Base Timer Reset Register** note 1 is added, figures partially modified
		149	• **Figure 13.21 Prescaler Function and Gate Function** Note 1 modified
		155	• **Table 13.10 SR Waveform Output Mode Specifications** Specification modified
		157	• **Table 13.11 Pin Setting for Time Measurement and Waveform Generating Functions** port direction modified
			Serial I/O
		160	• **Figure 14.1 Block Diagram of UARTi** Figure modified
		169	• **Table 14.1 Clock Synchronous Serial I/O Mode Specifications** Note 2 modified
		177	• **Table 14.5 UART Mode Specifications** Note 1 modified
		185	• **Table 14.10 I^2C bus Mode Specifications** Note 2 modified
		195	• **Table 14.15 Special Mode 2 Specifications** Note 2 modifed
		201	• **Table 14.18 SIM Mode Specifications** Note 1 modified
		206	• **14.2 SI/O3 and SI/O4** Note is added

Rev.	Date	Description	
		Page	Summary
		207	• **Figure 14.36 S3BRG and S4BRG Registers** register name modified
		210	• **14.2.3 Functions for Setting an SOUTi Initial Value** Description modified
			Multi-master I^2C bus Interface
		245	• **Figure 16.1 Block Diagram of Multi-master I^2C bus Interface** S30 register deleted, system clock select circuit partially modified
		247	• **Figure 16.3 S00 Register** Bit name in note 1 modified
		268	• **16.11 STOP Condition Generation Method** Description added
			CRC Calculation Circuit
		276	• **17.1 CRC Snoop** Description partially modified
			Programmable I/O Ports
		281	• **Figure 18.1 I/O Ports (1)** Partially modified
		283	• **Figure 18.3 I/O Ports (3)** Partially modified
		291	• **Figure 18.12 Functioning of Digital Debounce Filter** Figure partially modified
			Flash Memory Version
		295	• **19.2 Memory Map** description modified
			• **Figure 19.1 Flash Memory Block Diagram (ROM capacity 64K)** added
		297	• **19.3.1 ROM Code Protect Function** Description is modified
		301	• **19.5.1 Flash Memory Control Register 0 FMR01 Bit and FMR02 Bit**: descriptions are modified
		302	• **19.5.2 Flash Memory Control Register 1 FMR16 Bit**: description is modified
		303	• **Figure 19.5 FMR1 Register** Note 1 and note 3 modified
		314	• **Table 19.7 Errors and FMR0 Register Status** Register name modified
			Electrical Characteristics
		325	• **Table 20.3 A/D Conversion Characteristics** Unit for t_{CONV} is modified
		326	• **Table 20.5 Flash Memory Version Electrical Characteristics for 10000 E/W cycle products** Note 1, 4, 10, and 11 are modified
		327	• td(P-R) and td(ROC) timing lines modified
		329	• **Table 20.8 Electrical Characteristics** Table is modified
		336	• **Table 20.23 Electrical Characteristics** Note 1 is modified
		337	• **Table 20.24 Electrical Characteristics** Table is modified
		346	• **Table 20.41 A/D Conversion Characteristics** modified
		347	• **Table 20.42 and 20.43 Flash Memory Version Electrical Characteristics** Note 4, 10, and 11 are modified
		348	• td(P-R) and td(ROC) timing lines modified
		350	• **Table 20.46 Electrical Characteristics** Table is modified, note 4 is added
			Precaution
		-	• **Reset** section deleted
		357	• **21.1.3 Register Setting** newly added
		360, 361	• **(e)Low Power Consumption Control Register** newly added

Rev.	Date	Description	
		Page	Summary
		365	• **21.4.6 Rewrite the Interrupt Control Register** Example 1 modified
		371	• **21.6.3 Three-phase Motor Control Timer Function** newly added
		372	• **21.7.1 Rewrite the G1IR Register** Description modified
		373	• **21.7.4 IC/OC Base Timer Interrupt** Section newly added
		381	• **21.13.1 Internal ROM Area** partially added
		383	• **21.14.9 Interrupts EW Mode 1** Description about watchdog timer interrupt deleted • **21.14.10 How to Access** partially deleted • **21.14.13 Regarding Programming/Erasure Times and Execution Time** Description partially modified
		-	**Functional Comparison** • **Difference between M16C/28 Group and M16C/29 Group (Normal-ver.)** is deleted
		388	• **Appendix 2.1 Difference between M16C/28 Group Normal-ver. and m16C/28 Group T-ver./V-ver.** flash memory added

RENESAS 16-BIT SINGLE-CHIP MICROCOMPUTER
HARDWARE MANUAL
M16C/28 Group (T-ver./V-ver.)

Publication Data : Rev.1.00 Dec. 26, 2005
Rev.1.10 Mar. 30, 2007

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

M16C/28 Group (T-ver./V-ver.)
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0287-0110

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Microprocessors - MPU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MC68302EH20C MC7457RX1000LC MC7457RX1267LC MC7457VG1267LC A2C00010998 A A2C52004004 R5F117BCGNA#20 R5F52106BDLA#U0 R5S72690W266BG#U0 ADJ3400IAA5DOE MPC8245TVV266D MPC8245TZU300D MPC8260ACVVMHBB MPC8323ECVRAFDCA MPC8323VRADDCA MPC8536ECVJAVLA BOXNUC5PGYH0AJ 20-668-0024 P1010NSN5DFB P2010NSE2KHC P2010NSN2MHC P2020NXE2HHC P5020NSE7QMB P5020NSE7TNB P5020NSE7VNB LS1020ASN7KQB LS1020AXN7HNB LS1020AXN7KQB A2C00010729 A A2C00039344 T1022NSE7MQB T1022NXN7PQB T1023NSE7MQA T1024NXE7PQA T1042NSE7MQB T1042NSN7MQB T1042NXN7WQB T2080NSE8TTB T2080NSN8PTB T2080NXE8TTB T2081NXN8TTB R5F101AFASP#V0 MC68302CEH20C TS68040MF33A MPC8260ACVVMIBB MPC8280CZUUPEA MPC8313ECVRAFFC MPC8313ECVRAGDC MPC8313EVRADDC MPC8313EVRAFFC