REU09B0001-0090Z

16

M16C/26 Group

Hardware Manual

RENESAS 16-BIT CMOS SINGLE-CHIP MICROCOMPUTER

M16C FAMILY / M16C/20 SERIES

Hardware Manual

Before using this material, please visit the our website to confirm that this is the most current document available.

Rev. 0.90
Revision date: Sep. 1. 2003

RenesasTechnology
www.renesas.com

Keep safety first in your circuit designs!

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

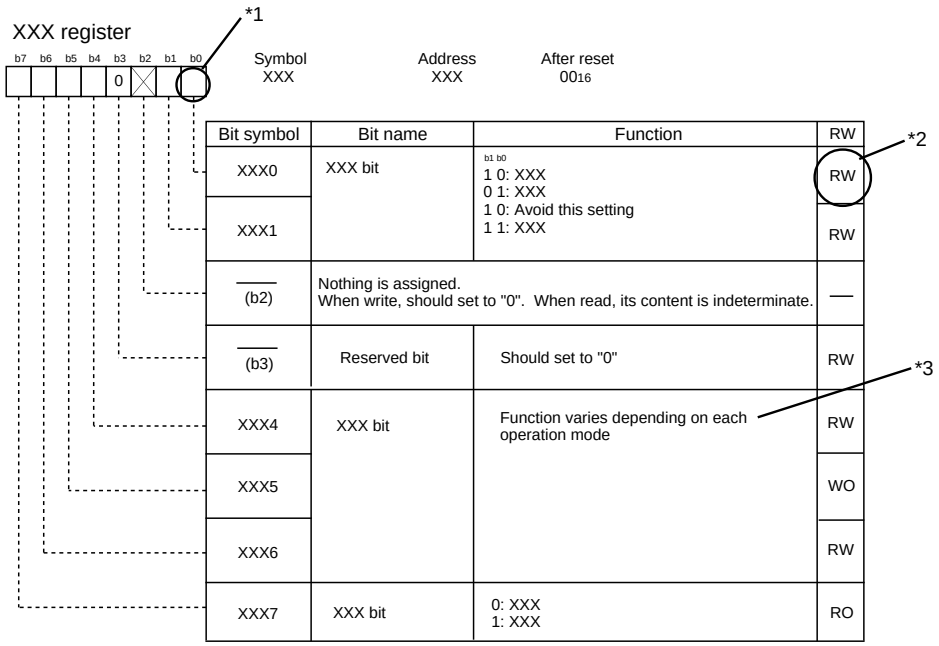
Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
- Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
  The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
  Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation for further details on these materials or t he products contained therein.

How to Use This Manual

This hardware manual provides detailed information on features in the M16C/26 Group microcomputer.
Users are expected to have basic knowledge of electric circuits, logical circuits and microcomputer.

Each register diagram contains bit functions with the following symbols and descriptions.



*1

Blank: Set to "0" or "1" according to your intended use
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

Terms to use here are explained as follows.

• Nothing is assigned
Nothing is assigned to the bit concerned. When write, set to "0" for new function in future plan.

• Reserved bit
Reserved bit. Set the specified value.

• Avoid this setting
The operation at having selected is not guaranteed.

• Function varies depending on each operation mode
Bit function varies depending on peripheral function mode.
Refer to register diagrams in each mode.

M16C Family Documents

Document	Contents
Short Sheet	Hardware overview
Data Sheet	Hardware overview and electrical characteristics
Hardware Manual	Hardware specifications (pin assignments, memory maps, specifications of peripheral functions, electrical characteristics, timing charts)
Software Manual	Detailed description about instructions and microcomputer performance by each instruction
Application Note	• Application examples of peripheral functions • Sample programs • Introductory description about basic functions in M16C family • Programming method with the assembly and C languages

Table of Contents

Description

The M16C/26 group of single-chip microcomputers incorporates Renesas' 16-bit M16C/60 Series CPU core, fabricated in a high-performance silicon gate CMOS process and packaged in 48-pin plastic molded LQFPs. Noted for fast instruction processing, these microcomputers are designed to operate using a sophisticated, yet highly-efficient instruction set. Their design is optimized for general application to the home, office, audio, and industrial sectors.

The M16C/26 group comprises a range of products of varied configuration. Note that the term "Virtual EEPROM" is used in this manual to denote Flash ROM blocks capable of a very high number of erase/write cycles. Simulated byte or word erase/write (E/W) requires firmware (available from Renesas).

Features

- Memory ..... Flash ROM 24K to 64K bytes (1K minimum erase/write cycles)
  plus Flash ROM 2K bytes x 2 blocks as Virtual EEPROM
  (10K minimum erase/write cycles, 100K typical erase/write cycles)
  Erase/write voltage for flash is from Vcc = 2.7 to 5.5 V
  RAM 1K to 2K bytes.
- Shortest Instruction Execution Time ..... 50ns (f(XIN)=20 MHz, Vcc=3.0V to 5.5V)
  100ns (f(Xin)=10 MHz, Vcc=2.7V to 5.5V)
- Supply Voltage ..... Vcc = 3.0V to 5.5V (f(Xin) = 20MHz, no wait)
  Vcc = 2.7V to 5.5V (f(Xin) = 10MHz, no wait)
- Low Power Consumption ..... 0.7uA (typ) Icc, Stop mode, Vcc = 3.0V,
  1.8 uA (typ) Icc at 32kHz Wait mode, Vcc = 3.0V
  25uA (typ) Icc at 32KHz mode with RAM access (Flash sleep)
  16mA (typ) Icc at 20 MHz, Vcc = 5.0V
- Interrupts ..... 20 internal and 7 external interrupt sources
  4 software interrupt sources; 7 priority levels
- Serial I/O ..... 2 U(S)ART/SIO (UART 0/1)
  1 U(S)ART/SIO/I^2C bus/IEBus (UART 2) (Note 1,2)
- DMAC ..... 2 Channels
- A-D Converter ..... 10 bits X 8 channels
- Watchdog Timer ..... 1 15-bit timer (switchable to internal ring oscillator)
- Timers ..... 8 16-bit timers
  - 5 Timer A (Event counter, Event timer, One-shot generator, Pulse width measurement generation, Quadrature counter)
  - 3 Timer B (Event counter, Period measurement, Pulse width measurement, Timer)
  - 3-phase PWM generators for motor control
- Key-on Wake Up ..... 4 inputs
- Programmable I/O ..... 38 lines
- Brown-out Monitor ..... With interrupt, reset and RAM retention flag capability.
- Clock Generation Circuit ..... 3 built-in clock generation circuits
  - 2 internal clock generators (supports ceramic or quartz oscillator); crystal failure detect
  - 1 internal ring oscillator

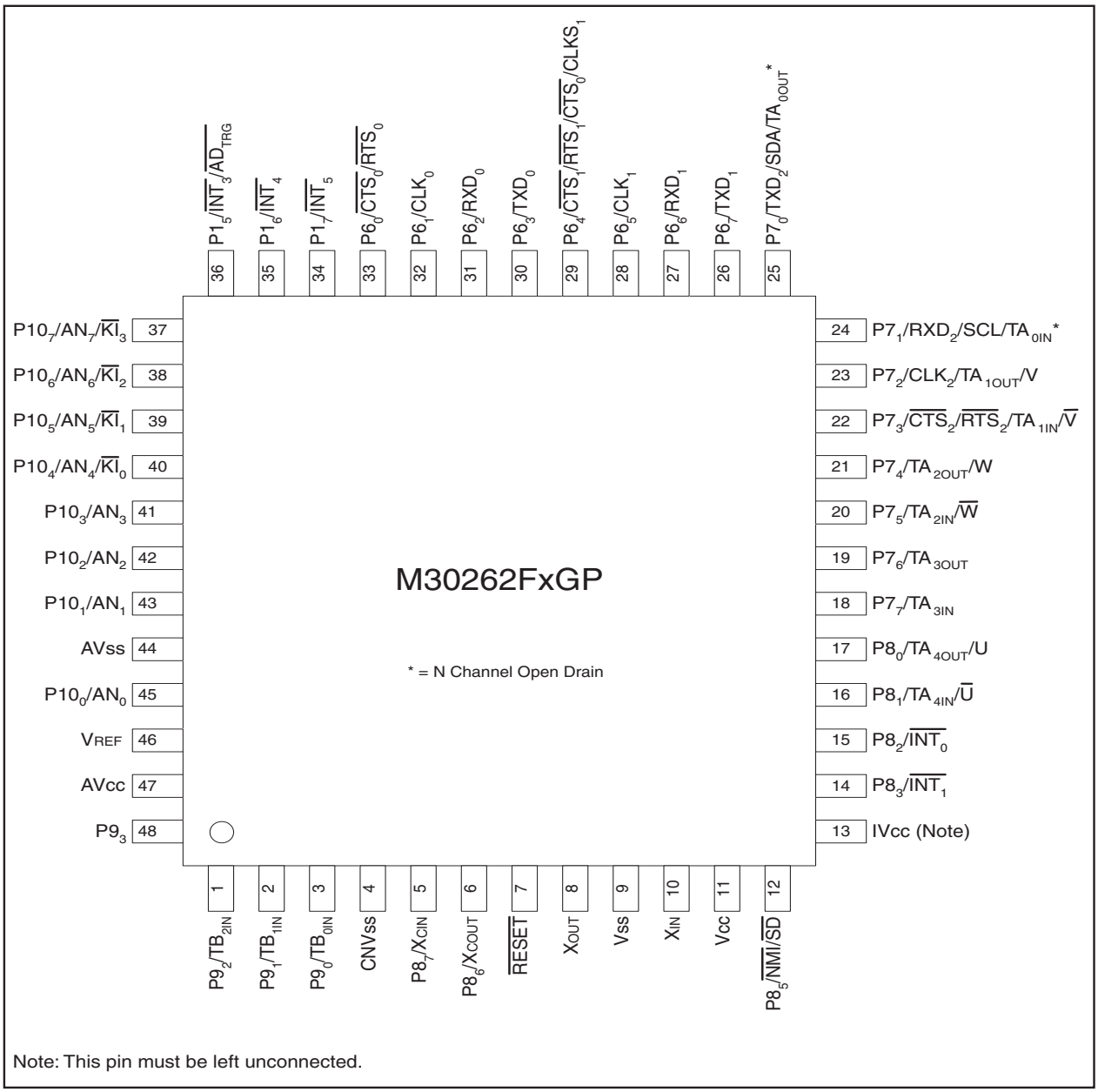
Note 1: I^2C bus is a registered trademark of Koninklijke Philips Electronics N. V.
Note 2: IEBus is a registered trademark of NEC Electronics Corporation.

Applications

Industrial/audio/office equipment, home appliances, and other portable equipment.

Pin Configuration

Figure 1.1 shows the pin configuration for the M16C/26 group. Table 1.2 lists the pin descriptions.



Note: This pin must be left unconnected.

Figure 1.1. Pin configuration diagram (top view) of the M16C/26 group (48P6Q package)

Block Diagram

Figure 1.2 shows a block diagram of the M16C/26 group device.

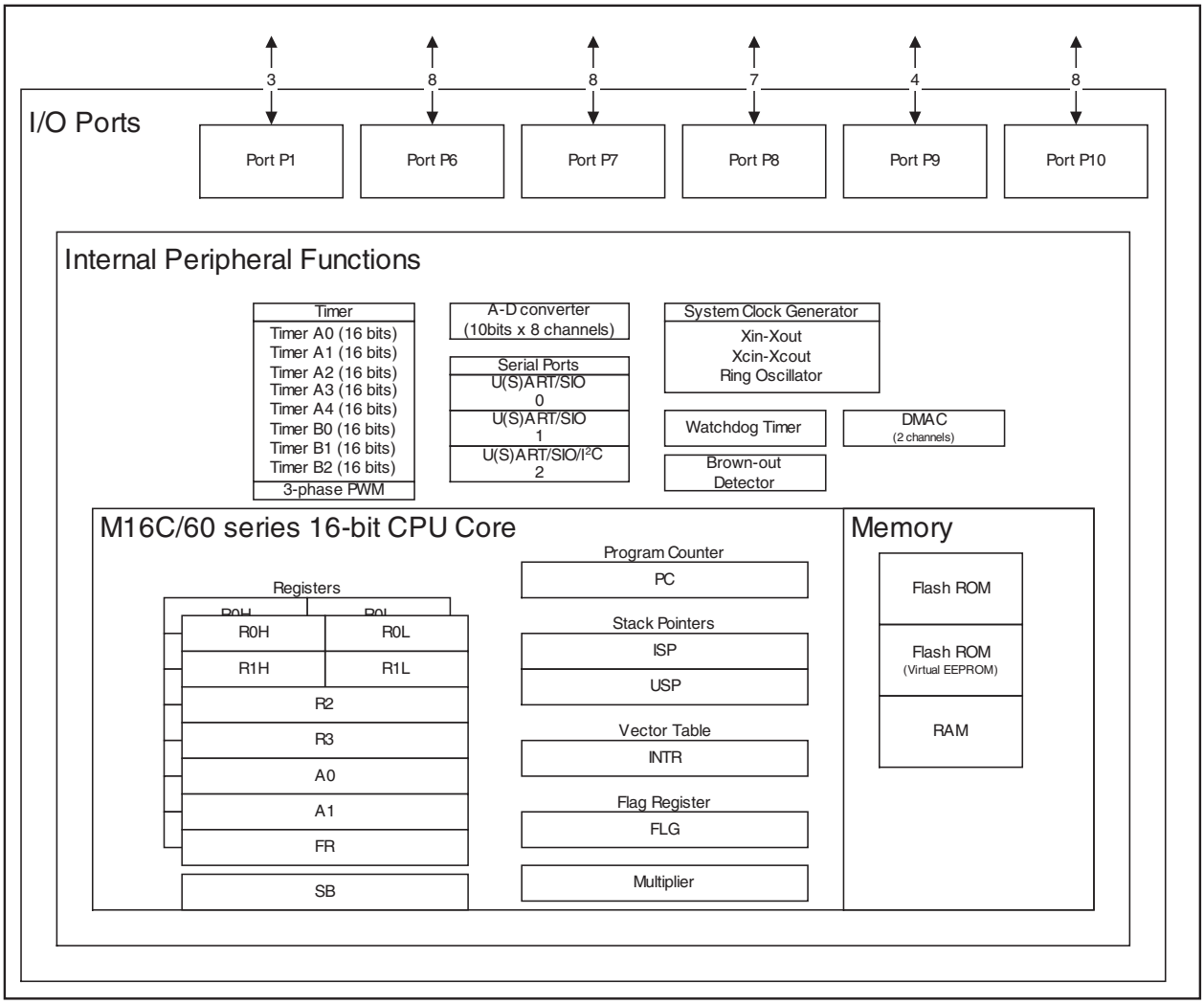


Figure 1.2. M16C/26 group block diagram

Performance Outline

Table 1.1 lists the parameters and functional description of the M16C/26 group.

Table 1.1. Performance outline

Parameters		Functional Description
Number of Basic Instructions		91
Shortest Instruction Execution Time		50ns (f(XIN)=20 MHz, Vcc=3.0V to 5.5V) 100ns (f(XIN)=10 MHz, Vcc=2.7V to 5.5V)
Memory	Flash ROM (Note 3)	(see the ROM expansion diagram) 1K (min) erase/write cycles, Erase/write Vcc = 2.7V to 5.5V
	Flash ROM as Virtual EEPROM (Note 3)	4 Kbytes (2 Kbyte x 2 Blocks) 10K minimum erase/write, 100K typical erase/write cycles Erase/write Vcc = 2.7V to 5.5V
	RAM	1K to 2K bytes
Multifunctional 16-bit Timers	TA0, TA1, TA2, TA3, TA4	Timer mode, Event counter, One-shot generator, PWM generation, Quadrature Counter
	TB0, TB1, TB2	Event counter, Period measurement, Pulse width measurement, Timer
	U/V/W	3-phase motor control PWM generator
Serial I/O		2 U(S)ART/SIO (UART0/1 type)
		1 U(S)ART/SIO/I^2C bus/IEBus (UART2) (Note 1,2)
A-D Converter		10 bits x 8 channels
DMAC		2 channels
Watchdog Timer		1 15-bit timer (switchable to internal ring oscillator)
Interrupts		20 internal, 7 external sources, 4 software, 7 priority levels
Clock Generation Circuit		3 built-in clock generating circuits • 2 internal clock generators (supports oscillator, crystal, resonator); crystal failure detect • 1 internal ring oscillator
Brown-out Monitor		With interrupt, reset and RAM retention flag capability
Programmable I/O		38 lines
Temperature Range		-20˚C to 85˚C / -40˚C to 85˚C (Note 3)
Supply Voltage		Vcc = 3.0V to 5.5 (f(fin) = 20MHz, no wait) Vcc = 2.7V to 5.5 (f(fin) = 10MHz, no wait)
Low Power Consumption		0.7 uA (typ) Icc, Stop mode, At Vcc = 3.0 V,
		1.8 µA (typ) Icc at 32 kHz, Wait mode, Vcc = 3.0 V
		25uA (typ) Icc at 32KHz mode with RAM access (Flash sleep)
		16 mA (typ) Icc at 20 MHz, Vcc = 5.0 V
I/O Characteristics	I/O Withstand Voltage	5.0V
	Output Current	5mA
Device Configuration		CMOS high performance silicon gate
Packages		48-pin LQFP

Note 1: I^2C bus is a registered trademark of Koninklijke Philips Electronics N. V.

Note 2: IEBus is a registered trademark of NEC Electronics Corporation.

Note 3: See Table 1.2b for erase/write cycle and temperature range information.

M16C Family Group

Figure 1.3 shows the type number, memory size and package for the M16C/26 group.

Tables 1.2a, 1.2b, and 1.2c list package, memory, and product marking information for configurations of the M16C/26 group.

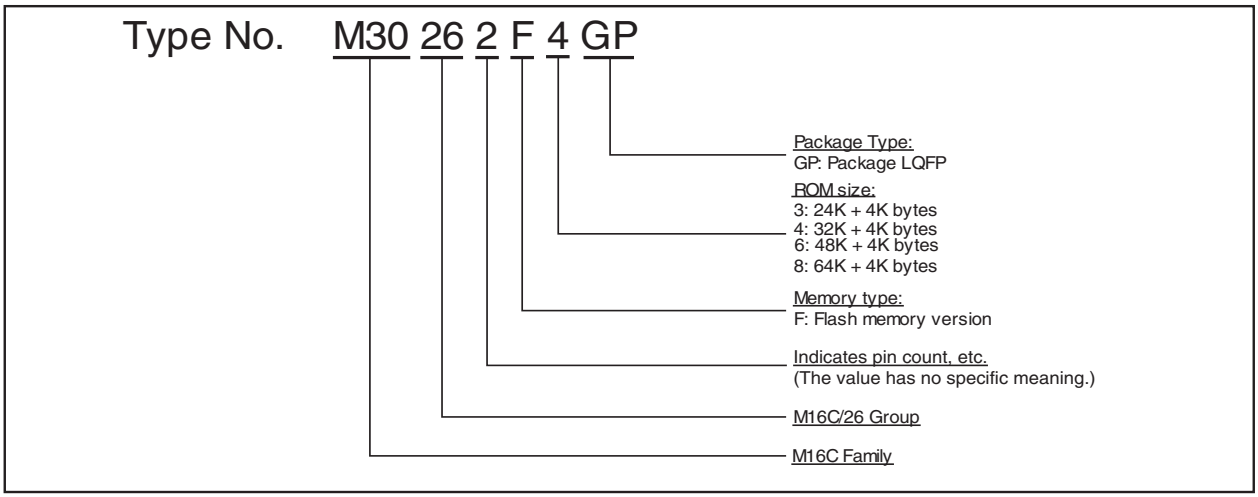


Figure 1.3. M3026x Product Family

Table 1.2a. Product list

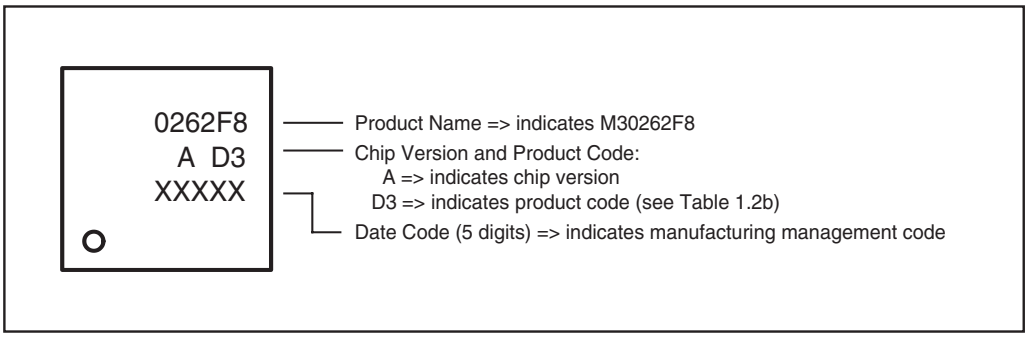
Type No.	Package Type	RAM Capacity	Flash ROM	Virtual EEPROM Flash
M30262F3GP	48 LQFP (48P6Q)	1024 bytes	24 Kbyte (8K x 3)	4 Kbyte (2 Kbyte x 2)
M30262F4GP			32 Kbyte (16K x 1, 8K x 2)	
M30262F6GP		2048 bytes	48 Kbyte (16K x 2, 8K x 2)	
M30262F8GP			64 Kbyte (32K x 1, 16K x 1, 8K x 2)	

Table 1.2b. Microcomputer versus Flash Erase/Write (E/W) Operating Ranges

Product Code	Package	Internal ROM Block (0,1,2,3)		Internal ROM Block (A,B)		Microcomputer operating temperature
		E/W cycles	Temperature range	E/W cycles	Temperature range	
D3	non-Pb free	100	0C to 60C	100	0C to 60C	-40C to 85C
D5						-20C to 85C
D7		1,000		10,000	-40C to 85C	-40C to 85C
D9					-20C to 85C	-20C to 85C
U3 ★	Pb free	100		100	0C to 60C	-40C to 85C
U5 ★						-20C to 85C
U7 ★		1,000		10,000	-40C to 85C	-40C to 85C
U9 ★					-20C to 85C	-20C to 85C

★: under planning

Table 1.2c. Product Marking (top view)



Pin Description

Pin name	Signal name	I/O type	Function
V_{CC}, V_{SS}	Power supply input		Supply 2.7V to 5.5V to the V_{CC} pin. Supply 0V to the V_{SS} pin.
CNV_{SS}	CNV_{SS}	Input	This pin is used to select, at reset, operation in either single-chip user or serial boot programming mode. If connected to V_{SS}, operation after reset will employ single-chip user mode. If connected to V_{CC}, reset will cause entry into boot mode.
RESET	Reset input	Input	A "L" on this input resets the microcomputer.
IV_{CC}	Power supply		The user must leave this pin unconnected.
X_{IN} X_{OUT}	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the X_{IN} and the X_{OUT} pins. To use an externally-derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
AV_{CC}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{CC}.
AV_{SS}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{SS}.
V_{REF}	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
$P1_5$ to $P1_7$	I/O port P1	Input/output	This is a 3-bit I/O port. When used for input pin, the port can be set to have or not have a pull-up resistor by software. $P1_5$ to $P1_7$ also function as the input pins for external interrupts INT_3 to INT_5 as selected by software. Additionally, $P1_5$ can be configured to act as the A/D trigger source (AD_{TRG}).
$P6_0$ to $P6_7$	I/O port P6	Input/output	This is an 8-bit I/O port. When used for input pin, the port can be set to have or not have a pull-up resistor in units of four bits by software. Pins in this port also function as UART0 and UART1 I/O pins as selected by software.
$P7_0$ to $P7_7$	I/O port P7	Input/output	This is an 8-bit I/O port. When used for input pin, the port can be set to have or not have a pull-up resistor in units of four bits by software ($P7_0$ to $P7_1$ are N channel open-drain outputs). Pins in this port also function as timer A0-A3 I/O, UART2 I/O pins, or 3-phase PWM V/W outputs as selected by software.
$P8_0$ to $P8_3$, $P8_5$ to $P8_7$	I/O port P8	Input/output	This is a 7-bit I/O port. When used for input pin, the port can be set to have or not have a pull-up resistor in units of four bits by software. Using software, they can be made to function as the I/O pins for timer A4, the input pins for external interrupts INT0, INT1, NMI, or 3-phase PWM U outputs and 3-phase PWM shutdown input. $P8_6$ to $P8_7$ can be set using software to function as the I/O pins for a sub clock generation circuit. In this case, connect a quartz oscillator between $P8_6$ (X_{COUT} pin) and $P8_7$ (X_{CIN} pin).
$P9_0$ to $P9_3$	I/O port P9	Input/output	This is a 4-bit I/O port. When used for input pin, the port can be set to have or not have a pull-up resistor in units of four bits by software. Pins in this port also function as Timer B0-B2 input pins as selected by software.
$P10_0$ to $P10_7$	I/O port P10	Input/output	This is an 8-bit I/O port. When used for input pin, the port can be set to have or not have a pull-up resistor in units of four bits by software Pins in this port also function as A-D converter input pins as selected by software. Furthermore, $P10_4$-$P10_7$ also function as input pins for the key input interrupt function pins as selected by software.

Table 1.3. Pin Descriptions

Functional Block Operation

The M16C/26 group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, serial I/O, A-D converter, and I/O ports.

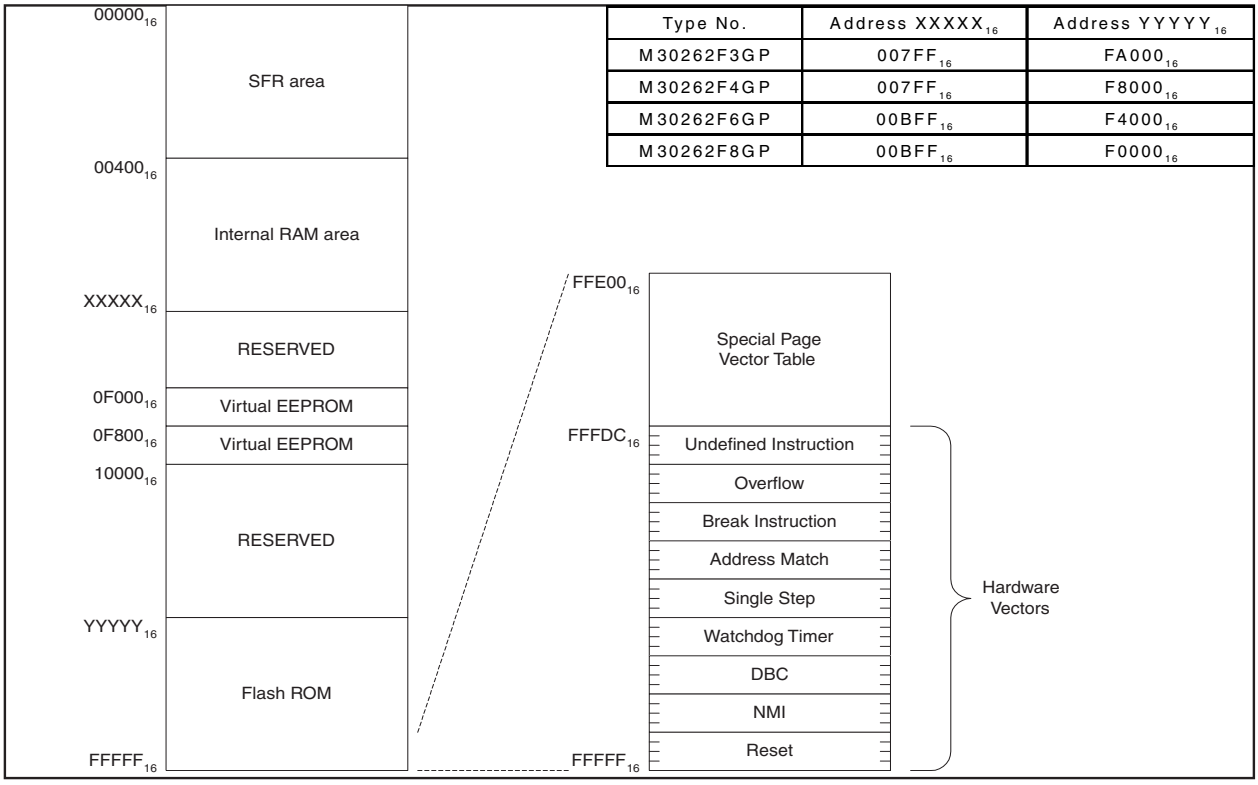
Memory

Figure 1.4.1 is a memory map of the M16C/26 group. The linear address space of 1M bytes extends from address 0000016 to FFFFF16. From FFFFF16 down is ROM. For example, in the M30262F4, there is 32K bytes of internal ROM from F800016 to FFFFF16. The vector table for fixed interrupts such as the reset and NMI are mapped to FFFDC16 to FFFFF16. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 0040016 up is RAM. For example, in the M30262F4, 1K bytes of internal RAM is mapped to the space from 0040016 to 07FF16. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

These devices also contain two blocks of Flash ROM as Virtual EEPROM memory to store data. These 2 blocks of 2K bytes are located from 0F00016 to 0FFFF16 on all versions.

The SFR area is mapped from 0000016 to 003FF16. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.



Type No.	Address XXXXX$_{16}$	Address YYYYY$_{16}$
M30262F3GP	007FF$_{16}$	FA000$_{16}$
M30262F4GP	007FF$_{16}$	F8000$_{16}$
M30262F6GP	00BFF$_{16}$	F4000$_{16}$
M30262F8GP	00BFF$_{16}$	F0000$_{16}$

Figure 1.4.1. Memory map

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 1.4.2. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

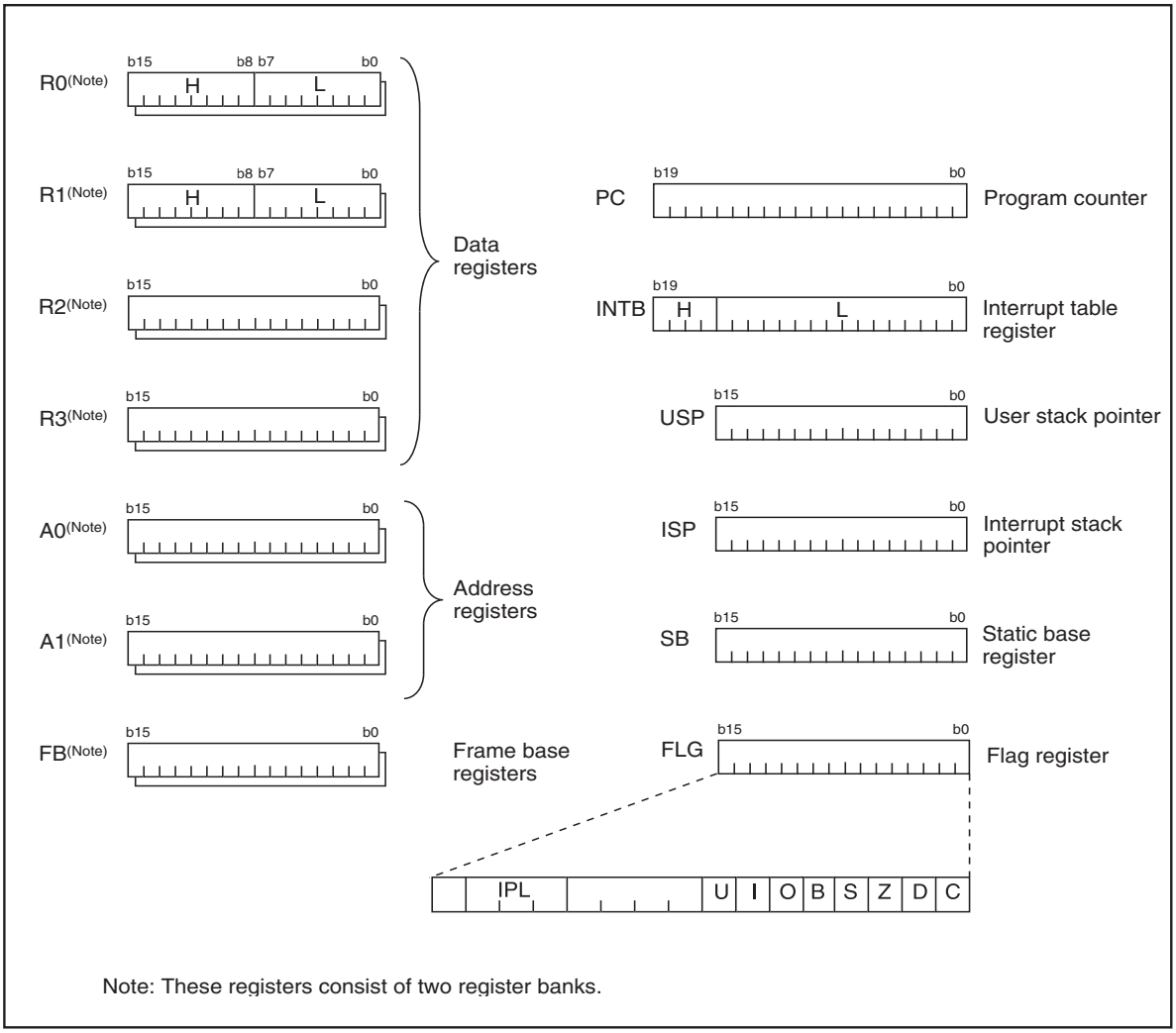


Figure 1.4.2. Central processing unit register

(1) Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, and R3)

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 can each be used as dual 8-bit data registers. As such, their high-order bytes are designated R0H/R1H, while their low-order bytes become R0L/R1L, respectively. In some instructions, registers R2 and R0, as well as R3 and R1, can be combined to serve as 32-bit data registers (R2R0/R3R1).

(2) Address registers (A0 and A1)

Address registers (A0 and A1) consist of 16 bits each. Individually, they can be used for three types of register-based addressing: register direct, address register indirect, and address register relative. They are also used for transfer and arithmetic/logic operations, as well as three special instruction addressing modes: address register relative with 20-bit displacement, 32-bit address register indirect, and 32-bit register direct. These last two modes combine A0 and A1 for use as a single 32-bit register (A1A0).

(3) Frame base register (FB)

Frame base register (FB) is configured with 16 bits, and is used for FB relative addressing.

(4) Program counter (PC)

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

(5) Interrupt table register (INTB)

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

(6) Stack pointer (USP/ISP)

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

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag). This flag is located at the position of bit 7 in the flag register (FLG).

(7) Static base register (SB)

Static base register (SB) is configured with 16 bits, and is used for SB relative addressing.

(8) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 1.4.3 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**
  This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.
- **Bit 1: Debug flag (D flag)**
  This flag enables a single-step interrupt.
  When this flag is “1”, a single-step interrupt is generated after instruction execution. This flag is cleared to “0” when the interrupt is acknowledged.
- **Bit 2: Zero flag (Z flag)**
  This flag is set to “1” when an arithmetic operation resulted in 0; otherwise, cleared to “0”.
- **Bit 3: Sign flag (S flag)**
  This flag is set to “1” when an arithmetic operation resulted in a negative value; otherwise, cleared to “0”.
- **Bit 4: Register bank select flag (B flag)**
  This flag chooses a register bank. Register bank 0 is selected when this flag is “0” ; register bank 1 is selected when this flag is “1”.
- **Bit 5: Overflow flag (O flag)**
  This flag is set to “1” when an arithmetic operation resulted in overflow.
- **Bit 6: Interrupt enable flag (I flag)**
  This flag enables a maskable interrupt.
  An interrupt is disabled when this flag is “0”, and is enabled when this flag is “1”. This flag is cleared to “0” when the interrupt is acknowledged.
- **Bit 7: Stack pointer select flag (U flag)**
  Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.
  This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.

- **Bits 8 to 11: Reserved area**
- **Bits 12 to 14: Processor interrupt priority level (IPL)**
  Processor interrupt priority level (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 the processor interrupt priority level (IPL), the interrupt is enabled.
- **Bit 15: Reserved area**

The C, Z, S, and O flags are changed when instructions are executed. See the software manual for details.

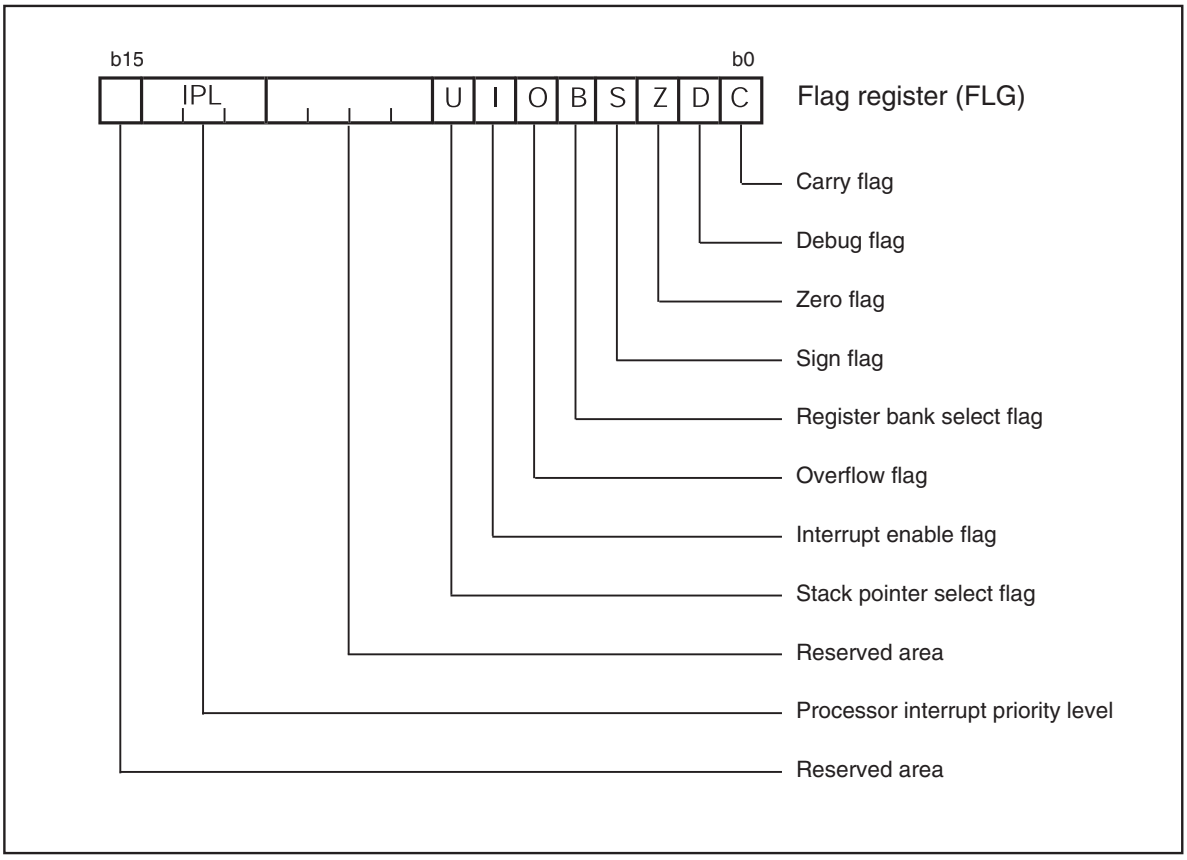


Figure 1.4.3. Flag register (FLG)

Reset

There are two types of reset: a hardware reset and a software reset (See “Software Reset” for details on software resets.)

Hardware reset

There are two kinds of hardware reset: hardware reset 1 and hardware reset 2.

Hardware reset 1

A reset is applied using the RESET pin. When a low-level signal (“L”) is applied to the RESET pin, while the power supply voltage is within the recommended operating voltage, the pins are initialized (see Table 1.5.1). When the input level of the RESET pin is released from low to high, the CPU and SFR are initialized. The program is executed starting from the address indicated by the reset vector. The internal RAM is not initialized. If the RESET pin is pulled low while writing to the internal RAM, the internal RAM becomes indeterminate.

Figure 1.5.1 shows an example of the reset circuit. Table 1.5.1 shows the status of the other pins while the RESET pin is “L”. Figure 1.5.2 shows the reset sequence. Figure 1.5.3 shows the status of the CPU registers. Refer to the SFR section on the status of SFR registers after reset.

1. When the power supply is stable (warm start)
   (1) Apply a low-level signal to the RESET pin
   (2) Supply at least 20 clock cycles to the XIN pin
   (3) Apply a high-level signal to the RESET pin
2. Power on (cold start)
   (1) Apply a low-level signal to the RESET pin
   (2) Turn on or increase the power supply voltage until it meets the recommended operation voltage.
   (3) Wait at least 2ms for the power supply to stabilize
   (4) Supply at least 20 clock cycles to the XIN pin
   (5) Apply a high-level signal to the RESET pin

Hardware reset 2

This reset is generated by the microcomputer's internal voltage detection circuit. The voltage detection circuit monitors the voltage supplied to the V_{CC} pin. A reset is generated when the V_{CC} input voltage drops to V_{DET3} or less while VCR2 register's VC26 bit is equal to “1” (V_{DET3} detection circuit enabled). When the input voltage increases so it is greater than V_{DET3}, the CPU, SFR, and pins are initialized, and the program is executed starting from the address indicated by the reset vector. After V_{DET3} is detected, it takes about 2ms before the program is executed. The status of the pins and registers after the reset is similar to that of hardware reset 1.

Take note, however, that the value of VC26 has no effect in stop mode. Therefore, a reset is not applied even if the V_{CC} input voltage drops to V_{DET3} or lower.

Software reset

When the PM0 register's PM03 bit is set “1” (microcomputer reset), the CPU, SFR, and pins of the microcomputer are initialized similar to a hardware reset. After reset, the program is executed starting from the address indicated by the reset vector.

Figure 1.5.2 shows the reset sequence and Figure 1.5.3 shows the status of the CPU registers after reset.

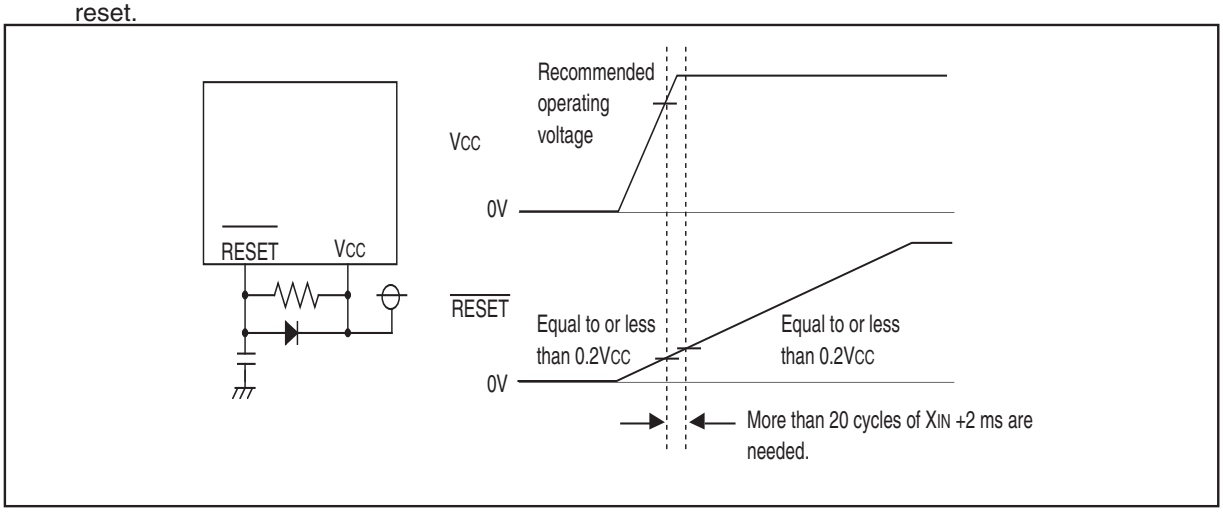


Figure 1.5.1. Example reset circuit

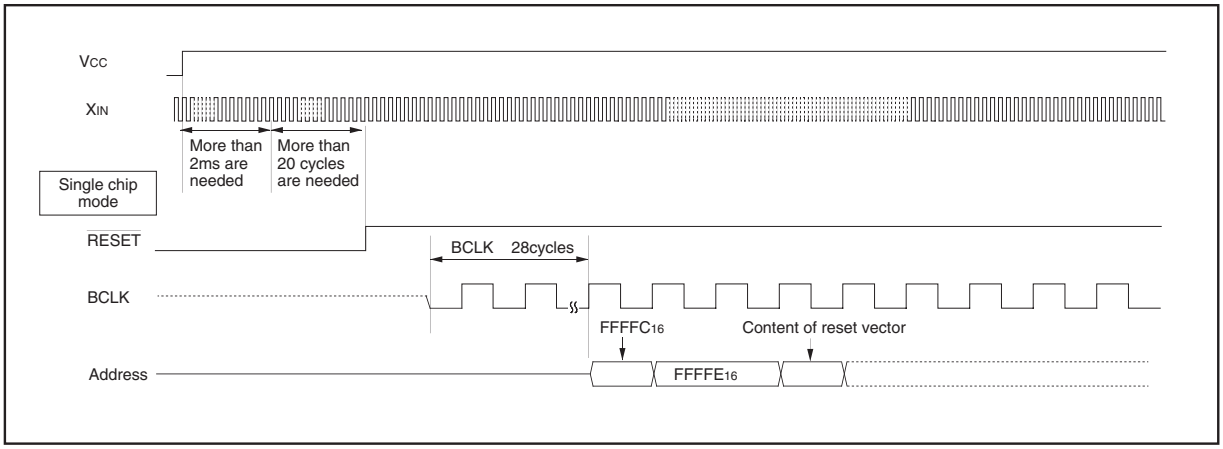


Figure 1.5.2. Reset sequence

Table 1.5.1. Pin status at RESET for User / Boot operating modes

Pin name	Status	
	User Mode (CNVSS = VSS)	Boot Mode (CNVSS = VCC)
P15 to P17	Input port (high impedance)	Data input (high impedance)
P60 to P63	Input port (high impedance)	Data input (high impedance)
P64	Input port (high impedance)	BUSY output
P65	Input port (high impedance)	SCLK input
P66	Input port (high impedance)	RXD input
P67	Input port (high impedance)	TXD output
P7	Input port (high impedance)	Data input (high impedance)
P80 to P83, P85	Input port (high impedance)	Data input (high impedance)
P86	Input port (high impedance)	$\overline{CE}$ input
P87	Input port (high impedance)	Data input (high impedance)
P90 to P93, P10	Input port (high impedance)	Data input (high impedance)

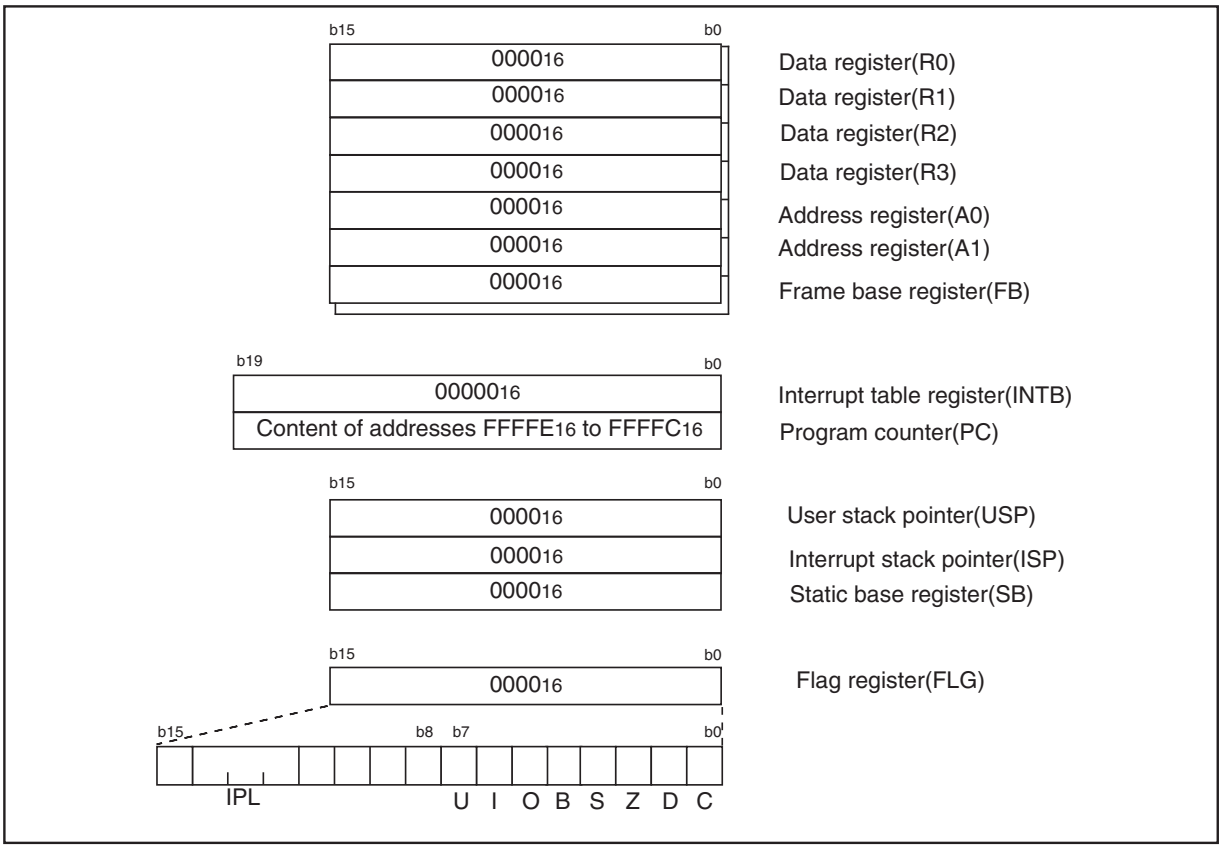


Figure 1.5.3. CPU register status after reset

Voltage detection circuit

The voltage detection circuit has monitoring circuits to check the input voltage of the V_{CC} pin. These circuits monitor the the input voltage at V_{DET3} and V_{DET4}. Bits VC26 and VC27 of VCR2 register are used to enable/disable these monitoring circuits.

The VDET3 monitoring circuit is used for detecting the minimum VCC that guarantees proper chip operation.

The V_{DET4} monitoring circuit can be set to detect whether the input voltage is equal to and greater or less than V_{DET4} depending on the value of VC13 bit of the VCR1 register. In addition, the V_{DET4} monitoring circuit, if enabled, can generate an interrupt.

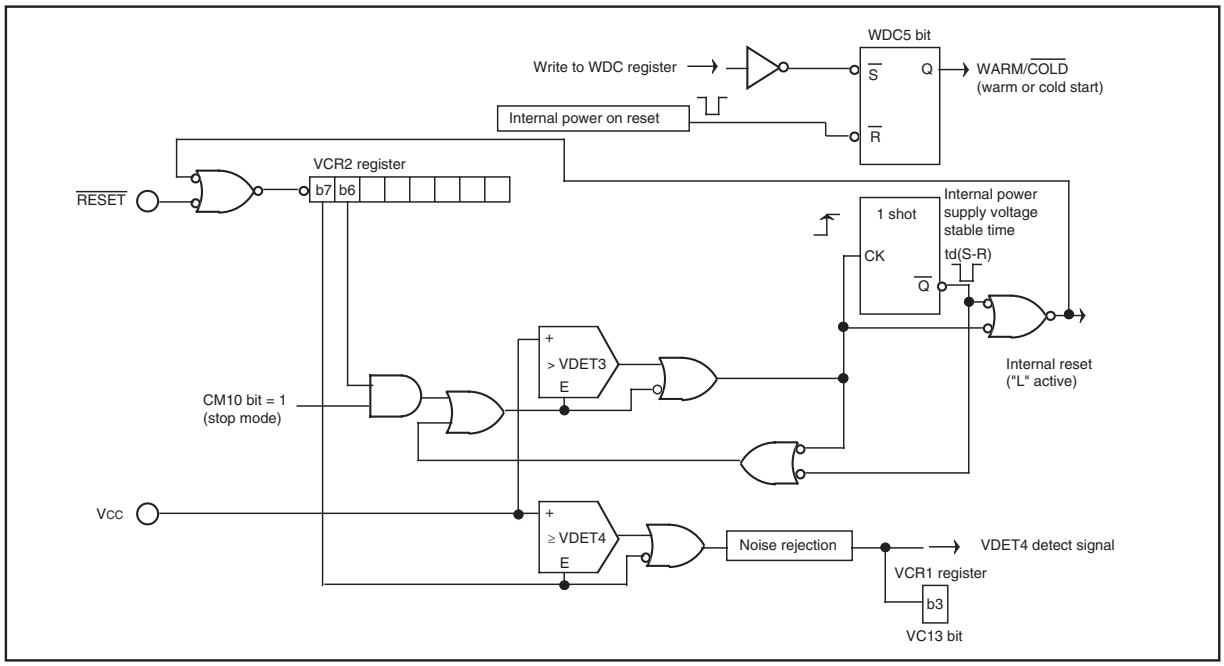


Figure 1.5.4. Reset circuit block

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol: WDC
Address: 000F16
After reset: 00?XXXX2(Note1)

Bit symbol	Bit name	Function	RW
High-order bit of watchdog timer			RO
WDC5	Cold start / warm start discrimination flag (Note 2)	0 : Cold start 1 : Warm start	RW
—	Reserved bit	Set to 0	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Note 1: The WDC5 bit is set to 0 immediately after power-on (cold start).
Note 2: The WDC5 bit will always be set to a 1 whenever the WDC SFR is written (regardless of value of WDC5 bit).

Figure 1.5.5. WDC register

Power supply detection register 1

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

Symbol: VCR1 | Address: 0019₁₆ | After reset: 00001000₂

Bit symbol	Bit name	F unction	RW
——	Reserved bit	Set to "0"	RW
VC13	V_{DET4} power supply monitor flag (Note)	0: $V_{CC} < V_{DET4}$ 1: $V_{CC} \geq V_{DET4}$	RO
——	Reserved bit	Set to "0"	RW

Note: The VC13 bit is useful when the VCR2 register's VC27 bit = 1 (V_{DET4} detection circuit enabled).
The VC13 bit is always 1 ($V_{CC} \geq V_{DET4}$) when the VCR2 register's VC27 bit = 0 (V_{DET4} detection circuit disabled).

Power supply detection register 2 (Note 1)

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

Symbol: VCR2 | Address: 001A₁₆ | After reset: 00₁₆

Bit symbol	Bit name	Function	RW
——	Reserved bit	Set to "0"	RW
VC26	Power supply V_{DET3} monitor bit	0: Disables detection circuit 1: Enables detection circuit	RW
VC27	Power supply V_{DET4} monitor bit (Note 2)	0: Disables detection circuit 1: Enables detection circuit	RW

Note 1: Write to this register after the PRCR register's PRC3 bit is set to "1" (write enabled).
Note 2: To use the VCR1 register's VC13 bit or D4INT register's D42 bit, set the VC27 bit to "1" (V_{DET4} detection circuit enabled).

Power supply V_{DET4} detection register (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

Symbol: D4INT | Address: 001F₁₆ | After reset: 00₁₆

Bit symbol	Bit name	Function	RW
D40	V_{DET4} detection interrupt enable bit.	0 : Disable 1 : Enable	RW
D41	STOP mode deactivation control bit (Note 4)	0: Disable (do not use the V_{DET4} detection interrupt to get out of stop mode) 1: Enable (use the V_{DET4} detection interrupt to get out of stop mode)	RW
D42	V_{DET4} up/down detection flag (Note 2)	0: Not detected 1: V_{DET4} up/down detected	RW (Note 3)
D43	WDT overflow detected flag	0: Not detected 1: Detected	RW (Note 3)
DF0	Sampling clock select bit	b5b4 00 : BCLK divided by 8 01 : BCLK divided by 16 10 : BCLK divided by 32 11 : BCLK divided by 64	RW
DF1			RW
—— (b7-b6)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

Note 1: Write to this register after the PRCR register's PRC3 bit is set to "1" (write enabled).
Note 2: Useful when the VCR2 register's VC27 bit = 1 (V_{DET4} detection circuit enabled). If the VC27 bit is cleared to 0 (V_{DET4} detection circuit disabled), the D42 bit is set to 0 (Not detected).
Note 3: This bit is cleared to "0" by writing a "0" in a program. (Writing a "1" has no effect.)
Note 4: If the V_{DET4} detection interrupt needs to be used to get out of stop mode again after once used for that purpose, reset the D41 bit by writing a 0 and then a 1.

Figure 1.5.6. VCR1, VCR2, and D4INT registers

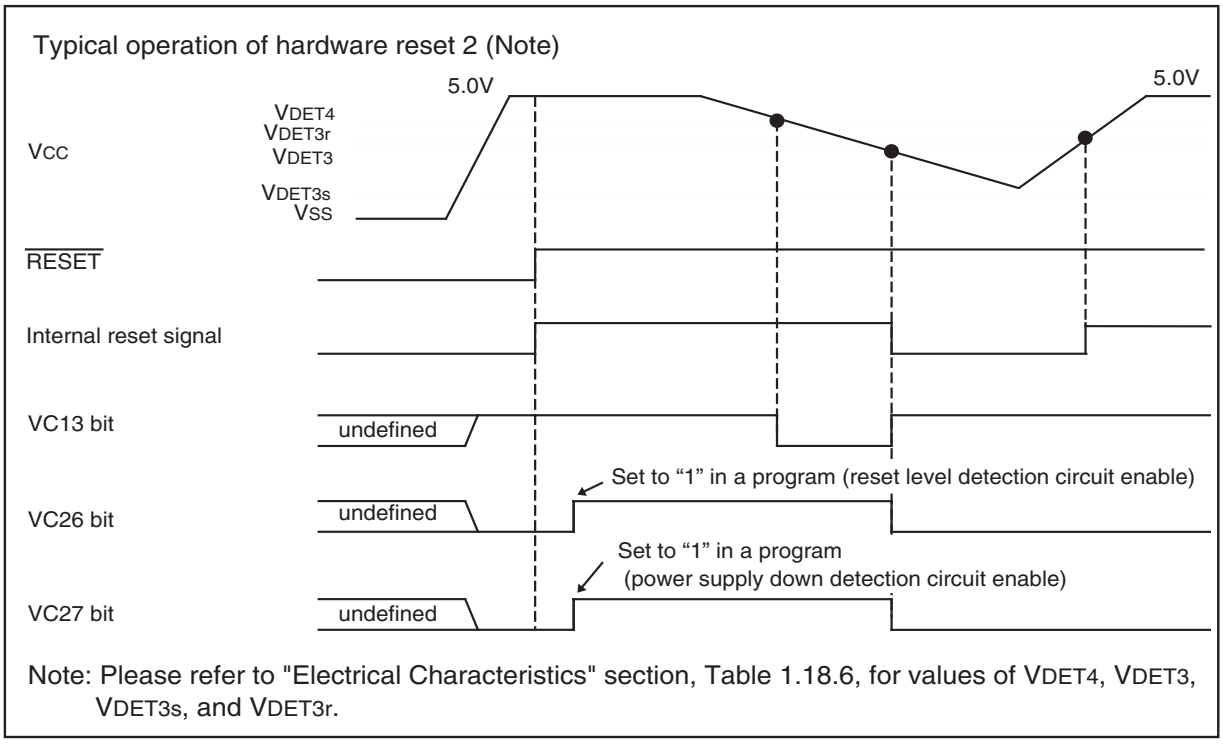


Figure 1.5.7. Typical operation of hardware reset 2

Voltage detection circuit

A V_{DET4} detection interrupt request is generated when the input voltage of the V_{CC} pin rises over V_{DET4} or drops under V_{DET4} while the D40 bit of the D4INT register is equal to “1” (V_{DET4} detection interrupt enabled). The V_{DET4} detection interrupt shares the interrupt vector with the watchdog timer and oscillation stop detection interrupts.

To use the V_{DET4} detection interrupt as a way to get out of stop mode, set D41 bit of the D4INT register to “1” (enabled) prior to going into stop mode.

D42 bit of the the D4INT register is set to “1” when an input voltage over or under V_{DET4} is detected. A V_{DET4} detection interrupt is generated when the D42 bit changes state from “0” to “1”. The D42 bit needs to be cleared to “0" in the program. However, while in stop mode, if the input voltage goes over V_{DET4} when the D41 bit is equal to “1”, a V_{DET4} detection interrupt is generated regardless of the value of D42, which causes the microcomputer to get out of stop mode.

Table 1.5.2 shows the conditions under which a V_{DET4} detection interrupt request is generated.

The sampling clock, used for detecting when the input voltage goes over or under V_{DET4}, is set using the DF1 and DF0 bits of the D4INT register. Table 1.5.3 shows the sampling clock periods.

Table 1.5.2. V_{DET4} detection interrupt request generation conditions

operation mode	VC27 bit	D40 bit	D41 bit	D42 bit	CM02 bit	VC13 bit
Normal operation mode (Note 1)	1	1	—— (Note 2)	0 to 1	——	0 to 1 (Note 3)
						1 to 0 (Note 3)
Wait mode (Note 4)			——	0 to 1	0	0 to 1 (Note 3)
						1 to 0 (Note 3)
Stop mode (Note 4)				——	1	0 to 1
			1	——	0	0 to 1

Note 1: The status is handled as normal mode when not in wait or stop modes. (Refer to "Clock Generating Circuit".)
Note 2: "——" implies either "0" or "1".
Note 3: An interrupt request for voltage reduction is generated a sampling time after the value of the VC13 bit has changed. Refer to Figure 1.5.9 "V_{DET4} detection interrupt generation circuit operation example" for details.
Note 4: Refer to "Limitations on stop mode", "Limitations on wait instructions with peripheral clocks turned off".

Table 1.5.3. Sampling clock periods

BCLK (MHz)	Sampling time (μs)			
	DF1 to DF0=00 (BCLK divided by 8)	DF1 to DF0=01 (BCLK divided by 16)	DF1 to DF0=10 (BCLK divided by 32)	DF1 to DF0=11 (BCLK divided by 64)
16	3.0	6.0	12.0	24.0

Precautions

1. Limitations on stop mode

With the VC13 bit of the VCR1 register equal to “1” ($V_{CC} \geq V_{DET4}$), VC27 bit of the VCR2 register equal to “1” (V_{DET4} detection circuit enabled), and D40 bit of the D4INT register equal to “1” (V_{DET4} detection interrupt enabled), if the CM10 bit of the CM1 register is set to “1” (stop mode), a V_{DET4} detection interrupt is immediately generated, causing the microcomputer to get out of stop mode.

In systems where the microcomputer enters the stop mode when the input voltage in the V_{CC} pin drops under V_{DET4} and exits from stop mode when the input voltage goes over V_{DET4}, ensure that the CM10 bit is set to “1” when VC13 = “0” ($V_{CC} < V_{DET4}$)

2. Limitations on wait instructions with peripheral clocks turned off

If the WAIT instruction is executed when bit VC13 of the VCR1 register is equal to "1" ($V_{CC} \geq V_{DET4}$), bit VC27 of the VCR2 register is equal to "1" (V_{DET4} detection circuit enabled), and bit D40 of the D4INT register is equal to "1" (V_{DET4} detection interrupt enabled), a V_{DET4} detection interrupt is immediately generated, causing the microcomputer to exit from wait mode.

In systems where the microcomputer enters wait mode when the input voltage in the V_{CC} pin drops under V_{DET4} and exits from wait mode when the input voltage goes over V_{DET4}, ensure that the WAIT instruction is executed when VC13 = "0" ($V_{CC} < V_{DET4}$).

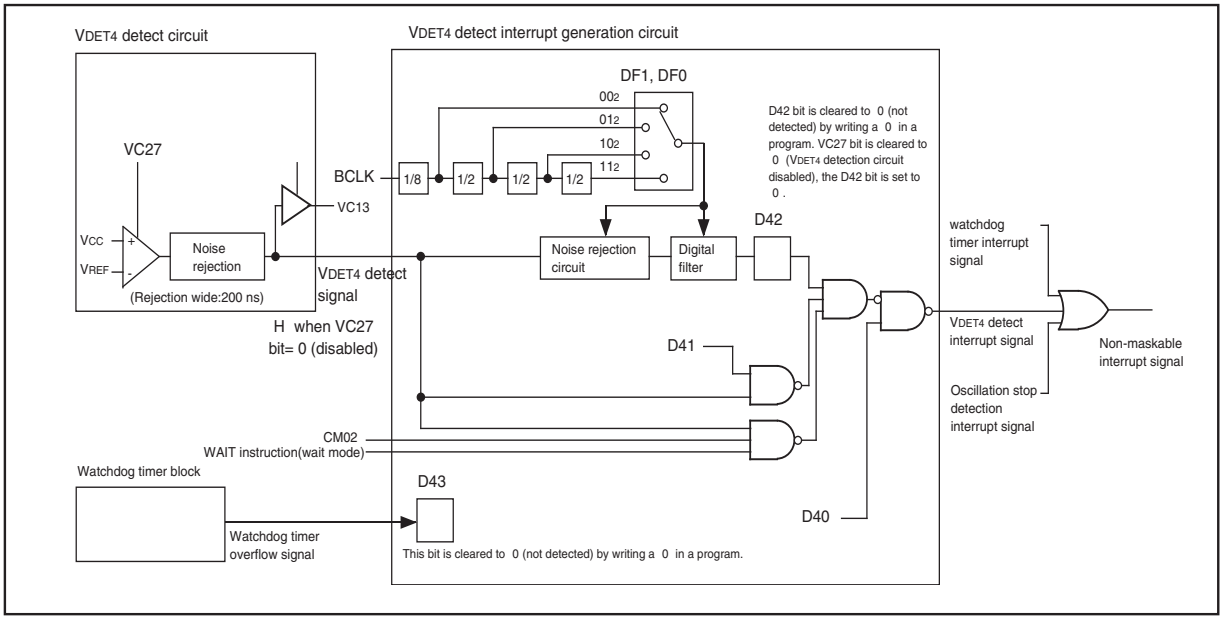
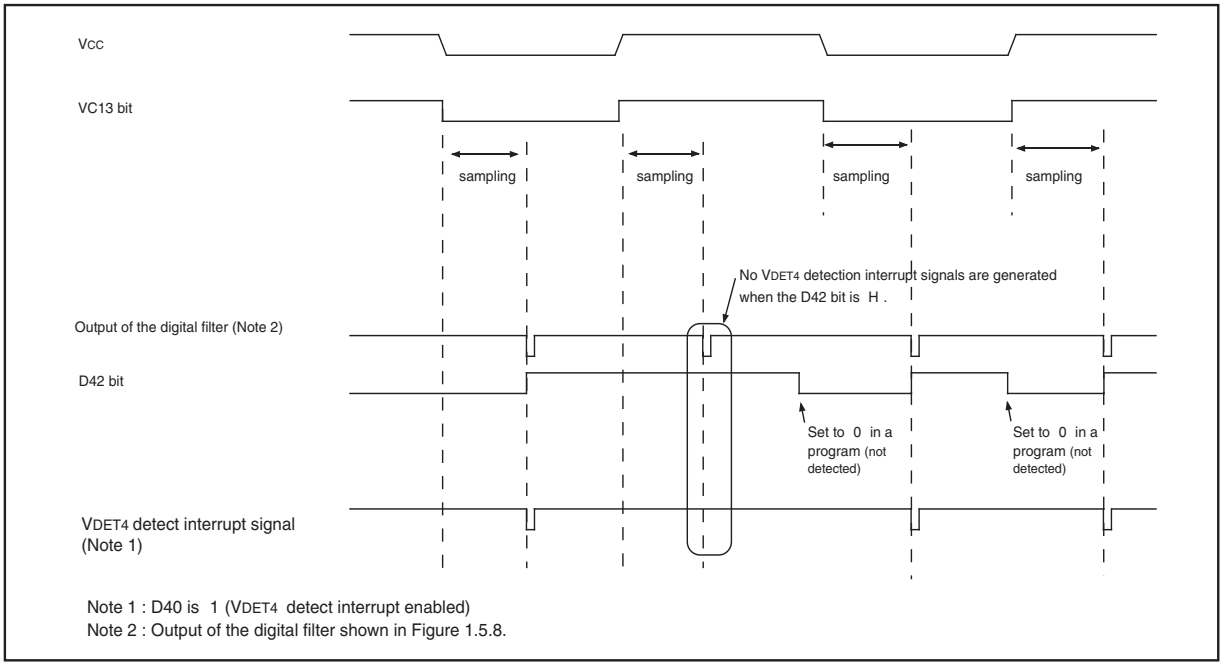


Figure 1.5.8. V_{DET4} detection interrupt generation block



Note 1 : D40 is 1 (V_{DET4} detect interrupt enabled)
Note 2 : Output of the digital filter shown in Figure 1.5.8.

Figure 1.5.9. V_{DET4} detection interrupt generation circuit operation example

Special Function Registers

Figures 1.6.1 through 1.6.5 display special function register (SFR) names, addresses, acronyms, and reset values.

Address	Register Name	Acronym	Value after Reset
0000$_{16}$			
0001$_{16}$			
0002$_{16}$			
0003$_{16}$			
0004$_{16}$	Processor mode register 0	PM0	0 0 0 0 0 0 0 0
0005$_{16}$	Processor mode register 1	PM1	0 0 0 0 1 0 0 0
0006$_{16}$	System clock control register 0	CM0	0 1 0 0 1 0 0 0
0007$_{16}$	System clock control register 1	CM1	0 0 1 0 0 0 0 0
0008$_{16}$			
0009$_{16}$	Address match interrupt enable register	AIER	0 0
000A$_{16}$	Protect register	PRCR	0 0 0 0 0 0
000B$_{16}$			
000C$_{16}$	Oscillation stop detection register	CM2	0 0 0 0 0 0 0 0
000D$_{16}$			
000E$_{16}$	Watchdog timer start register	WDTS	?
000F$_{16}$	Watchdog timer control register	WDC	0 0 ? ? ? ? ? ?
0010$_{16}$	Address match interrupt register 0 (low)	RMAD0	00$_{16}$
0011$_{16}$	Address match interrupt register 0 (mid)		00$_{16}$
0012$_{16}$	Address match interrupt register 0 (high)		0 0 0 0
0013$_{16}$			
0014$_{16}$	Address match interrupt register 1 (low)	RMAD1	00$_{16}$
0015$_{16}$	Address match interrupt register 1 (mid)		00$_{16}$
0016$_{16}$	Address match interrupt register 1 (high)		0 0 0 0
0017$_{16}$			
0018$_{16}$			
0019$_{16}$	Power supply detection register 1	VCR1	0 0 0 0 1 0 0 0
001A$_{16}$	Power supply detection register 2	VCR2	0 0 0 0 0 0 0 0
001B$_{16}$			
001C$_{16}$			
001D$_{16}$			
001E$_{16}$	Processor mode register 2	PM2	0 0 0 0 0
001F$_{16}$	Power supply 4 V detection register	D4INT	0 0 0 0 0 0 0 0
0020$_{16}$	DMA0 source pointer (low)	SAR0	?
0021$_{16}$	DMA0 source pointer (mid)		?
0022$_{16}$	DMA0 source pointer (high)		? ? ? ?
0023$_{16}$			
0024$_{16}$	DMA0 destination pointer (low)	DAR0	?
0025$_{16}$	DMA0 destination pointer (mid)		?
0026$_{16}$	DMA0 destination pointer (high)		? ? ? ?
0027$_{16}$			
0028$_{16}$	DMA0 transfer counter (low)	TCR0	?
0029$_{16}$	DMA0 transfer counter (high)		?
002A$_{16}$			
002B$_{16}$			
002C$_{16}$	DMA0 control register	DM0CON	0 0 0 0 0 ? 0 0
002D$_{16}$			
002E$_{16}$			
002F$_{16}$			
0030$_{16}$	DMA1 source pointer (low)	SAR1	?
0031$_{16}$	DMA1 source pointer (mid)		?
0032$_{16}$	DMA1 source pointer (high)		? ? ? ?
0033$_{16}$			
0034$_{16}$	DMA1 destination pointer (low)	DAR1	?
0035$_{16}$	DMA1 destination pointer (mid)		?
0036$_{16}$	DMA1 destination pointer (high)		? ? ? ?
0037$_{16}$			
0038$_{16}$	DMA1 transfer counter (low)	TCR1	?
0039$_{16}$	DMA1 transfer counter (high)		?
003A$_{16}$			
003B$_{16}$			
003C$_{16}$	DMA1 control register	DM1CON	0 0 0 0 0 ? 0 0
003D$_{16}$			
003E$_{16}$			
003F$_{16}$			

	Reserved address, do not write
?	Value indeterminate at reset
0	reset value is zero
1	reset value is one
0	reserved bit, must write to zero
1	reserved bit, must write to one
	Nothing is mapped to this bit (value is indeterminate when read)

Figure 1.6.1. Location of peripheral unit control registers (1)

Address	Register Name	Acronym	Value after Reset
0040$_{16}$			
0041$_{16}$			
0042$_{16}$			
0043$_{16}$			
0044$_{16}$	INT3 interrupt control register	INT3IC	XX00?000
0045$_{16}$			
0046$_{16}$			
0047$_{16}$			
0048$_{16}$	INT5 interrupt control register	INT5IC	XX00?000
0049$_{16}$	INT4 interrupt control register	INT4IC	XX00?000
004A$_{16}$	Bus collision detection interrupt control register	BCNIC	XXXX?000
004B$_{16}$	DMA0 interrupt control register	DM0IC	XXXX?000
004C$_{16}$	DMA1 interrupt control register	DM1IC	XXXX?000
004D$_{16}$	Key input interrupt control register	KUPIC	XXXX?000
004E$_{16}$	A-D conversion interrupt control register	ADIC	XXXX?000
004F$_{16}$	UART2 transmit interrupt control register	S2TIC	XXXX?000
0050$_{16}$	UART2 receive interrupt control register	S2RIC	XXXX?000
0051$_{16}$	UART0 transmit interrupt control register	S0TIC	XXXX?000
0052$_{16}$	UART0 receive interrupt control register	S0RIC	XXXX?000
0053$_{16}$	UART1 transmit interrupt control register	S1TIC	XXXX?000
0054$_{16}$	UART1 receive interrupt control register	S1RIC	XXXX?000
0055$_{16}$	Timer A0 interrupt control register	TA0IC	XXXX?000
0056$_{16}$	Timer A1 interrupt control register	TA1IC	XXXX?000
0057$_{16}$	Timer A2 interrupt control register	TA2IC	XXXX?000
0058$_{16}$	Timer A3 interrupt control register	TA3IC	XXXX?000
0059$_{16}$	Timer A4 interrupt control register	TA4IC	XXXX?000
005A$_{16}$	Timer B0 interrupt control register	TB0IC	XXXX?000
005B$_{16}$	Timer B1 interrupt control register	TB1IC	XXXX?000
005C$_{16}$	Timer B2 interrupt control register	TB2IC	XXXX?000
005D$_{16}$	INT0 interrupt control register	INT0IC	XX00?000
005E$_{16}$	INT1 interrupt control register	INT1IC	XX00?000
005F$_{16}$			
to			
01B3$_{16}$	Flash Control Register 4	FMR4	01000000
01B4$_{16}$			
01B5$_{16}$	Flash Control Register 1	FMR1	0XX00101
01B6$_{16}$			
01B7$_{16}$	Flash Control Register 0	FMR0	00000001
01B8$_{16}$			
01B9$_{16}$			
01BA$_{16}$			
01BB$_{16}$			
01BC$_{16}$			
01BD$_{16}$			
01BE$_{16}$			
01BF$_{16}$			
01C0$_{16}$			
01C1$_{16}$			
01C2$_{16}$			
to			
025C$_{16}$			
025D$_{16}$			
025E$_{16}$	Peripheral Clock Select Register	PLCKR	00000011
025F$_{16}$			
0260$_{16}$			
to			
033E$_{16}$			
033F$_{16}$			

	Reserved address, do not write
?	Value indeterminate at reset
0	reset value is zero
1	reset value is one
0	reserved bit, must write to zero
1	reserved bit, must write to one
	Nothing is mapped to this bit (value is indeterminate when read)

Figure 1.6.2. Location of peripheral unit control registers (2)

Address	Register Name	Acronym	Value after Reset
0340_{16}			
0341_{16}			
0342_{16}	Timer A1-1 register (low)	TA11	?
0343_{16}	Timer A1-1 register (high)		?
0344_{16}	Timer A2-1 register (low)	TA21	?
0345_{16}	Timer A2-1 register (high)		?
0346_{16}	Timer A4-1 register (low)	TA41	?
0347_{16}	Timer A4-1 register (high)		?
0348_{16}	Three-phase PWM control register 0	INVC0	0 0 0 0 0 0 0 0
0349_{16}	Three-phase PWM control register 1	INVC1	0 0 0 0 0 0 0 0
$034A_{16}$	Three-phase output buffer register 0	IDB0	0 0 0 0 0 0 0 0
$034B_{16}$	Three-phase output buffer register 1	IDB1	0 0 0 0 0 0 0 0
$034C_{16}$	Dead time timer	DTT	?
$034D_{16}$	Timer B2 interrupt occurrences frequency set counter	ICTB2	? ? ? ? ? ? ? ?
$034E_{16}$			
$034F_{16}$			
0350_{16}			
0351_{16}			
0352_{16}			
0353_{16}			
0354_{16}			
0355_{16}			
0356_{16}			
0357_{16}			
0358_{16}			
0359_{16}			
$035A_{16}$			
$035B_{16}$			
$035C_{16}$			
$035D_{16}$			
$035E_{16}$			
$035F_{16}$	Interrupt request cause select register	IFSR	0 0 0 0 0 0 0 0
0360_{16}			
0361_{16}			
0362_{16}			
0363_{16}			
0364_{16}			
0365_{16}			
0366_{16}			
0367_{16}			
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	0 0 0 0 0 0 0 0
0375_{16}	UART2 special mode register 3	U2SMR3	0 0 0 0 0 0
0376_{16}	UART2 special mode register 2	U2SMR2	0 0 0 0 0 0 0
0377_{16}	UART2 special mode register	U2SMR	0 0 0 0 0 0 0
0378_{16}	UART2 transmit/receive mode register	U2MR	0 0 0 0 0 0 0 0
0379_{16}	UART2 bit rate generator	U2BRG	?
$037A_{16}$	UART2 transmit buffer register (low)	U2TB	?
$037B_{16}$	UART2 transmit buffer register (high)		?
$037C_{16}$	UART2 transmit/receive control register 0	U2C0	0 0 0 0 1 0 0 0
$037D_{16}$	UART2 transmit/receive control register 1	U2C1	0 0 0 0 0 0 1 0
$037E_{16}$	UART2 receive buffer register (low)	U2RB	?
$037F_{16}$	UART2 receive buffer register (high)		? ? ? ? ? ?

	Reserved address, do not write
?	Value indeterminate at reset
0	reset value is zero
1	reset value is one
0	reserved bit, must write to zero
1	reserved bit, must write to one
	Nothing is mapped to this bit (value is indeterminate when read)

Figure 1.6.3. Location of peripheral unit control registers (3)

Address	Register Name	Acronym	Value after Reset
0380$_{16}$	Count start flag	TABSR	0 0 0 0 0 0 0 0
0381$_{16}$	Clock prescaler reset flag	CPSRF	0
0382$_{16}$	One-shot start flag	ONSF	0 0 0 0 0 0 0 0
0383$_{16}$	Trigger select register	TRGSR	0 0 0 0 0 0 0 0
0384$_{16}$	Up/down flag	UDF	0 0 0 0 0 0 0 0
0385$_{16}$			
0386$_{16}$	Timer A0 register (low)	TA0	?
0387$_{16}$	Timer A0 register (high)		?
0388$_{16}$	Timer A1 register (low)	TA1	?
0389$_{16}$	Timer A1 register (high)		?
038A$_{16}$	Timer A2 register (low)	TA2	?
038B$_{16}$	Timer A2 register (high)		?
038C$_{16}$	Timer A3 register (low)	TA3	?
038D$_{16}$	Timer A3 register (high)		?
038E$_{16}$	Timer A4 register (low)	TA4	?
038F$_{16}$	Timer A4 register (high)		?
0390$_{16}$	Timer B0 register (low)	TB0	?
0391$_{16}$	Timer B0 register (high)		?
0392$_{16}$	Timer B1 register (low)	TB1	?
0393$_{16}$	Timer B1 register (high)		?
0394$_{16}$	Timer B2 register (low)	TB2	?
0395$_{16}$	Timer B2 register (high)		?
0396$_{16}$	Timer A0 mode register	TA0MR	0 0 0 0 0 0 0 0
0397$_{16}$	Timer A1 mode register	TA1MR	0 0 0 0 0 0 0 0
0398$_{16}$	Timer A2 mode register	TA2MR	0 0 0 0 0 0 0 0
0399$_{16}$	Timer A3 mode register	TA3MR	0 0 0 0 0 0 0 0
039A$_{16}$	Timer A4 mode register	TA4MR	0 0 0 0 0 0 0 0
039B$_{16}$	Timer B0 mode register	TB0MR	0 0 ? ? 0 0 0 0
039C$_{16}$	Timer B1 mode register	TB1MR	0 0 ? 0 0 0 0
039D$_{16}$	Timer B2 mode register	TB2MR	0 0 ? 0 0 0 0
039E$_{16}$	Timer B2 special mode register	TB2SC	0 0
039F$_{16}$			
03A0$_{16}$	UART0 transmit/receive mode register	U0MR	0 0 0 0 0 0 0 0
03A1$_{16}$	UART0 bit rate generator	U0BRG	?
03A2$_{16}$	UART0 transmit buffer register (low)	U0TB	?
03A3$_{16}$	UART0 transmit buffer register (high)		?
03A4$_{16}$	UART0 transmit/receive control register 0	U0C0	0 0 0 0 1 0 0 0
03A5$_{16}$	UART0 transmit/receive control register 1	U0C1	0 0 0 0 0 0 1 0
03A6$_{16}$	UART0 receive buffer register (low)	U0RB	?
03A7$_{16}$	UART0 receive buffer register (high)		? ? ? ? ? ?
03A8$_{16}$	UART1 transmit/receive mode register	U1MR	0 0 0 0 0 0 0 0
03A9$_{16}$	UART1 bit rate generaT1r	U1BRG	?
03AA$_{16}$	UART1 transmit buffer register (low)	U1TB	?
03AB$_{16}$	UART1 transmit buffer register (high)		?
03AC$_{16}$	UART1 transmit/receive control register 0	U1C0	0 0 0 0 1 0 0 0
03AD$_{16}$	UART1 transmit/receive control register 1	U1C1	0 0 0 0 0 0 1 0
03AE$_{16}$	UART1 receive buffer register (low)	U1RB	?
03AF$_{16}$	UART1 receive buffer register (high)		? ? ? ? ? ?
03B0$_{16}$	UART transmit/receive control register 2	UCON	0 0 0 0 0 0 0
03B1$_{16}$			
03B2$_{16}$			
03B3$_{16}$			
03B4$_{16}$			
03B5$_{16}$			
03B6$_{16}$			
03B7$_{16}$			
03B8$_{16}$	DMA0 request cause select register	DM0SL	0 0 0 0 0 0 0
03B9$_{16}$			
03BA$_{16}$	DMA1 request cause select register	DM1SL	0 0 0 0 0 0 0
03BB$_{16}$			
03BC$_{16}$			
03BD$_{16}$			
03BE$_{16}$			
03BF$_{16}$			

	Reserved address, do not write
?	Value indeterminate at reset
0	reset value is zero
1	reset value is one
0	reserved bit, must write to zero
1	reserved bit, must write to one
	Nothing is mapped to this bit (value is indeterminate when read)

Figure 1.6.4. Location of peripheral unit control registers (4)

Address	Register Name	Acronym	Value after Reset
03C0$_{16}$	A-D register 0 (low)	AD0	? ? ? ? ? ? ? ?
03C1$_{16}$	A-D register 0 (high)		? ?
03C2$_{16}$	A-D register 1 (low)	AD1	? ? ? ? ? ? ? ?
03C3$_{16}$	A-D register 1 (high)		? ?
03C4$_{16}$	A-D register 2 (low)	AD2	? ? ? ? ? ? ? ?
03C5$_{16}$	A-D register 2 (high)		? ?
03C6$_{16}$	A-D register 3 (low)	AD3	? ? ? ? ? ? ? ?
03C7$_{16}$	A-D register 3 (high)		? ?
03C8$_{16}$	A-D register 4 (low)	AD4	? ? ? ? ? ? ? ?
03C9$_{16}$	A-D register 4 (high)		? ?
03CA$_{16}$	A-D register 5 (low)	AD5	? ? ? ? ? ? ? ?
03CB$_{16}$	A-D register 5 (high)		? ?
03CC$_{16}$	A-D register 6 (low)	AD6	? ? ? ? ? ? ? ?
03CD$_{16}$	A-D register 6 (high)		? ?
03CE$_{16}$	A-D register 7 (low)	AD7	? ? ? ? ? ? ? ?
03CF$_{16}$	A-D register 7 (high)		? ?
03D0$_{16}$			
03D1$_{16}$			
03D2$_{16}$			
03D3$_{16}$			
03D4$_{16}$	A-D control register 2	ADCON2	0 0 0 0 0 0 0 0
03D5$_{16}$			
03D6$_{16}$	A-D control register 0	ADCON0	0 0 0 0 0 ? ? ?
03D7$_{16}$	A-D control register 1	ADCON1	0 0 0 0 0 0 0 0
03D8$_{16}$			
03D9$_{16}$			
03DA$_{16}$			
03DB$_{16}$			
03DC$_{16}$			
03DD$_{16}$			
03DE$_{16}$			
03DF$_{16}$			
03E0$_{16}$			
03E1$_{16}$	Port P1 register	P1	? ? ? 0 0 0 0 0
03E2$_{16}$			
03E3$_{16}$	Port P1 direction register	PD1	0 0 0 0 0 0 0 0
03E4$_{16}$			
03E5$_{16}$			
03E6$_{16}$			
03E7$_{16}$			
03E8$_{16}$			
03E9$_{16}$			
03EA$_{16}$			
03EB$_{16}$			
03EC$_{16}$	Port P6 register	P6	? ? ? ? ? ? ? ?
03ED$_{16}$	Port P7 register	P7	? ? ? ? ? ? ? ?
03EE$_{16}$	Port P6 direction register	PD6	0 0 0 0 0 0 0 0
03EF$_{16}$	Port P7 direction register	PD7	0 0 0 0 0 0 0 0
03F0$_{16}$	Port P8 register	P8	? ? ? 0 ? ? ? ?
03F1$_{16}$	Port P9 register	P9	0 0 0 0 ? ? ? ?
03F2$_{16}$	Port P8 direction register	PD8	0 0 0 0 0 0 0 0
03F3$_{16}$	Port P9 direction register	PD9	0 0 0 0 0 0 0 0
03F4$_{16}$	Port P10 register	P10	? ? ? ? ? ? ? ?
03F5$_{16}$			
03F6$_{16}$	Port P10 direction register	PD10	0 0 0 0 0 0 0 0
03F7$_{16}$			
03F8$_{16}$			
03F9$_{16}$			
03FA$_{16}$			
03FB$_{16}$			
03FC$_{16}$	Pull-up control register 0	PUR0	0 0 0 0 0 0 0 0
03FD$_{16}$	Pull-up control register 1	PUR1	0 0 0 0 0 0 0 0
03FE$_{16}$	Pull-up control register 2	PUR2	0 0 0 0 0 0 0 0
03FF$_{16}$	Port control register	PCR	0 0 0 0 0 0 0 0

	Reserved address, do not write
?	Value indeterminate at reset
0	reset value is zero
1	reset value is one
0	reserved bit, must write to zero
1	reserved bit, must write to one
	Nothing is mapped to this bit (value is indeterminate when read)

Figure 1.6.5. Location of peripheral unit control registers (5)

Processor Mode

This device functions in single-chip mode only. In single-chip mode, only internal memory space (SFR, internal RAM, and internal ROM) can be accessed. Ports P1_5 to P1_7, P6, P7, P8, P9_0 to P9_3 and P10 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions. Figure 1.7.1 shows the processor mode registers 0 and 1. Figure 1.7.2 shows the processor mode register 2.

Processor mode register 0 (Note)

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

Symbol: PM0 | Address: 0004$_{16}$ | When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
Reserved Bit		Must always be set to "0".	○	○
PM03	Software reset bit	The device is reset when this bit is set to 1. The value of this bit is 0 when read.	○	○
Reserved Bit		Must always be set to "0".	○	○

Note: Set bit 1 of the protect register (address 000A$_{16}$) to 1 when writing new values to this register.

Processor mode register 1 (Note 1)

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

Symbol: PM1 | Address: 0005$_{16}$ | When reset: 00001000$_2$

Bit symbol	Bit name	Function	R	W
PM10	Flash data block access bit (Note 4)	0: Disabled 1: Enabled (Note 3)	○	○
Reserved bit		Must always be set to "0".	○	○
PM12	Watchdog timer function select bit	0 : Interrupt 1 : Reset (Note 2)	○	○
Reserved bit		Must always be set to "1".	○	○
Reserved bit		Must always be set to "0".	○	○
Reserved bit		Must always be set to 0 . The value, if read, turns out to be indeterminate.	×	○
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

Note 1: Set bit 1 of the protect register (address 000A$_{16}$) to 1 when writing new values to this register.
Note 2: This bit can only be set to 1.
Note 3: When EW entry bit (FMR01) =1, this bit is automatically set to 1 during that time.
Note 4: To access the two 2K byte data blocks in address range F800$_{16}$ to FFFF$_{16}$, this bit must be set to 1.

Figure 1.7.1. Processor mode register 0 and 1

Processor mode register 2 (Note 1)

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

Symbol: PM2 | Address: 001E16 | When reset: XXX000002

Bit symbol	Bit name	Function	R	W
PM20	Specifying wait when accessing SFR registers (Note 2)	0 : 2 waits 1 : 1 wait	○	○
PM21	System clock protective bit (Notes 3,4)	0 : Clock is protected by PRCR register 1 : Clock modification disabled	○	○
PM22	WDT count source protective bit (Notes 3,5)	0 : CPU clock is used for the watchdog timer count source 1 : Ring oscillator clock is used for the watchdog timer count source	○	○
Reserved bit		Must always be set to "0"	○	○
PM24	P85/$\overline{\text{NMI}}$ configuration bit (Notes 3,6,7)	0: P85 function ($\overline{\text{NMI}}$ disable) 1: $\overline{\text{NMI}}$ function	○	○
Nothing is assigned. Writes must set each bit to "0". Read values are indeterminate.			—	—

Note 1: Set bit 1 of the protect register (address 000A16) to "1" before writing new values to this register.
Note 2: When the system clock is 16Mhz and more, must set to 2 waits.
Note 3: This bit cannot be changed from "1" to "0" in software.
Note 4: Setting PM21 to "1" results in the following:
- BCLK is not halted by WAIT instruction.
- Writing to the following bits has no effect:
  CM02, of register CM0
  CM05, of register CM0 (main clock is not halted)
  CM07, of register CM0 (CPU clock source does not change)
  CM10, of register CM1 (stop mode is not entered)
  CM20, of register CM2 (oscillation stop, re-oscillation detection function settings do not change)

Note 5: Setting PM22 to "1" results in the following:
- The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.
- Writing to CM10 disabled (stop mode is not entered).
- Watchdog timer does not stop when in wait mode or hold state.

Note 6: For NMI function, this bit must be set to "1" in first instruction after reset.
Note 7: When this bit is set to "1", the P85 direction register must be "0".

Figure 1.7.2. Processor mode register 2

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 000416) applies a (software) reset to the microcomputer. A software reset has the same effect as a hardware reset except the contents of internal RAM are preserved after a software reset.

Note: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address 000A16) to “1”.

Software wait

A software wait can be inserted by setting the wait bit (bit 7) of the processor mode register 1 (address 0005_{16}) (Note) and bits 4 to 7 of the chip select control register (address 0008_{16}).

A software wait is inserted in the internal ROM/RAM area by setting the wait bit of processor mode register 1. When set to "0", each bus cycle is executed in one BCLK cycle. When set to "1", each bus cycle is executed in two BCLK cycles. After the microcomputer has been reset, this bit defaults to "0". When set to "1", a wait is applied to all memory areas (two BCLK cycles). Set this bit after referring to the recommended operating conditions (main clock input oscilliation frequecny) of the electrical characteristics.

The SFR area is always accessed in two BCLK cycles regardless of the setting of these control bits.

Table 1.7.1 shows the software wait and bus cycles.

Note: Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address $000A_{16}$) to “1”.

Table 1.7.1. Software waits and bus cycles

Area	Wait bit	Bus cycle
Internal ROM/RAM	0	1 BCLK cycle
	1	2 BCLK cycles

Clock Generating Circuit

The clock generating circuit contains three oscillator circuits as follows:

(1) Main clock generating circuit

(2) Sub clock generating circuit

(3) Ring oscillator

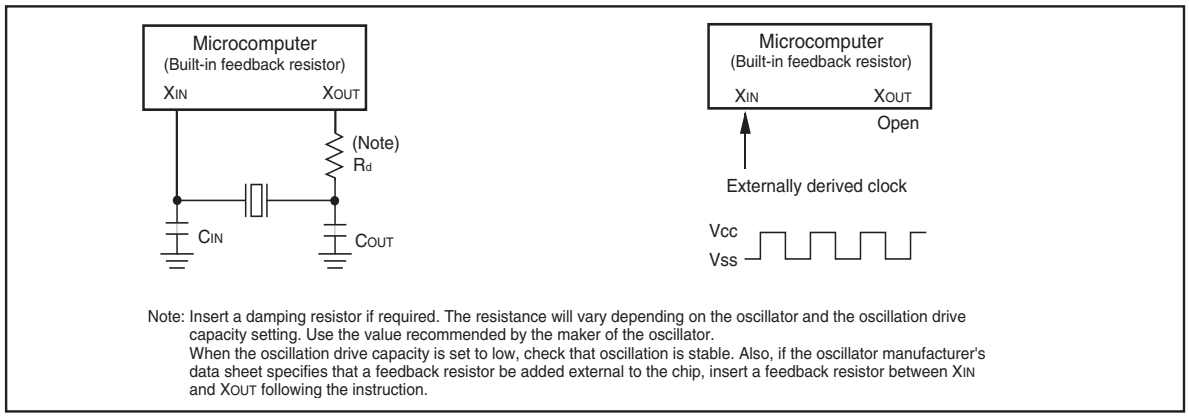
Table 1.8.1 lists the clock generating circuit specifications.

Table 1.8.1. Clock generating circuits

Example of oscillator circuit

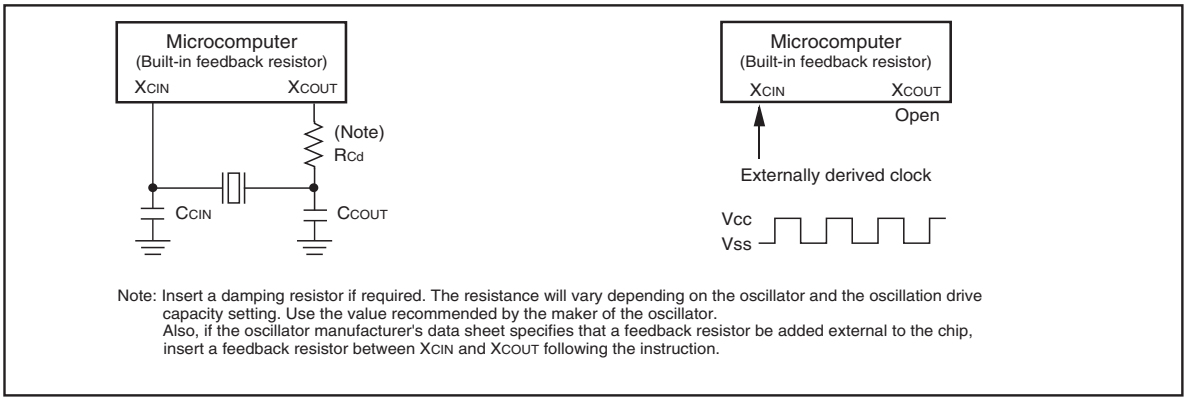
Item	Main clock generating circuit	Sub clock generating circuit	Ring oscillator
Use of clock	- CPU's operating clock source - Internal peripheral unit's operating	- CPU's operating clock source - Timer A/B's count clock source	- CPU clock source - Peripheral function clock source - CPU and peripheral function clock source when main clock frequency stops
Usable oscillator	- Ceramic oscillator - Crystal oscillator	- Crystal oscillator	About 1MHz
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	
Oscillation stop/ restart function	Available	Available	Available
Oscillator status after reset	Oscillating	Stopped	Stopped
Other	Externally derived clock can be input		

Figure 1.8.1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure 1.8.2 shows some examples of sub-clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures 1.8.1 and 1.8.2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XIN and XOUT following the instruction.

Figure 1.8.1. Examples of main clock



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XCIN and XCOUT following the instruction.

Figure 1.8.2. Examples of sub-clock

Clock Control

Figure 1.8.3 shows the block diagram of the clock generating circuit.

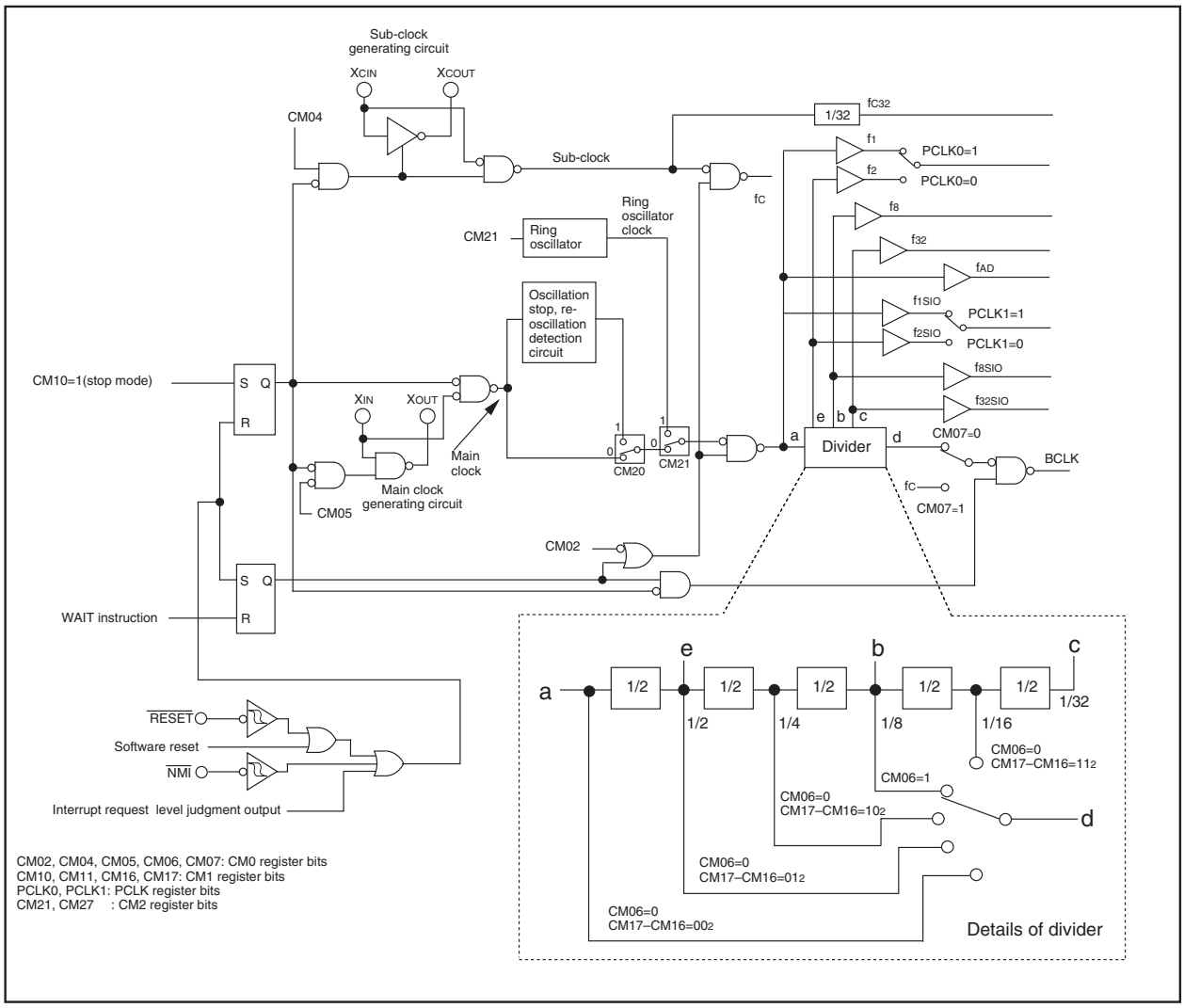


Figure 1.8.3. Clock generating circuit

The following paragraphs describes the clocks generated by the clock generating circuit.

(1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock, after switching the operating clock source of CPU to the sub-clock, reduces the power dissipation. After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.

(2) Sub-clock

The sub-clock is generated by the sub-clock oscillation circuit. No sub-clock is generated after a reset. After oscillation is started using the port XC select bit (bit 4 at address 0006_{16}), the sub-clock can be selected as the BCLK by using the system clock select bit (bit 7 at address 0006_{16}). However, be sure that the sub-clock oscillation has fully stabilized before switching.

After the oscillation of the sub-clock oscillation circuit has stabilized, the drive capacity of the sub-clock oscillation circuit can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the sub-clock oscillation circuit reduces the power dissipation. This bit changes to "1" when the port XC select bit (bit 4 at address 0006_{16}) is set to "0" , shifting to stop mode and at a reset.

When the XCIN/XCOUT is used, set ports $P8_6$ and $P8_7$ as the input ports without pull-up.

(3) BCLK

The BCLK is the clock that drives the CPU, and is fC or the clock is derived by dividing the main clock by 1, 2, 4, 8, or 16. The BCLK is derived by dividing the main clock by 8 after a reset.

The main clock division select bit 0(bit 6 at address 0006_{16}) changes to "1" when shifting from high-speed/medium-speed to stop mode, shifting to low power dissipation mode and at reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.

(4) Peripheral function clock(f_1, f_8, f_{32}, f_{1SIO2}, f_{8SIO2},f_{32SIO2},f_{AD})

The clock for the peripheral devices is derived from the main clock or by dividing it by 1, 8, or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006_{16}) to "1" and then executing a WAIT instruction.

By setting the timer A, B clock select bit (bit 0 at address $001E_{16}$) and SIO clock select bit (bit 1 at address $001E_{16}$) to "1" respectively, f_1 and f_{1SIO2} can be changed to the main clock divided by 2.

(5) fC32

This clock is derived by dividing the sub-clock by 32. It is used for the timer A and timer B counts.

(6) fC

This clock has the same frequency as the sub-clock. It is used for the BCLK and for the watchdog timer.

Figure 1.8.4a shows the system clock control registers 0 and 1 and Figure 1.8.4b shows peripheral clock select register.

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CM0 | Address: 000616 | When reset: 4816

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to 0	O	O
Reserved bit		Must always be set to 0	O	O
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	O	O
CM03	XCIN-XCOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM04	Port XC select bit (Note 10)	0 : I/O port 1 : XCIN-XCOUT generation (Note 9)	O	O
CM05	Main clock (XIN-XOUT) stop bit (Note 3, 4, 5)	0 : On 1 : Off	O	O
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	O	O
CM07	System clock select bit (Note 6)	0 : XIN, XOUT 1 : XCIN, XCOUT	O	O

Note 1: Set bit 0 of the protect register (address 000A16) to 1 before writing to this register.
Note 2: Changes to 1 when shifting to stop mode and at a reset.
Note 3: When entering low power dissipation mode, main clock stops by using this bit. To stop the main clock, when the sub clock oscillation is stable, set system clock select bit (CM07) to 1 before setting this bit to 1 . The main clock division select bit 0 (CM06) and the XIN-XOUT drive capacity select bit (CM15) change to 1 when this bit is set to 1 .
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to 1 , XOUT turns H . The built-in feedback resistor remains being connected, so XIN turns pulled up to XOUT (H) via the feedback resistor.
Note 6: Set port XC select bit (CM04) to 1 and stabilize the sub-clock oscillating before setting this bit from 0 to 1 . Do not write to both bits at the same time. And also, set the main clock stop bit (CM05) to 0 and stabilize the main clock oscillating before setting this bit from 1 to 0 .
Note 7: This bit changes to 1 when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.
Note 8: fC32 is not included. Do not set to 1 when using low-speed or low power dissipation mode.
Note 9: When the XCIN/XCOUT is used, set ports P86 and P87 as the input ports without pull-up.
Note 10: When setting this bit to "0", XCIN-XCOUT drive capacity select bit becomes "1".

System clock control register 1 (Note 1)

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

Symbol: CM1 | Address: 000716 | When reset: 2016

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note4)	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bit		Must always be set to 0	O	O
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM16	Main clock division select bit 1 (Note 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	O	O
CM17				

Note 1: Set bit 0 of the protect register (address 000A16) to 1 before writing to this register.
Note 2: This bit changes to 1 when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 000616) is 0 . If 1 , division mode is fixed at 8.
Note 4: If this bit is set to 1 , XOUT turns H , and the built-in feedback resistor is cut off. XCIN and XCOUT turn high-impedance state.

Figure 1.8.4a. Clock control registers 0 and 1

Peripheral clock select register (Note)

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

Symbol: PCLKR
Address: 025E16
When reset: 0316

Bit symbol	Bit name	Function	R	W
PCLK0	Timers A, B clock select bit (Clock source for the timers A, B, and the dead time timer)	0 : f_2 1 : f_1	○	○
PCLK1	SI/O clock select bit (Clock source for UART0 to UART2)	0 : f_{2SIO} 1 : f_{1SIO}	○	○
Reserved bit		Must always be set to 0	○	○

Note: Set the bit 0 of the protect register (address 000A16) to "1" before rewriting this register.

Figure 1.8.4b. Peripheral clock select register

Processor mode register 2 (Note 1)

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

Symbol: PM2
Address: 001E16
When reset: XXX000002

Bit symbol	Bit name	Function	R	W
PM20	Specifying wait when accessing SFR registers (Note 2)	0 : 2 waits 1 : 1 wait	○	○
PM21	System clock protective bit (Notes 3,4)	0 : Clock is protected by PRCR register 1 : Clock modification disabled	○	○
PM22	WDT count source protective bit (Notes 3,5)	0 : CPU clock is used for the watchdog timer count source 1 : Ring oscillator clock is used for the watchdog timer count source	○	○
Reserved bit		Must always be set to "0"	○	○
PM24	$P8_5/\overline{NMI}$ configuration bit (Notes 3,6,7)	0: $P8_5$ function ($\overline{NMI}$ disable) 1: $\overline{NMI}$ function	○	○
Nothing is assigned. Writes must set each bit to "0". Read values are indeterminate.			—	—

Note 1: Set bit 1 of the protect register (address 000A16) to "1" before writing new values to this register.
Note 2: When the system clock is 16Mhz and more, must set to 2 waits.
Note 3: This bit cannot be changed from "1" to "0" in software.
Note 4: Setting PM21 to "1" results in the following:
- BCLK is not halted by WAIT instruction.
- Writing to the following bits has no effect:
  CM02, of register CM0
  CM05, of register CM0 (main clock is not halted)
  CM07, of register CM0 (CPU clock source does not change)
  CM10, of register CM1 (stop mode is not entered)
  CM20, of register CM2 (oscillation stop, re-oscillation detection function settings do not change)

Note 5: Setting PM22 to "1" results in the following:
- The ring oscillator starts oscillating, and the ring oscillator clock becomes the watchdog timer count source.
- Writing to CM10 disabled (stop mode is not entered).
- Watchdog timer does not stop when in wait mode or hold state.

Note 6: For NMI function, this bit must be set to "1" in first instruction after reset.
Note 7: When this bit is set to "1", the $P8_5$ direction register must be "0".

Figure 1.8.4c. Processor mode register 2

Oscillation Stop Detection Function

The oscillation stop detection function detects abnormal stopping of the clock by causes such as opening and shorting of the XIN oscillation circuit. When oscillation stop is detected, either an internal reset or an oscillation stop detection interrupt is generated. The selection depends on the value in the bit 7 of the oscillation stop detection register (address $000C_{16}$). When an oscillation stop detection interrupt is generated, the ring oscillator in the microcomputer operates automatically and is used as the system clock in place of the XIN clock. This allows interrupt processing.

The oscillation stop detection function can be enabled/disabled with bit 0 of the oscillation stop detection register. When this bit is set to "1," the function is enabled. After the reset is released, the oscillation stop detection function becomes disabled because the bit value is "0."

Table 1.8.2 gives an specification overview of the oscillation stop detection function, Figure 1.8.5 is a configuration diagram of the oscillation stop detection circuit and Figure 1.8.6 shows the configuration of the oscillation stop detection register.

Table 1.8.2. Specification overview of the oscillation stop detection function

Item	Specification
Oscillation stop detectable clock and frequency bandwidth	f(XIN) ≥ 2 MHz
Enabling condition for oscillation stop detection function	When the oscillation stop detection bit (bit 0 of address $000C_{16}$) is set to "1"
Operation at oscillation stop, re-oscillation detection	• Internal reset occurs (when bit CM27 = "0") • Oscillation stop, re-oscillation detection interrupt occurs (when bit CM27 = "1")
Notes on STOP mode	Before setting up the stop mode, write "0" in the oscillation stop detection enable bit to disable the oscillation stop detection function. Write "1" in the bit again after the stop mode is.

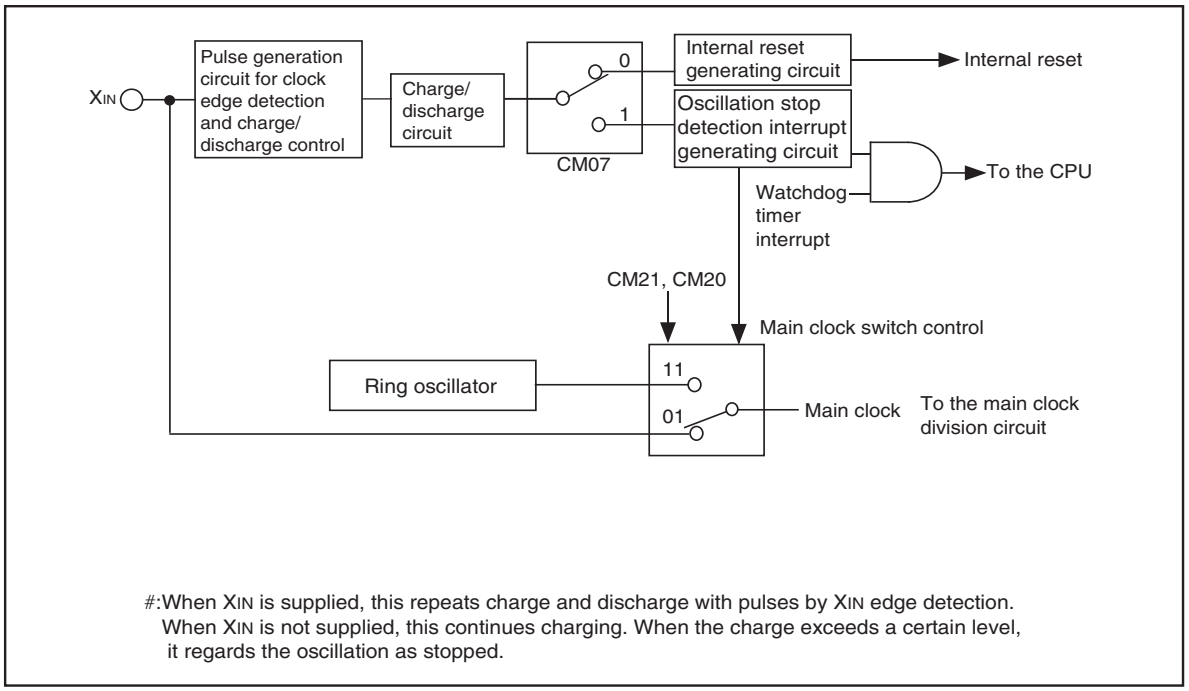
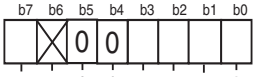


Figure 1.8.5. Oscillation stop detection circuit

Oscillation stop detection register (Note 1)



Symbol CM2 | Address $000C_{16}$ | When reset $0X000000_2$ (Note 10)

Bit symbol	Bit name	Function	R	W
CM20	Oscillation stop, re-oscillation detection bit (Notes 7, 8, 9, 10)	0 : Oscillation stop, re-oscillation detection function is disabled. 1 : Oscillation stop, re-oscillation detection function enabled.	○	○
CM21	CPU clock switch bit (Notes 2, 3, 6, 10)	0 : Main clock (Ring oscillator turned off 1 : Ring oscillator clock (Ring oscillator oscillating)	○	○
CM22	Oscillation stop, re-oscillation detection flag (Note 4)	0 : Main clock stop, re-oscillation not detected 1 : Main clock stop, re-oscillation detected	○	○
CM23	XIN monitor flag (Note 5)	0 : Main clock oscillating 1 : Main clock turned off	○	×
Reserved bit		Must always be set to "0".	○	○
Nothing is assigned. Writes must set to "0". Reads are indeterminate.			—	—
CM27	Operation select bit (when an oscillation stop, re-oscillation is detected) (Note 10)	0 : Reset 1 : Oscillation stop, re-oscillation detection interrupt	○	○

Note 1: Write to this register after setting bit PRC0, of register PRCR, to "1" (write enable).
Note 2: CM21 is set to "1" (ring oscillator clock) when the main clock is stopped, and the following conditions exist:
- CM20 is "1" (oscillation stop, re-oscillation detection function enabled).
- CM27 is "1" (oscillation stop, re-oscillation detection interrupt).
- CPU clock source is the main clock.

Note 3: If CM20 is "1", and CM23 is "1" (main clock turned off), do not set CM21 to "0".
Note 4: This bit is set to "1" by main clock stop detection and re-oscillation detection. Each oscillation stop, re-oscillation detection interrupt must correspond to a "0" to "1" transition of CM22. Once an interrupt is received, the bit can only be cleared via an explicit "0" write operation. Otherwise, further interrupts will be disabled. Only "0" writes can affect the bit value. CM22 is also used as a means for distinguishing between those interrupts caused by the oscillation stop, re-oscillation detection circuitry, and those initiated by the watchdog timer.
Note 5: Read CM23 in an oscillation stop, re-oscillation detection interrupt handling routine to determine the main clock status.
Note 6: Effective when bit CM07, of register CM0, is "0".
Note 7: When bit PM21, of register PM2, is "1" (clock modification disabled), writing to CM20 has no effect.
Note 8: Set CM20 to "0" (disable) before entering stop mode. After exiting stop mode, set CM20 back to "1" (enable).
Note 9: Set CM20 to "0" (disable) before setting bit CM05, of register CM0.
Note 10: CM20, CM21, and CM27 do not change at oscillation stop detection reset.

Figure 1.8.6. Oscillation stop detection register

Oscillation stop detection bit (CM20)

You can start the oscillation stop detection by setting this bit to “1”. The detection is not executed when this bit is set to “0” or in reset status. Be sure to set this bit to “0” before setting for the stop-mode. Set this bit again to “1” after release from stop-mode. This is because it is necessary to cancel the oscillation stop detection function due to a certain period of unstable oscillation after release from stop-mode. Set this bit to “0” also before setting the main clock stop bit (bit 5 at 0006_{16}) to “1”.

Do not set this bit to “1” if the frequency of XIN is lower than 2 MHz.

Main clock switch bit (CM21)

You can use the ring oscillator as a system clock by setting this bit to “1”. When this bit is "0", the ring oscillator is not in operation. For more explanation, see the section of the clock generating circuit.

Oscillation stop detection status (CM22)

You can see the status of the oscillation stop detection. When this bit is “1”, an oscillation stop is detected. For usage of this bit, see the explanation on CM27.

Clock monitor bit (CM23)

You can see the operation status of the XIN clock. When this bit is “0”, XIN is operating correctly. You can check the operation status of XIN when an oscillation stop detection interrupt is generated.

Operation select (when an oscillation stop is detected) bit (CM27)

(1) Operation when internal reset is selected (CM27 is set to “0”.)

An internal reset is generated when an abnormal stop of XIN is detected. The microcomputer stops in reset status and does not operate further.

Note: Release from this status is only possible through an external reset. However, in case of a defect XIN clock, further operation cannot be compensated.

All ports are configured to input port (floating) after an internal reset is generated.

(2) Operation when oscillation stop detection interrupt is selected (CM27 is set to “1”.)

An oscillation stop detection interrupt is generated when an abnormal stop of XIN is detected. The ring oscillator starts operation instead of the XIN clock which stopped abnormally. The operation goes further with the supply from the ring oscillator. For the oscillation stop detection interrupt judgement on the interrupt condition is necessary, because this interrupt shares the vector table with watchdog timer interrupt. Use the oscillation stop detection status (CM22) for the judgment. By clearing CM22, the acceptance of oscillation stop detection interrupt resumes. Figure 1.8.7 shows the flow of the judgment.

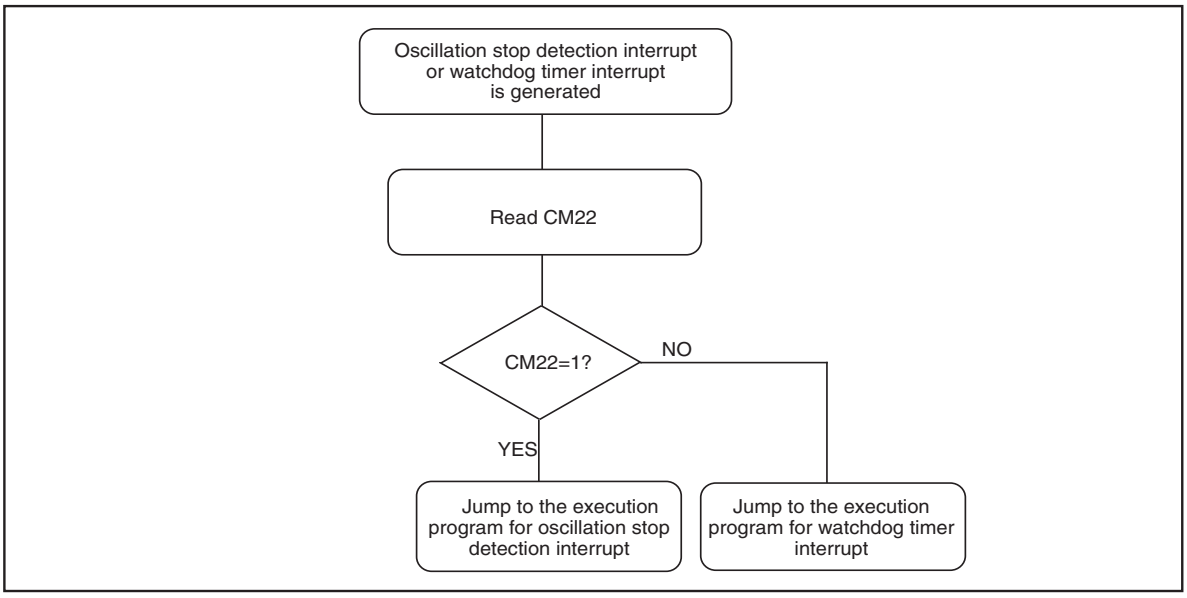


Figure 1.8.7. Flow of the judgment

Stop Mode

Writing “1” to the all-clock stop control bit (bit 0 at address 0007_{16}) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V_{CC} remains above 2V.

Because the oscillation , BCLK, f_1 to f_{32}, f_{1SIO2} to f_{32SIO2}, f_C, f_{C32}, and f_{AD} stops in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UARTi(i = 0 to 2) functions provided an external clock is selected. Port pins retain their status from before stop mode is entered.

Stop mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled, and the priority level of the interrupts which are not used to cancel must have been changed to 0. If returning by an interrupt, that interrupt routine is executed. If only a hardware reset or an NMI interrupt is used to cancel stop mode, change the priority level of all interrupts to 0, then shift to stop mode.

The main clock division select bit 0 (bit 6 at address 0006_{16}) changes to “1” when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained.

Wait Mode

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and watchdog timer stop. Writing “1” to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. However, peripheral function clock f_{C32} does not stop so that the peripherals using f_{C32} do not contribute to the power saving. When the microcomputer is running in low-speed or low power dissipation mode, do not enter WAIT mode with this bit set to “1”. Port pins retain their status from before wait mode is entered.

Wait mode is cancelled by a hardware reset or an interrupt. If an interrupt is used to cancel wait mode, that interrupt must first have been enabled, and the priority level of the interrupts which are not used to cancel must have been changed to 0. If returning by an interrupt, the clock in which the WAIT instruction executed is set to BCLK by the microcomputer, and the action is resumed from the interrupt routine. If only a hardware reset or an NMI interrupt is used to cancel wait mode, change the priority level of all interrupts to 0, then shift to wait mode.

Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 1.8.3 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

When reset, the device starts in division by 8 mode. The main clock division select bit 0 (bit 6 at address 0006_{16}) and the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}) change to "1" when shifting from high-speed/medium-speed mode to stop mode, shifting to low power dissipation mode and at a reset. When shifting from high-speed/medium-speed mode to low-speed mode, the value before high-speed/medium-speed mode is retained. The following shows the operational modes of BCLK.

(1) Division by 2 mode

The main clock is divided by 2 to obtain the BCLK.

(2) Division by 4 mode

The main clock is divided by 4 to obtain the BCLK.

(3) Division by 8 mode

The main clock is divided by 8 to obtain the BCLK. When reset, the device starts operating from this mode. Before the user can go from this mode to no division mode, division by 2 mode, or division by 4 mode, the main clock must be oscillating stably. When going to low-speed or lower power consumption mode, make sure the sub-clock is oscillating stably.

(4) Division by 16 mode

The main clock is divided by 16 to obtain the BCLK.

(5) No-division mode

The main clock is divided by 1 to obtain the BCLK.

(6) Low-speed mode

f_C is used as the BCLK. Note that oscillation of both the main and sub-clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub-clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

(7) Low power dissipation mode

f_C is the BCLK and the main clock is stopped.

Table 1.8.3. Operating modes dictated by settings of system clock control registers 0, 1, and 2

Modes		CM2 register	CM1 register		CM0 register			
		CM21	CM11	CM17, CM16	CM07	CM06	CM05	CM04
High-speed mode		0	0	00_2	0	0	0	—
Midium-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(Note 1)	1(Note 1)	1
Ring oscillator mode	divided by 1	1	—	0 0_2	0	0	0	—
	divided by 2	1	—	0 1_2	0	0	0	—
	divided by 4	1	—	1 0_2	0	0	0	—
	divided by 8	1	—	0 —		1	0	—
	divided by 16	1	—	1 1_2	0	0	0	—
Ring oscillator low power dissipation mode		1	—	— (Note 2)	0	(Note 2)	1	—

Note 1: When set the CM05 bit to “1” (main clock turned off), CM06 bit become “1” (divided by 8 mode).
Note 2: The divide-by-n value can be selected the same way as in ring oscillator mode.

Power control

The following is a description of the three available power control modes:

Modes

Power control is available in three modes.

(a) Normal operation mode

- **High-speed mode**

  Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the BCLK. Each peripheral function operates according to its assigned clock.

- **Medium-speed mode**

  Divide-by-2, divide-by-4, divide-by-8, or divide-by-16 frequency of the main clock becomes the BCLK. The CPU operates with the BCLK. Each peripheral function operates according to its assigned clock.

- **Low-speed mode**

  fC becomes the BCLK. The CPU operates according to the fC clock. The fC clock is supplied by the subclock. Each peripheral function operates according to its assigned clock.

- **Low power dissipation mode**

  The main clock operating in low-speed mode is stopped. The CPU operates according to the fC clock. The fC clock is supplied by the subclock. The only peripheral functions that operate are those with the subclock selected as the count source.

- **Ring oscillator mode**

  The ring oscillator clock divided by 1 (undivided), 2, 4, 8, or 16 provides BCLK. The ring oscillator clock is also the clock source for the peripheral function clocks. If the sub-clock is on, fC32 can be used as the count source for timers A and B.

- **Ring oscillator low power dissipation mode**

  The main clock is turned off after being placed in ring oscillator mode. The CPU clock can be selected as in the ring oscillator mode. The ring oscillator clock is the clock source for the peripheral function clocks. If the sub-clock is on, fC32 can be used as the count source for timers A and B. When the operation mode is returned to the high and medium speed modes, set the CM06 bit to “1” (divided by 8 mode).

(b) Wait mode

The CPU operation is stopped. The oscillators do not stop.

(c) Stop mode

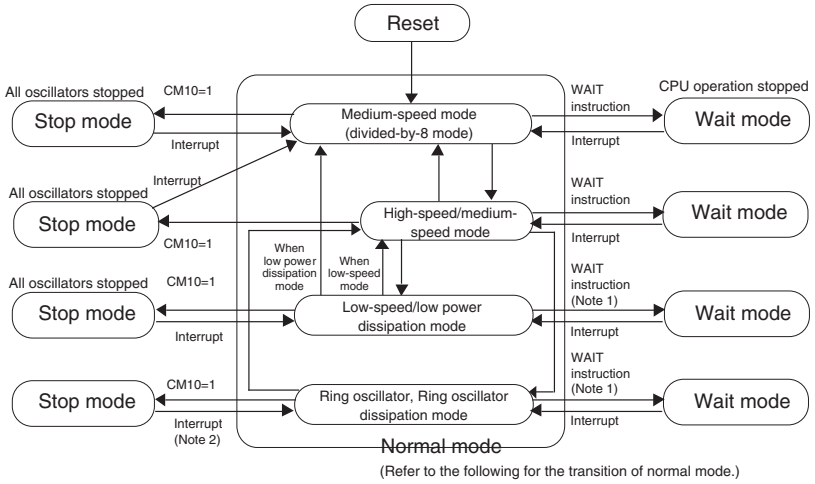
All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Table 1.8.4. Oscillation state in each mode

	XIN	XCIN	Ring
Medium-speed mode (divide-by-8 frequency after reset released)	BCLK	OFF	OFF
High/medium-speed	BCLK	ON/OFF	ON/OFF
Low-speed mode	ON	BCLK	OFF
Low power dissipation mode	OFF	BCLK	OFF
Ring oscillator mode	ON/OFF	ON/OFF	BCLK
Ring oscillator low power dissipation mode	OFF	ON/OFF	BCLK

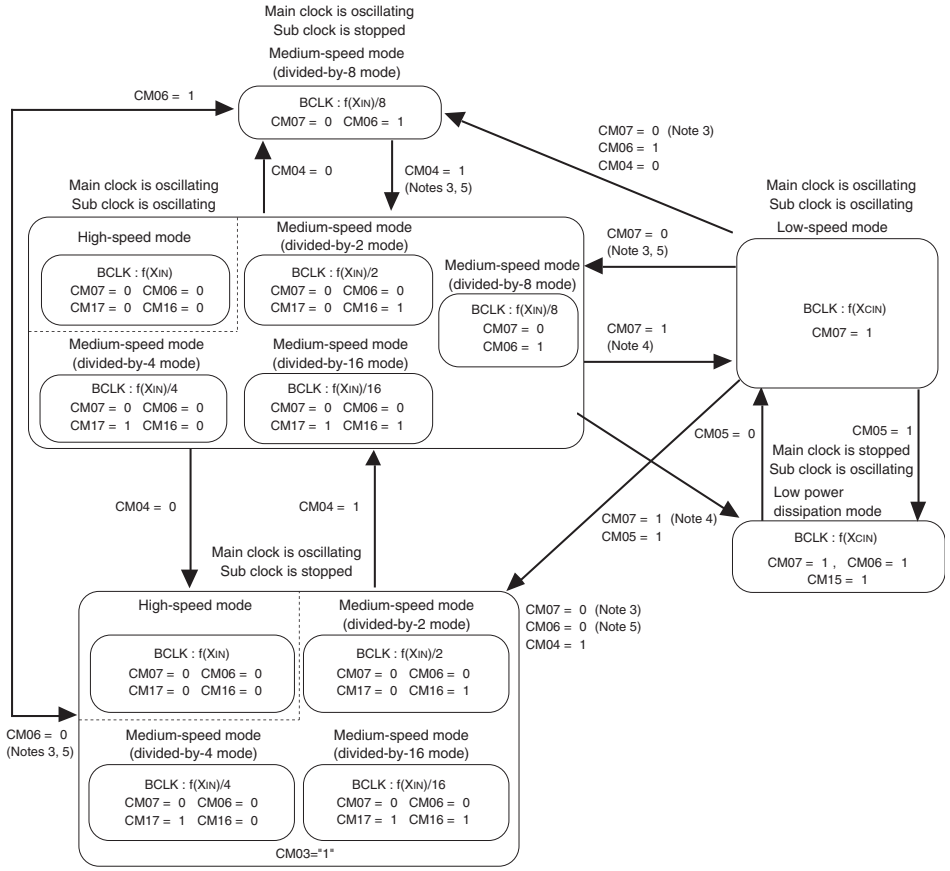
Figure 1.8.8 is the state transition diagram of the above modes.

Transition of stop mode, wait mode



(Refer to the following for the transition of normal mode.)

Transition of normal mode



Note 1: When bit PM21 = "0" (system clock protective function unused).
Note 2: The ring oscillator clock divided by 8 provides BCLK.
Note 3: Switch clock after oscillation of main clock is sufficiently stable.
Note 4: Switch clock after oscillation of sub clock is sufficiently stable.
Note 5: Change CM06 after changing CM17 and CM16.

Figure 1.8.8. State transition diagram of power control mode

Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure 1.8.9 shows the protect register. The values in the processor mode register 0 (address 0004_{16}), processor mode register 1 (address 0005_{16}), system clock control register 0 (address 0006_{16}), system clock control register 1 (address 0007_{16}), oscillation stop detection register (address $000C_{16}$), processor mode register 2 (001E16), peripheral clock select register (address $025E_{16}$), timer B2 special mode register ($039E_{16}$), port P9 direction register (address $03F3_{16}$), three-phase PWM control register 0 (address 0348_{16}) and three-phase PWM control register 1 (address 0349_{16}) can only be changed when the respective bit in the protect register is set to “1”. Therefore, important outputs can be allocated to port P9.

Protect register

b7	b6	b5	b4	b3	b2	b1	b0
×	×	×	0				

Symbol: PRCR
Address: $000A_{16}$
When reset: $XXX00000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16}), oscillation stop detection register (address $000C_{16}$), peripheral clock select register (address $025E_{16}$)	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0,1 and 2 (addresses 0004_{16},0005_{16} and $001E_{16}$), timer B2 special mode register (address $039E_{16}$), three-phase PWM control registers 0 and 1 (addresses 0348_{16} and 0349_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	Enables writing to port P9 direction register (address $03F3_{16}$) (note)	0 : Write-inhibited 1 : Write-enabled	○	○
PRC3	Enables writing to VCR2 (address $001A_{16}$) and D4INT (address $001F_{16}$)	0 : Write-inhibited 1 : Write-enabled	○	○
Reserved bit		Must always set to "0"	○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			—	—

Note: Writing a value to an address after 1 is written to this bit returns the bit to 0 . Other bits do not automatically return to 0 and they must therefore be reset by the program.

Figure 1.8.9. Protect register

Interrupts

Type of Interrupts

Figure 1.9.1 lists the types of interrupts.

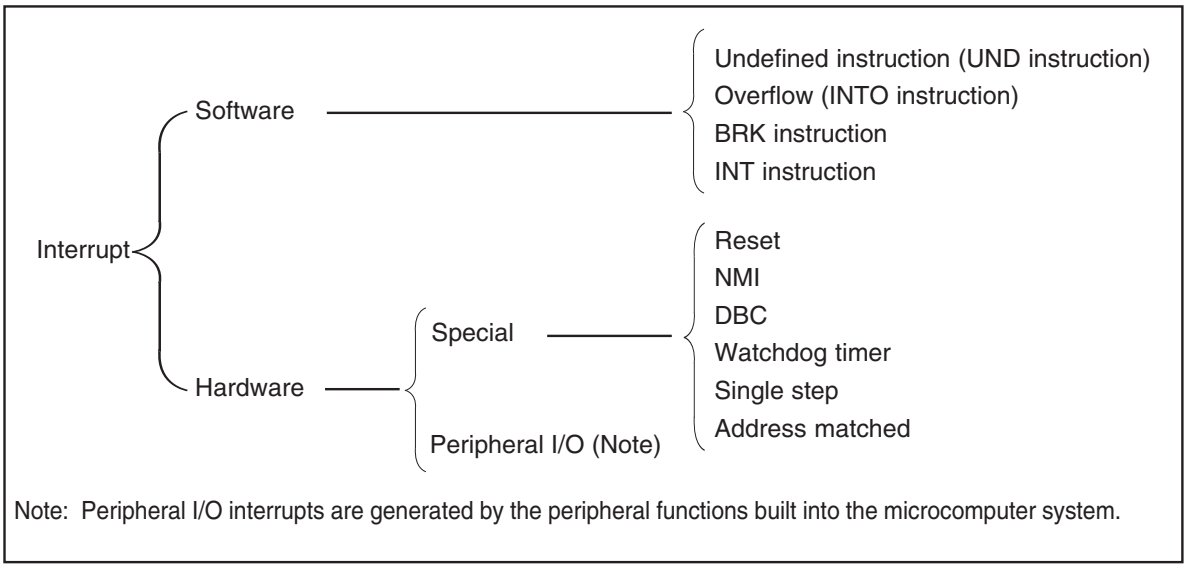


Figure 1.9.1. Classification of interrupts

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

Software Interrupts

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

- **Undefined instruction interrupt**
  An undefined instruction interrupt occurs when executing the UND instruction.
- **Overflow interrupt**
  An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to "1". The following are arithmetic instructions changes O flag:
  ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB
- **BRK interrupt**
  A BRK interrupt occurs when executing the BRK instruction.
- **INT interrupt**
  An INT interrupt occurs when specifying one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.
  The stack pointer (SP) used for the INT interrupt is dependent on the software interrupt number involved.
  For software interrupt numbers 0 through 31, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. The U flag is changed to "0" and the interrupt stack pointer (ISP) is selected, and then, the interrupt sequence is executed. When returning from the interrupt routine, the U flag is returned to the state it was before the interrupt request was accepted.
  For software numbers 32 through 63, the U flag is not changed, and so the stack pointer is not affected.

Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

(1) Special interrupts

Special interrupts are non-maskable interrupts.

- **Reset**

  Reset occurs if an “L” is input to the RESET pin.
- **NMI interrupt**

  If enabled, an NMI interrupt occurs if an “L” is input to the NMI pin.
- **DBC interrupt**

  This interrupt is exclusively for the debugger, do not use it in other circumstances.
- **Watchdog timer interrupt**

  Generated by the watchdog timer. Write to the watchdog timer start register after the watchdog timer interrupt occurs (initialize watchdog timer).
- **Single-step interrupt**

  This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to “1”, a single-step interrupt occurs after one instruction is executed.
- **Address match interrupt**

  An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1”. If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs.

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors too are dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **Bus collision detection interrupt**

  This is an interrupt that UART2 generates.
- **DMA0 interrupt, DMA1 interrupt**

  These are interrupts that DMA generates.
- **Key-input interrupt**

  A key-input interrupt occurs if an “L” is input to the KI pin.
- **A-D conversion interrupt**

  This is an interrupt that the A-D converter generates.
- **UART0, UART1, and UART2/NACK2 transmission interrupt**

  These are interrupts that the serial I/O transmission generates.
- **UART0, UART1, and UART2/ACK2 reception interrupt**

  These are interrupts that the serial I/O reception generates.
- **Timer A0 interrupt through timer A4 interrupt**

  These are interrupts that timer A generates
- **Timer B0 interrupt through timer B2 interrupt**

  These are interrupts that timer B generates.
- **INT0, INT1, INT3 through INT5 interrupt**

  An INT interrupt occurs if either a rising edge or a falling edge or a both edge is input to the INT pin.

Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. The first address of the interrupt routine is stored in the corresponding vector. Figure 1.9.2 shows the format for specifying the address.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.

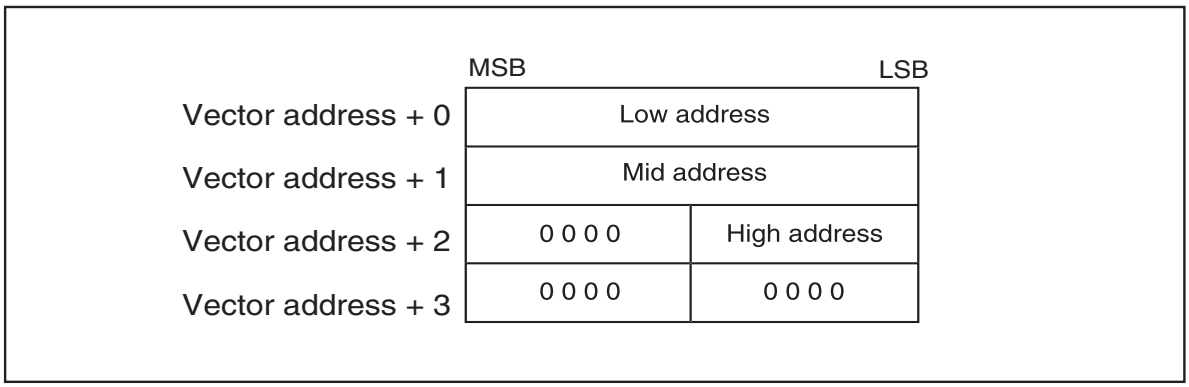


Figure 1.9.2. Format for specifying interrupt vector addresses

- **Fixed vector table**

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an address range extending from FFFDC16 to FFFFF16. A vector consists of four bytes. Set the first address of interrupt routine in the corresponding vector. Table 1.9.1 shows the interrupts assigned to the fixed vector table and the vector addresses.

Table 1.9.1. Interrupts assigned to the fixed vector tables and addresses of vector tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction
BRK instruction	FFFE416 to FFFE716	If the vector contains FF16, program execution starts from the address shown by the vector in the variable vector table.
Address match	FFFE816 to FFFEB16	There is an address-matching interrupt enable bit.
Single step (Note)	FFFEC16 to FFFEF16	Do not use.
Watchdog timer, Oscillation stop and re-oscillation detection, V_{DET4} detection	FFFF016 to FFFF316	
$\overline{DBC}$ (Note)	FFFF416 to FFFF716	Do not use.
$\overline{NMI}$	FFFF816 to FFFFB16	External interrupt by input to $\overline{NMI}$ pin.
Reset	FFFFC16 to FFFFF16	

Note: Interrupts used for debugging purposes only.

• **Variable vector tables**

The addresses in the variable vector table can be modified by user's settings. The first address of the variable vector table is specified by the user using the interrupt table register (INTB). The subsequent 256-byte addresses becomes the area for the variable vector tables. A vector consists of four bytes. Set the first address of the interrupt routine in the corresponding vector. Table 1.9.2 shows the interrupts assigned to the variable vector table and the vector addresses.

Table 1.9.2. Interrupts assigned to the variable vector tables and addresses of vector tables

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked I flag
Software interrupt number 4	+16 to +19 (Note 1)	$\overline{\text{INT3}}$	
Software interrupt number 5	+20 to +23 (Note 1)	Reserved	
Software interrupt number 6	+24 to +27 (Note 1)	Reserved	
Software interrupt number 7	+28 to +31 (Note 1)	Reserved	
Software interrupt number 8	+32 to +35 (Note 1)	$\overline{\text{INT5}}$	
Software interrupt number 9	+36 to +39 (Note 1)	$\overline{\text{INT4}}$	
Software interrupt number 10	+40 to +43 (Note 1)	UART 2 bus collision detection	
Software interrupt number 11	+44 to +47 (Note 1)	DMA0	
Software interrupt number 12	+48 to +51 (Note 1)	DMA1	
Software interrupt number 13	+52 to +55 (Note 1)	Key input interrupt	
Software interrupt number 14	+56 to +59 (Note 1)	A-D	
Software interrupt number 15	+60 to +63 (Note 1)	UART2 transmit/NACK2 (Note 2)	
Software interrupt number 16	+64 to +67 (Note 1)	UART2 receive/ACK2 (Note 2)	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note 1)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note 1)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note 1)	Timer A0	
Software interrupt number 22	+88 to +91 (Note 1)	Timer A1	
Software interrupt number 23	+92 to +95 (Note 1)	Timer A2	
Software interrupt number 24	+96 to +99 (Note 1)	Timer A3	
Software interrupt number 25	+100 to +103 (Note 1)	Timer A4	
Software interrupt number 26	+104 to +107 (Note 1)	Timer B0	
Software interrupt number 27	+108 to +111 (Note 1)	Timer B1	
Software interrupt number 28	+112 to +115 (Note 1)	Timer B2	
Software interrupt number 29	+116 to +119 (Note 1)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note 1)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note 1)	Reserved	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked I flag

Note 1: Address relative to address specified in interrupt table register (INTB).
Note 2: When IIC mode is selected, NACK and ACK interrupts are selected.

Interrupt Control

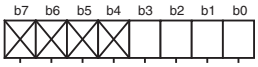
This section describes how to enable or disable maskable interrupts and how to set the priority to be accepted. The discussion here does not apply to non-maskable interrupts.

Maskable interrupts are enabled or disabled using the interrupt enable flag (I flag), the interrupt priority level selection bit, and the processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

Figure 1.9.3 shows the configuration and memory map of the interrupt control registers.

Interrupt control register (Note 2)

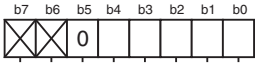
Symbol	Address	When reset
BCNIC	004A 16	XXXXX000 2
DMiIC(i=0, 1)	004B 16, 004C 16	XXXXX000 2
KUPIC	004D 16	XXXXX000 2
ADIC	004E 16	XXXXX000 2
SiTIC(i=0 to 2)	0051 16, 0053 16, 004F 16	XXXXX000 2
SiRIC(i=0 to 2)	0052 16, 0054 16, 0050 16	XXXXX000 2
TAiIC(i=0 to 4)	0055 16 to 0059 16	XXXXX000 2
TBiIC(i=0 to 2)	005A 16 to 005C 16	XXXXX000 2



Bit symbol	Bit name	Function	R	W
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	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note 1)
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			–	–

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. For details, see the precautions for interrupts.

Symbol	Address	When reset
INTiIC(i=3)	0044 16	XX00X000 2
INTiIC(i=4, 5)	0049 16, 0048 16	XX00X000 2
INTiIC(i=0, 1)	005D 16, 005E 16	XX00X000 2



Bit symbol	Bit name	Function	R	W
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	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	○	○ (Note 1)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○
Reserved bit		Must always be set to 0	○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			–	–

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. For details, see the precautions for interrupts.

Figure 1.9.3. Interrupt control registers

Interrupt Enable Flag (I flag)

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to "1" enables all maskable interrupts while setting it to "0" disables all maskable interrupts. This flag is set to "0" after reset.

Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to "0" disables the interrupt.
Table 1.9.3 shows the settings of interrupt priority levels and Table 1.9.4 shows the interrupt levels enabled, according to the contents of the IPL.

The following are conditions under which an interrupt is accepted:
· interrupt enable flag (I flag) = "1"
· interrupt request bit = "1"
· interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and so, not affected by one another.

Table 1.9.3. Settings of interrupt priority levels

Interrupt priority level select bit (b2 b1 b0)	Interrupt priority level	Priority order
0 0 0	Level 0 (interrupt disabled)	———
0 0 1	Level 1	Low ↓
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	High

Table 1.9.4. Interrupt levels enabled according to the contents of the IPL

IPL (IPL_2 IPL_1 IPL_0)	Enabled interrupt priority levels
0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

Rewrite the interrupt control register

To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    NOP                     ; Four NOP instructions are required when using HOLD function.
    NOP
    FSET    I               ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    MOV.W   MEM, R0         ; Dummy read.
    FSET    I               ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
    PUSHC   FLG             ; Push Flag register onto stack
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    POPC    FLG             ; Enable interrupts.
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

When changing an interrupt control register at a point where the interrupt is disabled, please read the following precautions on instructions used before changing the register.

(1) Changing a non-interrupt request bit (interrupt priority level)

If an interrupt request for an interrupt control register is generated during an instruction to rewrite the register is being executed, it is possible that the interrupt request bit is overwritten (reset to "0") and consequently the interrupt is ignored. This will depend on the instruction used. If this creates problems, use the instructions below to change the register.

Instructions : AND, OR, BCLR, BSET

(2) Changing the interrupt request bit

When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction used. If this creates problems, use the instructions below to change the register.

Instructions : MOV

Interrupt Sequence

This section describes an interrupt sequence — what are performed over a period from the time an interrupt is accepted until the time the interrupt routine is executed .

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.

In the interrupt sequence, the processor operates in the following sequence:

(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000_{16}. After this, the corresponding interrupt request bit becomes “0”.

(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.

(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to “0” (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)

(4) Saves the content of the temporary register (Note) within the CPU in the stack area.

(5) Saves the content of the program counter (PC) in the stack area.

(6) Sets the interrupt priority level of the accepted instruction in the IPL.

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

Note: This register cannot be utilized by the user.

Interrupt Response Time

'Interrupt response time' is the period between the time an interrupt occurs and the time the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 1.9.4 shows the interrupt response time.

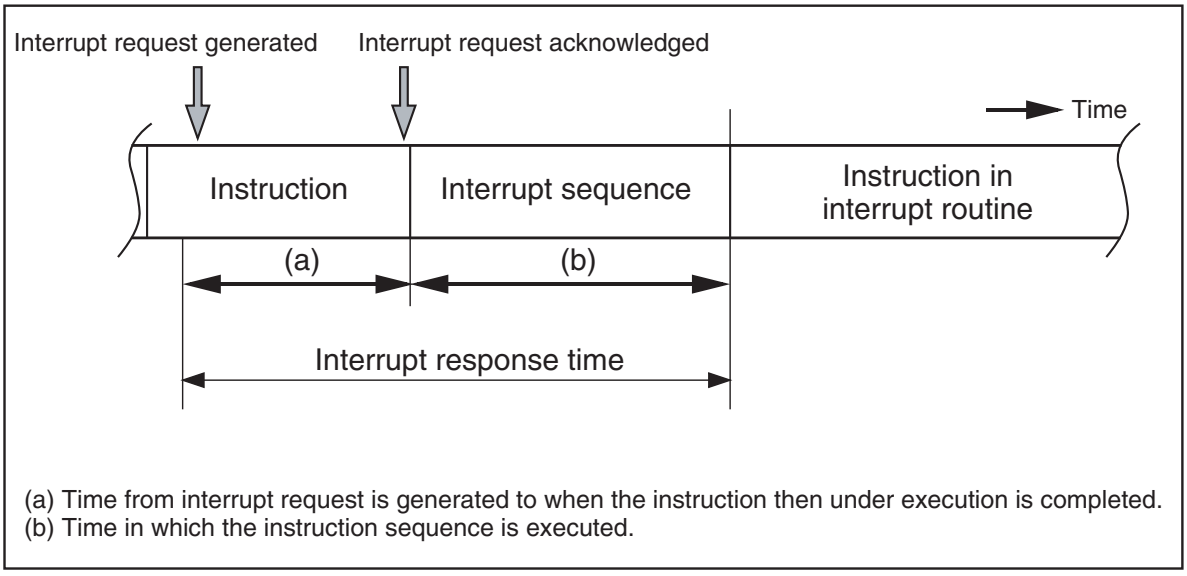


Figure 1.9.4. Interrupt response time

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction (without wait).

Time (b) is as shown in Table 1.9.5.

Table 1.9.5. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	6-Bit bus, without wait	8-Bit bus, without wait
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

Note 1: Add 2 cycles in the case of a DBC interrupt; add 1 cycle in the case either of an address match interrupt or of a single-step interrupt.

Note 2: Locate an interrupt vector address in an even address, if possible.

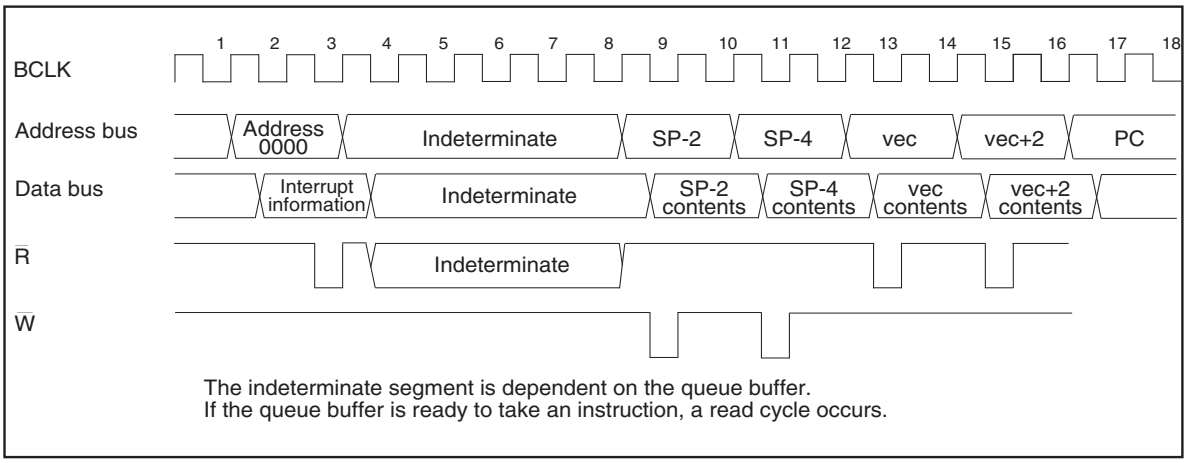


Figure 1.9.5. Time required for executing the interrupt sequence

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 1.9.6 is set in the IPL. Shown in Table 1.9.6 are the IPL values of software and special interrupts when they are accepted.

Table 1.9.6. Relationship between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Level that is set to IPL
Watchdog timer, $\overline{\text{NMI}}$, Oscillation stop and re-oscillation detection, V_{DET4} detection	7
Software, address match, $\overline{\text{DBC}}$, single-step	Not changed

Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the four higher-order bits of the program counter, and 4 upper-order bits and 8 lower-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 lower-order bits of the program counter. Figure 1.9.6 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

To save other necessary registers, use the PUSHM instruction to save these registers except the stack pointer (SP)at the beginning of the interrupt routine.

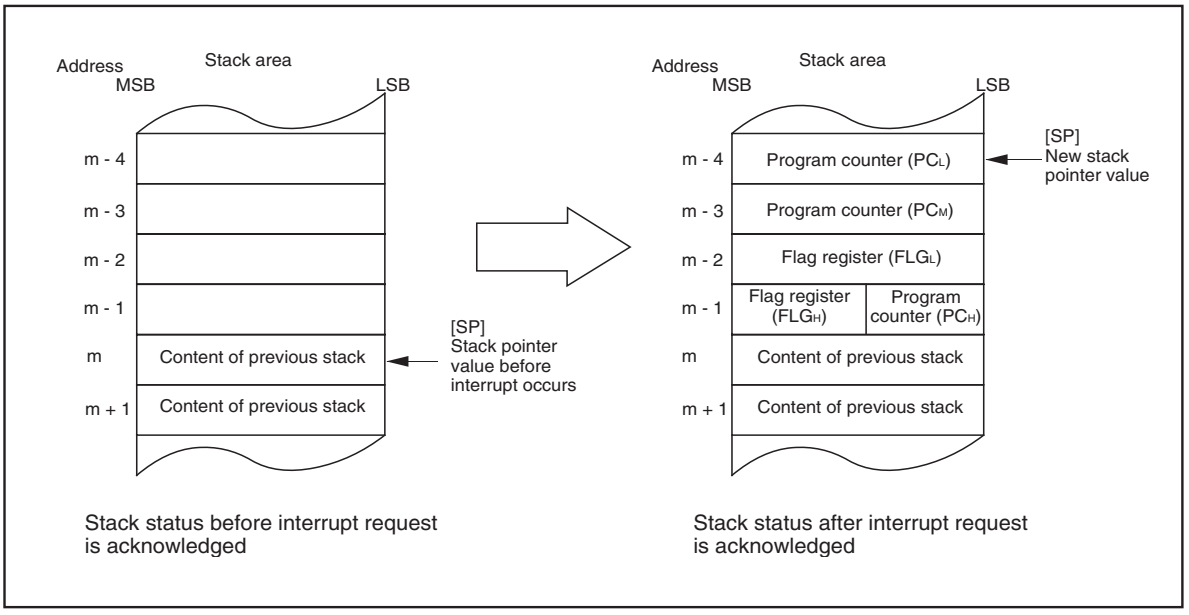
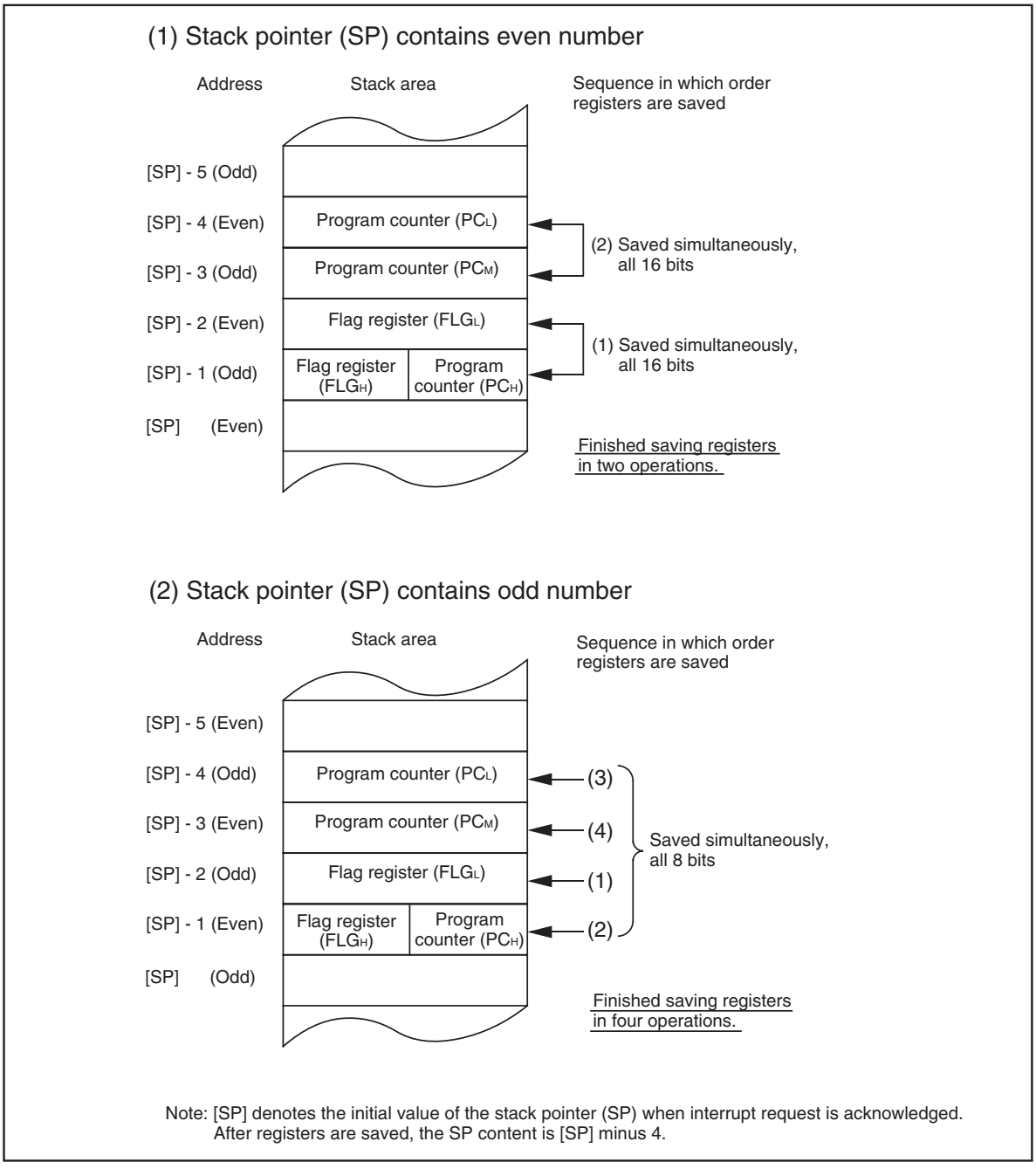


Figure 1.9.6. State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer (Note) , at the time of acceptance of an interrupt request, is even or odd. If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure 1.9.7 shows the operation of the saving registers.

Note: When any INT instruction in software numbers 32 to 63 has been executed, this is the stack pointer indicated by the U flag. Otherwise, it is the interrupt stack pointer (ISP).



Note: [SP] denotes the initial value of the stack pointer (SP) when interrupt request is acknowledged. After registers are saved, the SP content is [SP] minus 4.

Figure 1.9.7. Operation of saving registers

Returning from an Interrupt Routine

Executing a REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes.

To return the other registers saved by software within the interrupt routine, use the POPM or similar instruction before executing the REIT instruction.

Interrupt Priority

If there are two or more interrupt requests generated at the same time, the interrupt assigned with a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned with a higher hardware priority is accepted.

Priorities for the special interrupts, such as reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are controlled by hardware.

Figure 1.9.8 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.

Reset > NMI > DBC > Watchdog timer > Peripheral I/O > Single step > Address match

Figure 1.9.8. Hardware interrupts priorities

Interrupt resolution circuit

When two or more interrupts are generated simultaneously, the circuit in Figure 1.9.9 selects the interrupt based on the priority level shown.

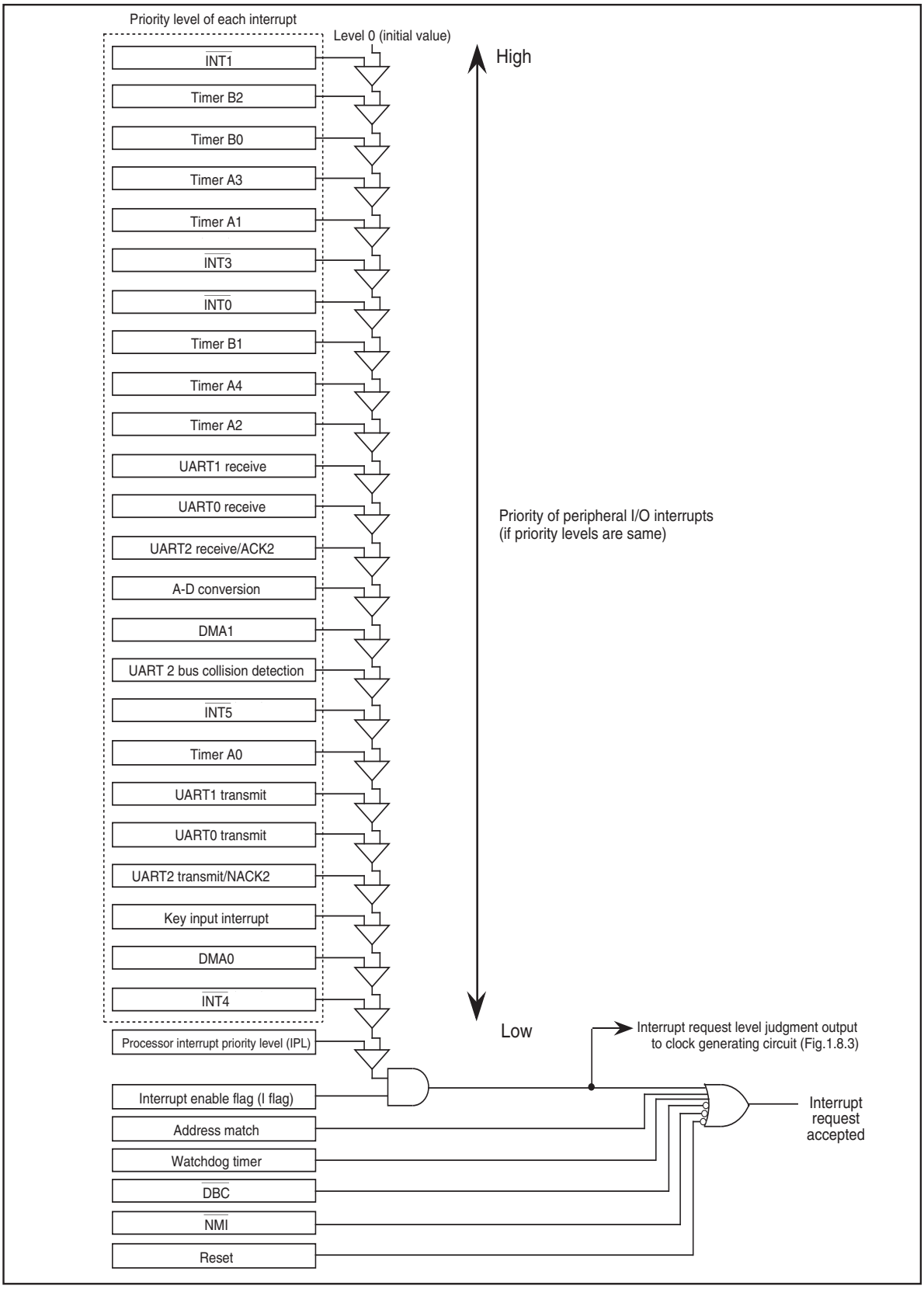


Figure 1.9.9. Maskable interrupts priorities (peripheral I/O interrupts)

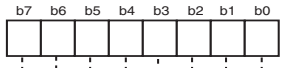
INT Interrupt

INT0, INT1, INT3 to INT5 are triggered by the edges of external inputs. The edge polarity of the interrupt is selected using the polarity select bit in the interrupt control registers, 0044$_{16}$, 0048$_{16}$, 0049$_{16}$, 005D$_{16}$ and 005E$_{16}$.

To generate interrupts on both rising and falling edges, set the INTi interrupt polarity switching bit of the interrupt request cause select register (035F$_{16}$) to "1". Interrupt on both rising and falling edges will be generated regardless of the value set in the polarity select bit of the corresponding interrupt control register.

Figure 1.9.10 shows the Interrupt request cause select register.

Interrupt request cause select register



Symbol: IFSR | Address: 035F$_{16}$ | When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
IFSR0	INT0 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR1	INT1 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR2	Reserved		—	—
IFSR3	INT3 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR4	INT4 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR5	INT5 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR6	Interrupt request cause select bit	0 : Reserved 1 : INT4 (note)	○	○
IFSR7	Interrupt request cause select bit	0 : Reserved 1 : INT5 (note)	○	○

Note: To use INTi (i=4,5) functionality, this bit must be set to a 1.

Figure 1.9.10. Interrupt request cause select register

NMI Interrupt

If enabled, an NMI interrupt is generated when the input to the P85/NMI pin changes from “H” to “L”. The NMI interrupt is a non-maskable external interrupt. The pin level can be checked in the port P85 register (bit 5 at address 03F016).

NMI is disabled by default after reset (the pin is a GPIO pin, P85) and can be enabled using bit 4 of processor mode register 2. Once enabled, it can only be disabled by a reset signal.

Key Input Interrupt

If the direction register of any of P104 to P107 is set for input and a falling edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. However, if you intend to use the key input interrupt, do not use P104 to P107 as A-D input ports. Figure 1.9.11 shows the block diagram of the key input interrupt. Note that if an “L” level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

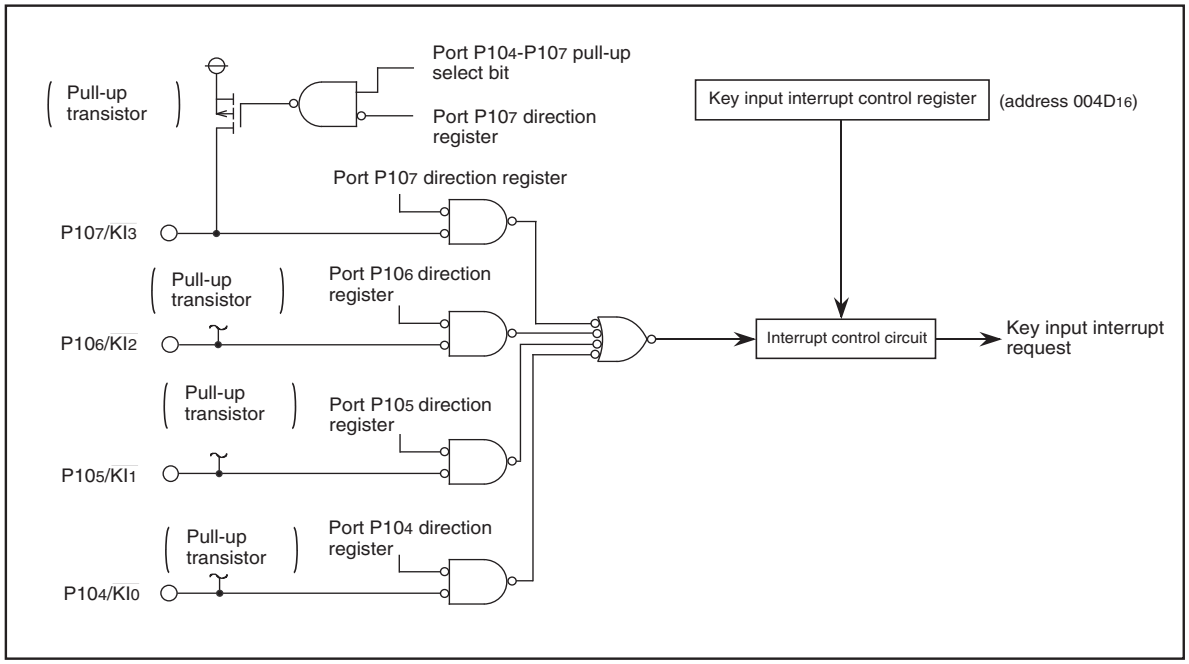


Figure 1.9.11. Block diagram of key input interrupt

Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Four address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). For an address match interrupt, the value of the program counter (PC) that is saved to the stack area varies depending on the instruction being executed. Note that when using the external data bus in width of 8 bits, the address match interrupt cannot be used for external area.

Figure 1.9.12 shows the address match interrupt-related registers.

Address match interrupt enable register

Symbol	Address	When reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminated.			—	—

Address match interrupt register i (i = 0, 1)

Symbol	Address	When reset
RMAD0	0012_{16} to 0010_{16}	$X0000_{16}$
RMAD1	0016_{16} to 0014_{16}	$X0000_{16}$

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminated.		—	—

Figure 1.9.12. Address match interrupt-related registers

Precautions for Interrupts

(1) Reading address 0000016

- Do not read address 0000016 by software.

When a maskable interrupt is generated, the CPU reads the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence. The interrupt request bit of this interrupt is written in address 0000016. Therefore, reading 0000016 may cancel the interrupt or generate an unexpected one.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000016. Accepting an interrupt before setting a value in the stack pointer may cause a runaway program. Be sure to set a value in the stack pointer before accepting an interrupt.

When using the NMI interrupt, initialize the stack pointer at the beginning of the program. Concerning the first instruction immediately after reset, the generation of any interrupts, including the NMI interrupt, is prohibited.

(3) The NMI interrupt

- The NMI interrupt is enabled or disabled in bit4 of the processor mode register 2. It is disabled by default (the pin is used as P85)after reset. Once enabled, it stays enabled until a reset is applied.
- The NMI interrupt input level can be determined by reading the contents of the P8 register.
- If NMI is enabled, do not attempt to go into stop mode with the NMI input in the “L” state. With the NMI input in the “L” state, the CM10 is fixed to “0”, therefore, any attempt to go into stop mode is turned down.
- If NMI is enabled, going into wait mode with the NMI input in the “L” state does not reduce the power consumption. With the NMI input in the “L” state, the CPU stops but the oscillation is not stop, so power consumption is not reduced. Having NMI in "L" state when going into a wait mode may also cause an unpredictable behavior of the program.
- Signal input to the NMI pin require an “L” level of '2 cycles of BCLK + 300ns' or more.

(4) External interrupt

- Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to pins INT0 , INT1, INT3 through INT5 regardless of the CPU operation clock.

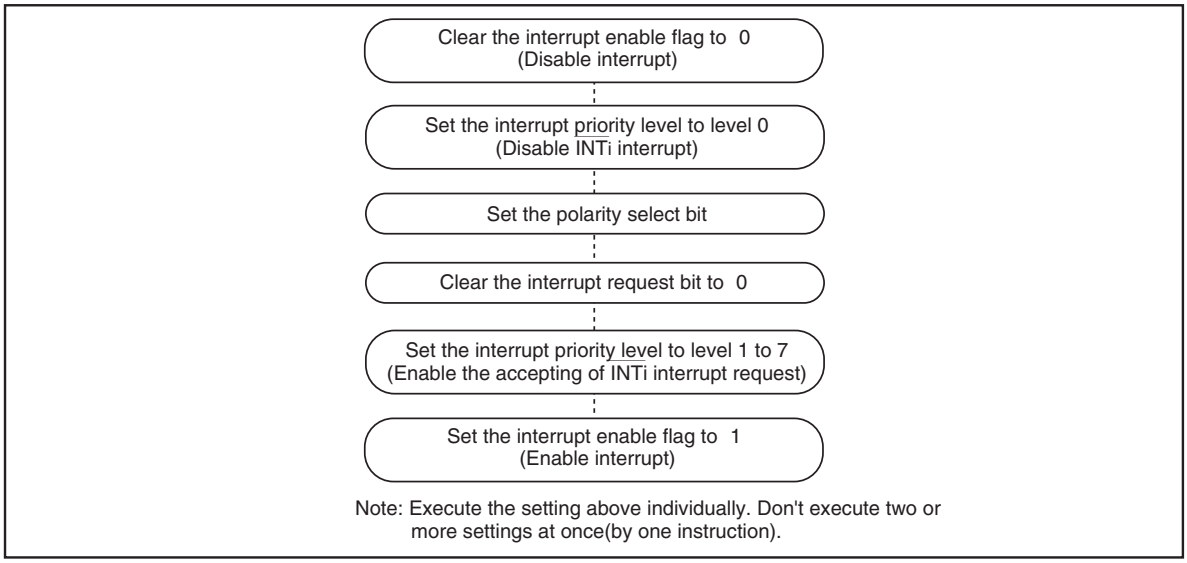


Figure 1.9.13. Switching condition of INT interrupt request

• When the polarity of the INT_0 , INT_1, INT_3 through INT_5 pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 1.9.13 shows the procedure for changing the INT interrupt generate factor.

(5) Watchdog timer interrupt

• Write to the watchdog timer start register after the watchdog timer interrupt occurs (initialize watchdog timer).

(6) Rewrite the interrupt control register

Example 1:

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    NOP                     ; Four NOP instructions are required when using HOLD function.
    NOP
    FSET    I               ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    MOV.W   MEM, R0         ; Dummy read.
    FSET    I               ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
    PUSHC   FLG             ; Push Flag register onto stack
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    POPC    FLG             ; Enable interrupts.
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

• To rewrite the interrupt control register, do so at a point where an interrupt request for that register is not generated. If there is possibility of the interrupt request occur, disable the interrupt before rewriting the interrupt control register. Some program examples are described as follow:

When changing an interrupt control register with interrupts enabled, please read the following precautions on instructions used before changing the register.

(1) Changing a non-interrupt request bit

If an interrupt request for an interrupt control register is generated during an instruction to rewrite the register is being executed, there is a case that the interrupt request bit is not set and consequently the interrupt is ignored. This will depend on the instruction. If this creates problems, use the instructions below to change the register.

Instructions : AND, OR, BCLR, BSET

(2) Changing the interrupt request bit

When attempting to clear the interrupt request bit of an interrupt control register, the interrupt request bit is not cleared sometimes. This will depend on the instruction. If this creates problems, use the instructions below to change the register.

Instructions : MOV

Watchdog Timer

The watchdog timer has 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 is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. Whether a watchdog timer interrupt is generated or reset is selected when an underflow occurs in the watchdog timer. When the watchdog timer interrupt is selected, write to the watchdog timer start register after the watchdog timer interrupt occurs (initialize watchdog timer). Watchdog timer interrupt is selected when bit 2 (PM12) of the processor mode register 1 (address 0005_{16}) is "0" and reset is selected when PM12 is "1". No value other than "1" can be written in PM12. Once when reset is selected (PM12="1"), watchdog timer interrupt cannot be selected by software.

When X_{IN} is selected for the BCLK, bit 7 of the watchdog timer control register (address $000F_{16}$) selects the prescaler division ratio (by 16 or by 128). When X_{CIN} is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address $000F_{16}$). Thus the watchdog timer's period can be calculated as given below. The watchdog timer's period is, however, subject to an error due to the prescaler.

With X_{IN} chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{prescaler dividing ratio (16 or 128) X watchdog timer count (32768)}}{\text{BCLK}}$$

With X_{CIN} chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{prescaler dividing ratio (2) X watchdog timer count (32768)}}{\text{BCLK}}$$

For example, suppose that BCLK runs at 16 MHz and that 16 has been chosen for the dividing ratio of the prescaler, then the watchdog timer's period becomes approximately 32.8 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address $000E_{16}$) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address $000E_{16}$). In stop mode, wait mode and hold state, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.

Also PM12 is initialized only when reset. The watchdog timer interrupt is selected after reset is cancelled.

Figure 1.10.1 shows the block diagram of the watchdog timer. Figure 1.10.2 shows the watchdog timer-related registers.

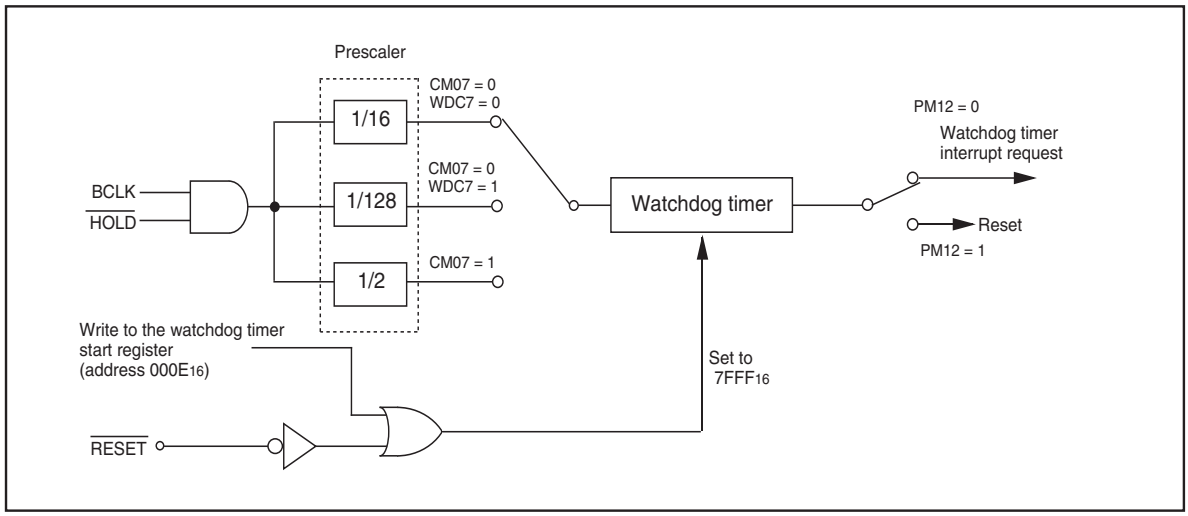


Figure 1.10.1. Block diagram of watchdog timer

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol: WDC
Address: 000F16
After reset: 00?XXXXX2(Note1)

Bit symbol	Bit name	Function	RW
	High-order bit of watchdog timer		RO
WDC5	Cold start / warm start discrimination flag (Note 2)	0 : Cold start 1 : Warm start	RW
—	Reserved bit	Set to 0	RW
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	RW

Note 1: The WDC5 bit is set to 0 immediately after power-on (cold start).
Note 2: The WDC5 bit will always be set to a 1 whenever the WDC SFR is written (regardless of value of WDC5 bit).

Watchdog timer start register

b7	b0

Symbol: WDTS
Address: 000E16
When reset: Indeterminate

Function	R	W
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to 7FFF16 regardless of whatever value is written.	X	O

Figure 1.10.2. Watchdog timer control and start registers

DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC shares the same data bus with the CPU. The DMAC is given a higher right of using the bus than the CPU, which leads to working the cycle stealing method. On this account, the operation from the occurrence of DMA transfer request signal to the completion of 1-word (16-bit) or 1-byte (8-bit) data transfer can be performed at high speed. Figure 1.11.1 shows the block diagram of the DMAC. Table 1.11.1 shows the DMAC specifications. Figures 1.11.2 to 1.11.4 show the registers used by the DMAC.

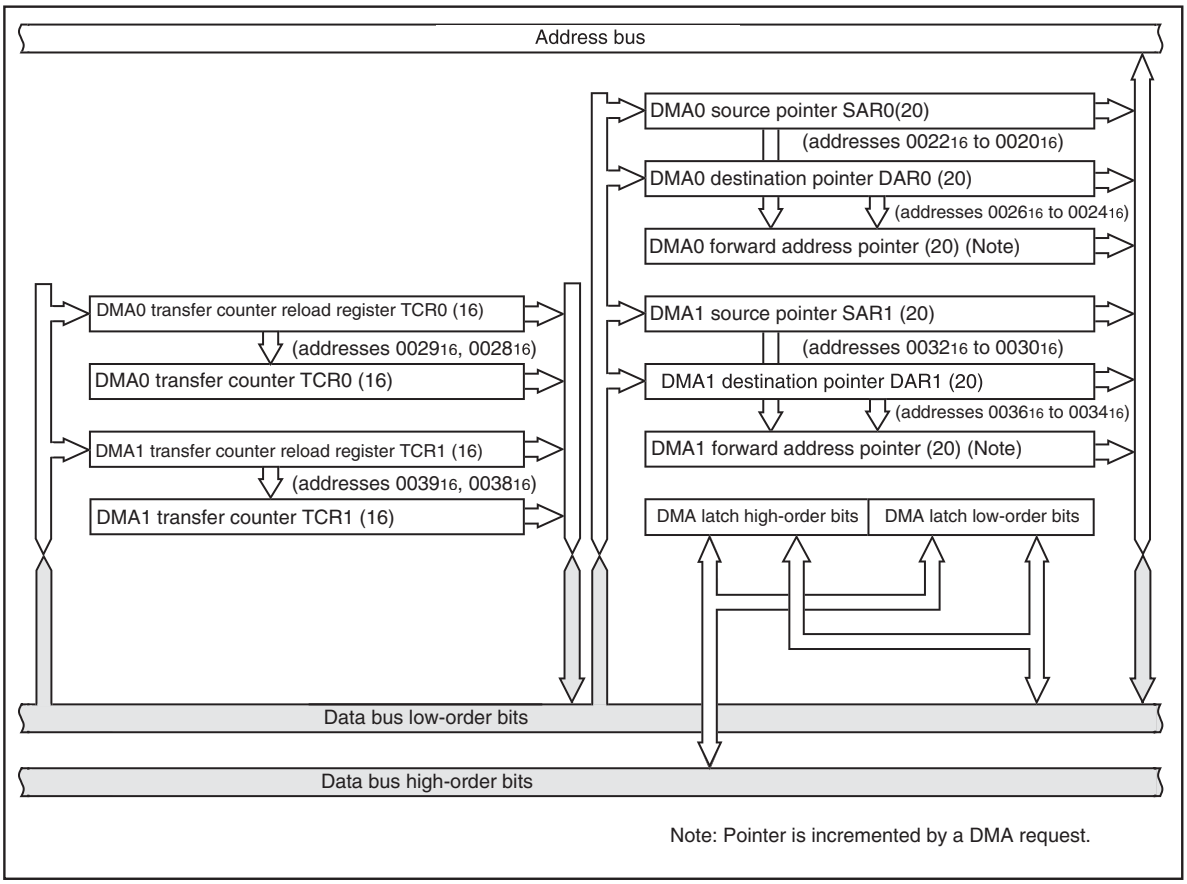


Figure 1.11.1. Block diagram of DMAC

Either a write to the software DMA request bit or an interrupt request factor are used as a DMA transfer request signal. The DMA transfer is not affected by the interrupt enable flag (I flag) nor by the interrupt priority level. The DMA transfer does not affect interrupts.

If the DMAC is active (the DMA enable bit is set to 1), data transfer starts every time a DMA transfer request signal occurs. If the cycle of the occurrences of DMA transfer request signals is higher than the DMA transfer cycle, there can be instances in which the number of transfer requests do not agree with the number of transfers. For details, see the description of the DMA request bit.

Table 1.11.1. 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 (Note that DMA-related registers [0020$_{16}$ to 003F$_{16}$] cannot be accessed)
Maximum No. of bytes transferred	128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge of INT0 or INT1 or both edge Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transfer and reception interrupt requests UART1 transfer and reception interrupt requests UART2 transfer and reception interrupt requests A-D conversion interrupt requests Software triggers
Channel priority	DMA0 takes precedence if DMA0 and DMA1 requests are generated simultaneously
Transfer unit	8 bits or 16 bits
Transfer address direction	forward/fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	• Single transfer mode After the transfer counter underflows, the DMA enable bit turns to “0”, and the DMAC turns inactive • Repeat transfer mode After the transfer counter underflows, the value of the transfer counter reload register is reloaded to the transfer counter. The DMAC remains active unless a “0” is written to the DMA enable bit.
DMA interrupt request generation timing	When an underflow occurs in the transfer counter
Active	When the DMA enable bit is set to “1”, the DMAC is active. When the DMAC is active, data transfer starts every time a DMA transfer request signal occurs.
Inactive	• When the DMA enable bit is set to “0”, the DMAC is inactive. • After the transfer counter underflows in single transfer mode
	At the time of starting data transfer immediately after turning the DMAC active, the value of one of source pointer and destination pointer - the one specified for the forward direction- is reloaded to the forward direction address pointer, and the value of the transfer counter reload register is reloaded to the transfer counter.
Writing to register	Registers specified for forward direction transfer are always write enabled. Registers specified for fixed address transfer are write-enabled when the DMA enable bit is “0”.
Reading the register	Can be read at any time. However, when the DMA enable bit is “1”, reading the register set up as the forward register is the same as reading the value of the forward address pointer.

Note: DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level.

DMA0 request cause select register

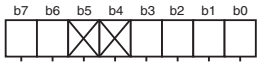
b7	b6	b5	b4	b3	b2	b1	b0
		X	X				

Symbol: DM0SL
Address: $03B8_{16}$
When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b3 b2 b1 b0 0 0 0 0 : Falling edge of $\overline{INT0}$ pin 0 0 0 1 : Software trigger 0 0 1 0 : Timer A0 0 0 1 1 : Timer A1 0 1 0 0 : Timer A2 0 1 0 1 : Timer A3 0 1 1 0 : Timer A4 (DMS=0) /two edges of $\overline{INT0}$ pin (DMS=1) 0 1 1 1 : Timer B0 (DMS=0) 1 0 0 0 : Timer B1 (DMS=0) 1 0 0 1 : Timer B2 (DMS=0) 1 0 1 0 : UART0 transmit 1 0 1 1 : UART0 receive 1 1 0 0 : UART2 transmit 1 1 0 1 : UART2 receive 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 transmit	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .			–	–
DMS	DMA request cause expansion select bit	0 : Normal 1 : Expanded cause	○	○
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to 1 (When read, the value of this bit is always 0)	○	○

Figure 1.11.2. DMAC register (1)

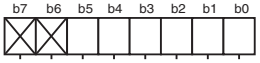
DMA1 request cause select register



Symbol: DM1SL | Address: 03BA16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b3 b2 b1 b0 0 0 0 0 : Falling edge of INT1 pin 0 0 0 1 : Software trigger 0 0 1 0 : Timer A0 0 0 1 1 : Timer A1 0 1 0 0 : Timer A2 0 1 0 1 : Timer A3 (DMS=0) 0 1 1 0 : Timer A4 (DMS=0) 0 1 1 1 : Timer B0 (DMS=0) /two edges of INT1 (DMS=1) 1 0 0 0 : Timer B1 1 0 0 1 : Timer B2 1 0 1 0 : UART0 transmit 1 0 1 1 : UART0 receive 1 1 0 0 : UART2 transmit 1 1 0 1 : UART2 receive/ACK2 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 receive	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .			–	–
DMS	DMA request cause expansion select bit	0 : Normal 1 : Expanded cause	○	○
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to 1 (When read, the value of this bit is always 0)	○	○

DMAi control register



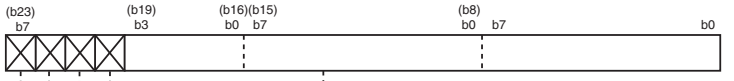
Symbol: DMiCON(i=0,1) | Address: 002C16, 003C16 | When reset: 00000X002

Bit symbol	Bit name	F unction	R	W
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	○	○
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	○	○
DMAS	DMA request bit (Note 1)	0 : DMA not requested 1 : DMA requested	○	○ (Note 2)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	○	○
DSD	Source address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
DAD	Destination address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .			–	–

Note 1: DMA request can be cleared by resetting the bit.
Note 2: This bit can only be set to 0 .
Note 3: Source address direction select bit and destination address direction select bit cannot be set to 1 simultaneously.

Figure 1.11.3. DMAC register (2)

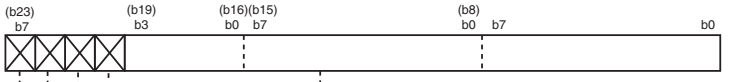
DMAi source pointer (i = 0, 1)



Symbol	Address	When reset
SAR0	0022$_{16}$ to 0020$_{16}$	Indeterminate
SAR1	0032$_{16}$ to 0030$_{16}$	Indeterminate

Function	Transfer address specification	R	W
Source pointer Stores the source address	00000$_{16}$ to FFFFF$_{16}$	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.		—	—

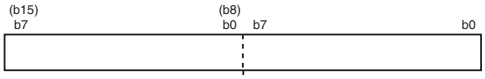
DMAi destination pointer (i = 0, 1)



Symbol	Address	When reset
DAR0	0026$_{16}$ to 0024$_{16}$	Indeterminate
DAR1	0036$_{16}$ to 0034$_{16}$	Indeterminate

Function	Transfer address specification	R	W
Destination pointer Stores the destination address	00000$_{16}$ to FFFFF$_{16}$	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.		—	—

DMAi transfer counter (i = 0, 1)



Symbol	Address	When reset
TCR0	0029$_{16}$, 0028$_{16}$	Indeterminate
TCR1	0039$_{16}$, 0038$_{16}$	Indeterminate

Function	Transfer count specification	R	W
Transfer counter Set a value one less than the transfer count	0000$_{16}$ to FFFF$_{16}$	○	○

Figure 1.11.4. DMAC register (3)

(1) Transfer cycle

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses. Also, the bus cycle itself is longer when software waits are inserted.

(a) Effect of source and destination addresses

When 16-bit data is transferred on a 16-bit data bus, and the source and destination both start at odd addresses, there are one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

(b) Effect of software wait

When the SFR area or a memory area with a software wait is accessed, the number of cycles is increased for the wait by 1 bus cycle. The length of the cycle is determined by BCLK.

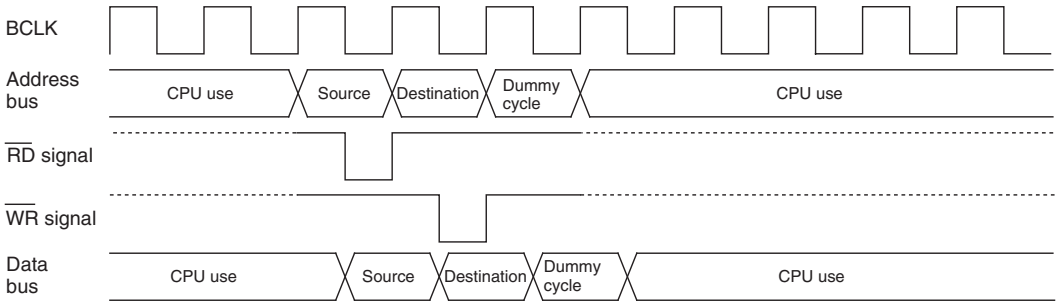
Figure 1.11.5 shows the example of the transfer cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle. For example (2) in Figure 1.11.5, if 16-bit data is being transferred from an odd source address to an odd destination address, two bus cycles are required for both the source read cycle and the destination write cycle.

(2) DMAC transfer cycles

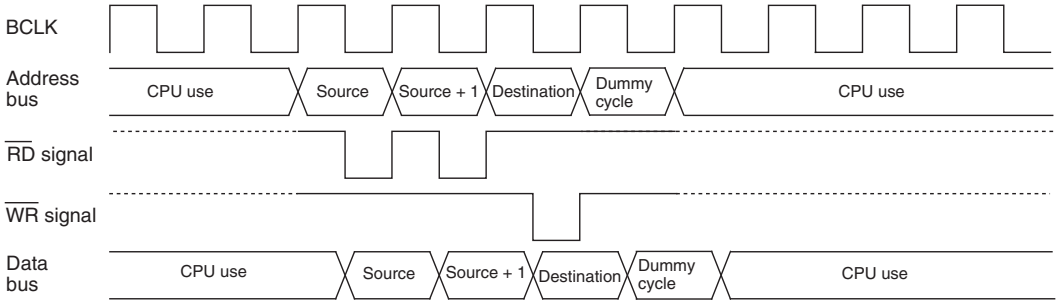
Any combination of even or odd transfer read and write addresses is possible. Table 1.11.2 shows the number of DMAC transfer cycles.
The number of DMAC transfer cycles can be calculated as follows:

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

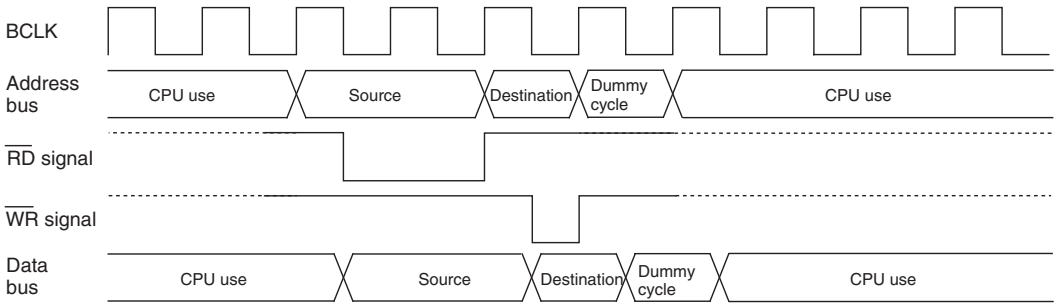
(1) 8-bit transfers
16-bit transfers and the source address is even.



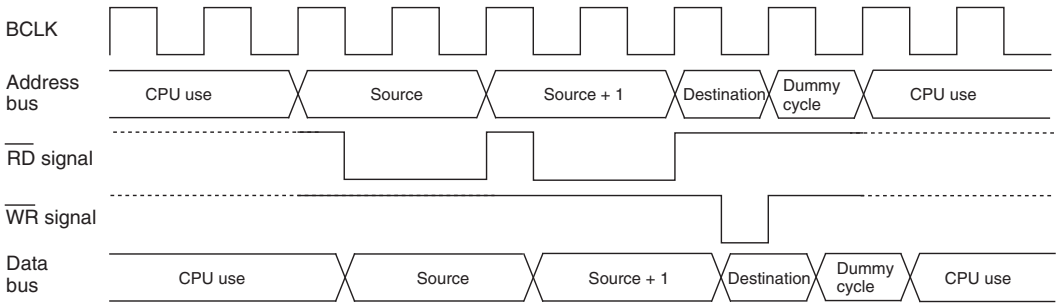
(2) 16-bit transfers and the source address is odd



(3) One wait is inserted into the source read under the conditions in (1)



(4) One wait is inserted into the source read under the conditions in (2)



Note: The same timing changes occur with the respective conditions at the destination as at the source

Figure 1.11.5. Example of the transfer cycles for a source read

Table 1.11.2. No. of DMAC transfer cycles

			Single Chip Mode	
Transfer Unit	Bus Width	Access Address	No. of Read Cycles	No. of Write Cycles
8-bit transfers (DMBIT="1")	16 bit	Even	1	1
		Odd	1	1
16-bit transfers (DMBIT="0")	16 bit	Even	1	1
		Odd	2	2

Coefficient j, k

Internal Memory		
Internal ROM/RAM No Wait	Internal ROM/RAM With Wait	SFR Area
1	2	3

DMA enable bit

Setting the DMA enable bit to “1” makes the DMAC active. The DMAC carries out the following operations at the time data transfer starts immediately after DMAC is turned active.

(1) Reloads the value of one of the source pointer and the destination pointer - the one specified for the forward direction - to the forward direction address pointer.

(2) Reloads the value of the transfer counter reload register to the transfer counter.

Thus overwriting “1” to the DMA enable bit with the DMAC being active carries out the operations given above, so the DMAC operates again from the initial state at the instant “1” is overwritten to the DMA enable bit.

DMA request bit

The DMAC can generate a DMA transfer request signal triggered by a factor chosen in advance out of DMA request factors for each channel.

DMA request factors include the following.

* Factors effected by using the interrupt request signals from the built-in peripheral functions and software DMA factors (internal factors) effected by a program.
* External factors effected by utilizing the input from external interrupt signals.

For the selection of DMA request factors, see the descriptions of the DMAi factor selection register.

The DMA request bit turns to “1” if the DMA transfer request signal occurs regardless of the DMAC's state (regardless of whether the DMA enable bit is set to “1” or “0”). It turns to “0” immediately before data transfer starts.

In addition, it can be set to “0” by use of a program, but cannot be set to “1”.

There can be instances in which a change in DMA request factor selection bit causes the DMA request bit to turn to “1”. So be sure to set the DMA request bit to “0” after the DMA request factor selection bit is changed.

The DMA request bit turns to “1” if a DMA transfer request signal occurs, and turns to “0” immediately before data transfer starts. If the DMAC is active, data transfer starts immediately, so the value of the DMA request bit, if read by use of a program, turns out to be “0” in most cases. To examine whether the DMAC is active, read the DMA enable bit.

Here follows the timing of changes in the DMA request bit.

(1) Internal factors

Except the DMA request factors triggered by software, the timing for the DMA request bit to turn to “1” due to an internal factor is the same as the timing for the interrupt request bit of the interrupt control register to turn to “1” due to several factors.

Turning the DMA request bit to “0” due to an internal factor is timed to be effected immediately before the transfer starts.

(2) External factors

An external factor is a factor caused to occur by the leading edge of input from the INTi pin (i depends on which DMAC channel is used).

Selecting the INTi pins as external factors using the DMA request factor selection bit causes input from these pins to become the DMA transfer request signals.

The timing for the DMA request bit to turn to “1” when an external factor is selected synchronizes with the signal's edge applicable to the function specified by the DMA request factor selection bit (synchronizes with the trailing edge of the input signal to each INTi pin, for example).

With an external factor selected, the DMA request bit is timed to turn to “0” immediately before data transfer starts similarly to the state in which an internal factor is selected.

(3) The priorities of channels and DMA transfer timing

If a DMA transfer request signal falls on a single sampling cycle (a sampling cycle means one period from the leading edge to the trailing edge of BCLK), the DMA request bits of applicable channels concurrently turn to "1". If the channels are active at that moment, DMA0 is given a high priority to start data transfer. When DMA0 finishes data transfer, it gives the bus right to the CPU. When the CPU finishes single bus access, then DMA1 starts data transfer and gives the bus right to the CPU.

An example in which DMA transfer is carried out in minimum cycles at the time when DMA transfer request signals due to external factors concurrently occur.

Figure 1.11.6 shows the DMA transfer effected by external factors.

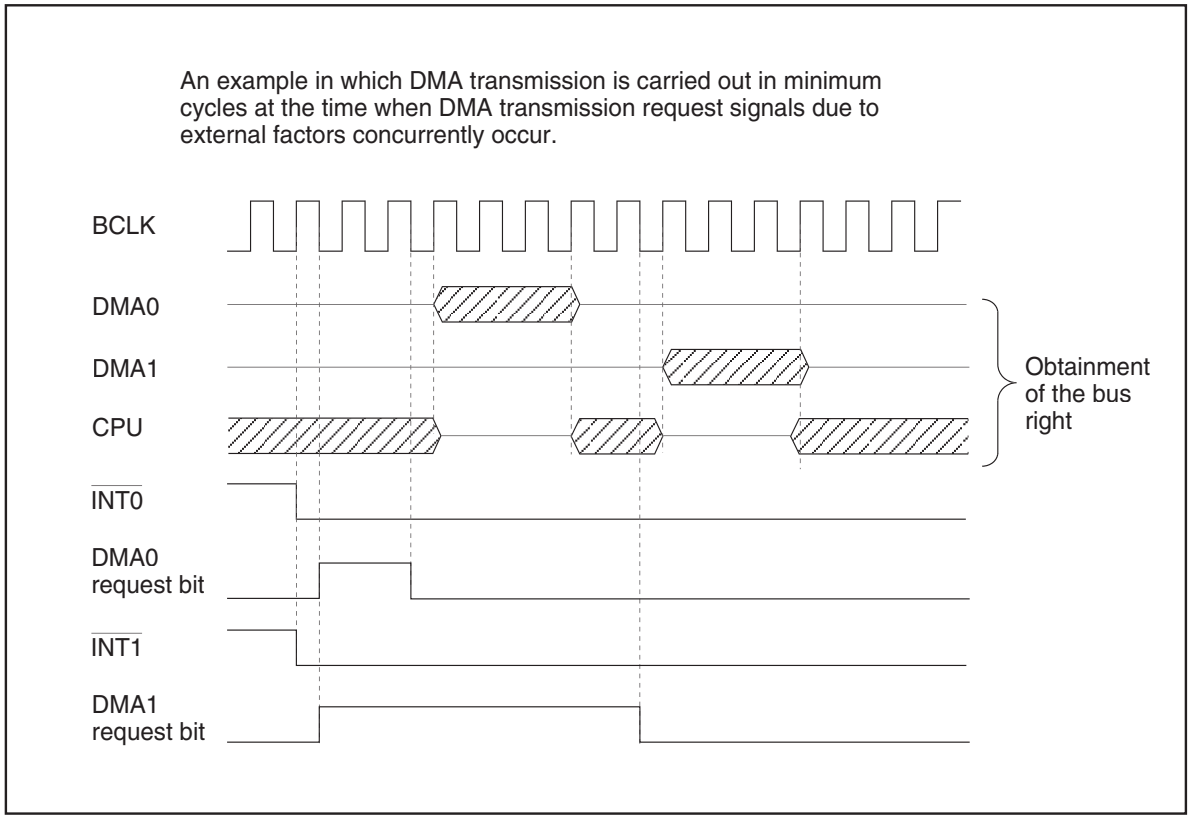


Figure 1.11.6. An example of DMA transfer effected by external factors

Timers

There are eight 16-bit timers. These timers can be classified by function into timers A (five) and timers B (three). All these timers function independently. Figures 1.12.1 and 1.12.2 show the block diagram of timers.

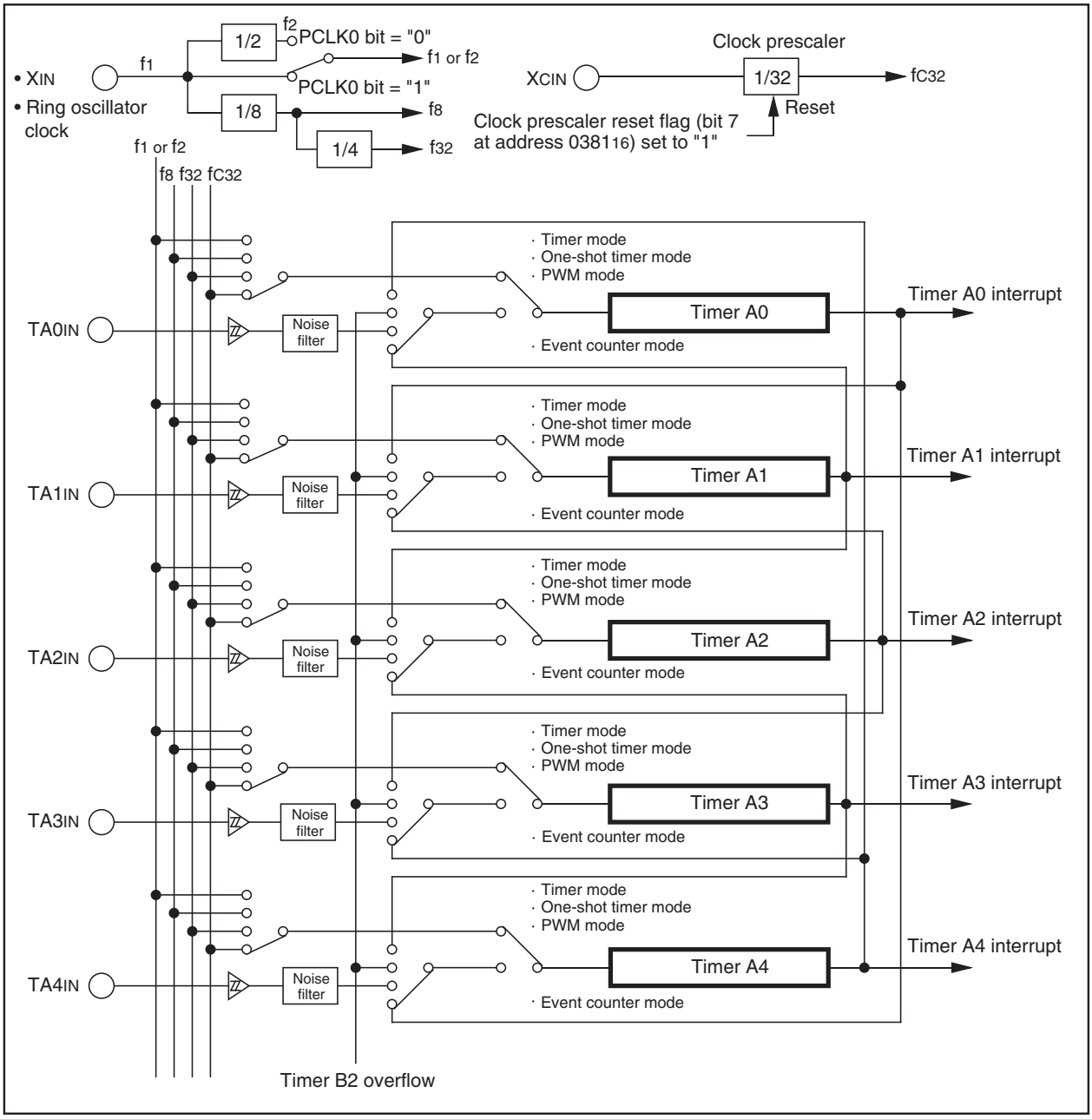


Figure 1.12.1. Timer A block diagram

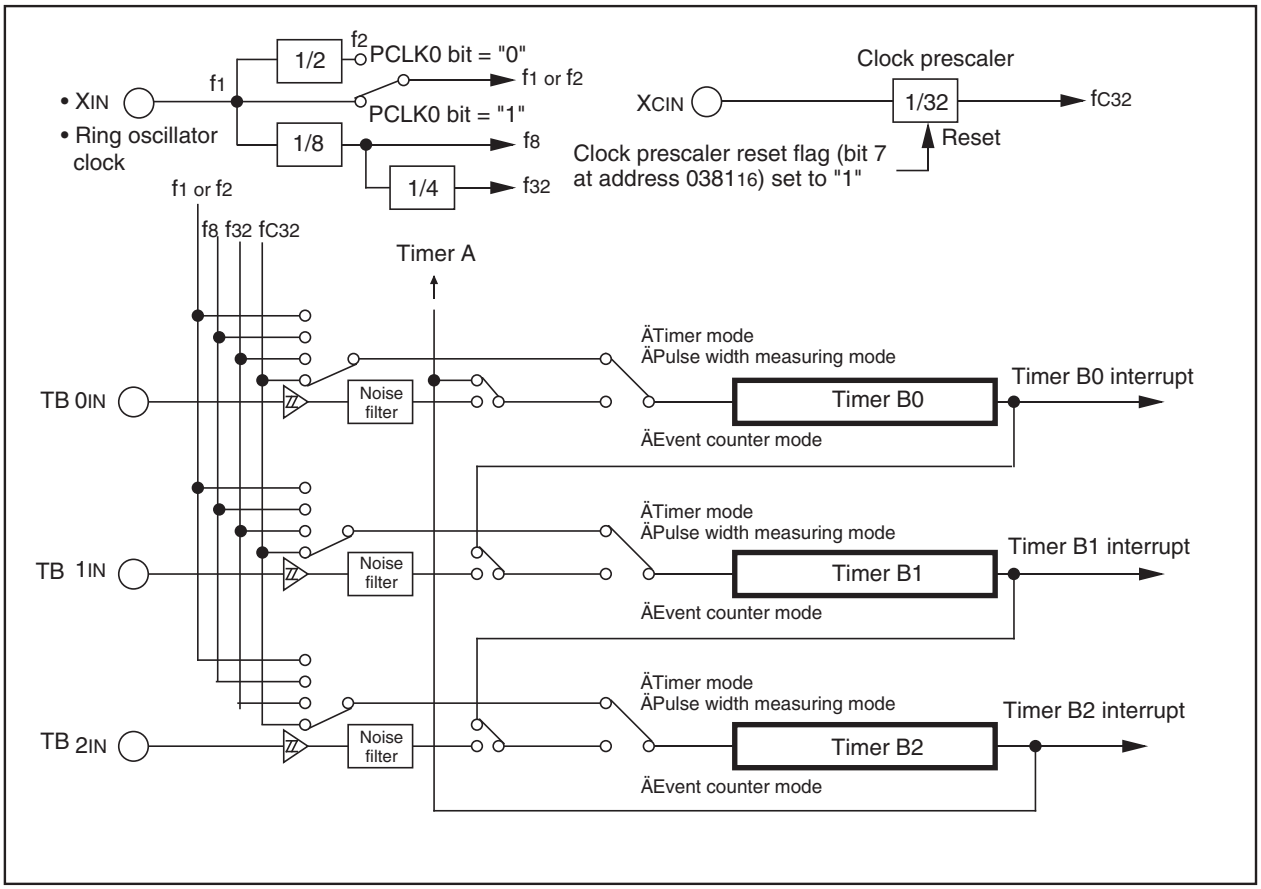


Figure 1.12.2. Timer B block diagram

Timer A

Figure 1.12.3 shows the block diagram of timer A. Figures 1.12.4 to 1.12.6 show the timer A-related registers.

Except in event counter mode, timers A0 through A4 all have the same function. Use the timer Ai mode register (i = 0 to 4) bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer over flow.
- One-shot timer mode: The timer stops counting when the count reaches "0000$_{16}$".
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

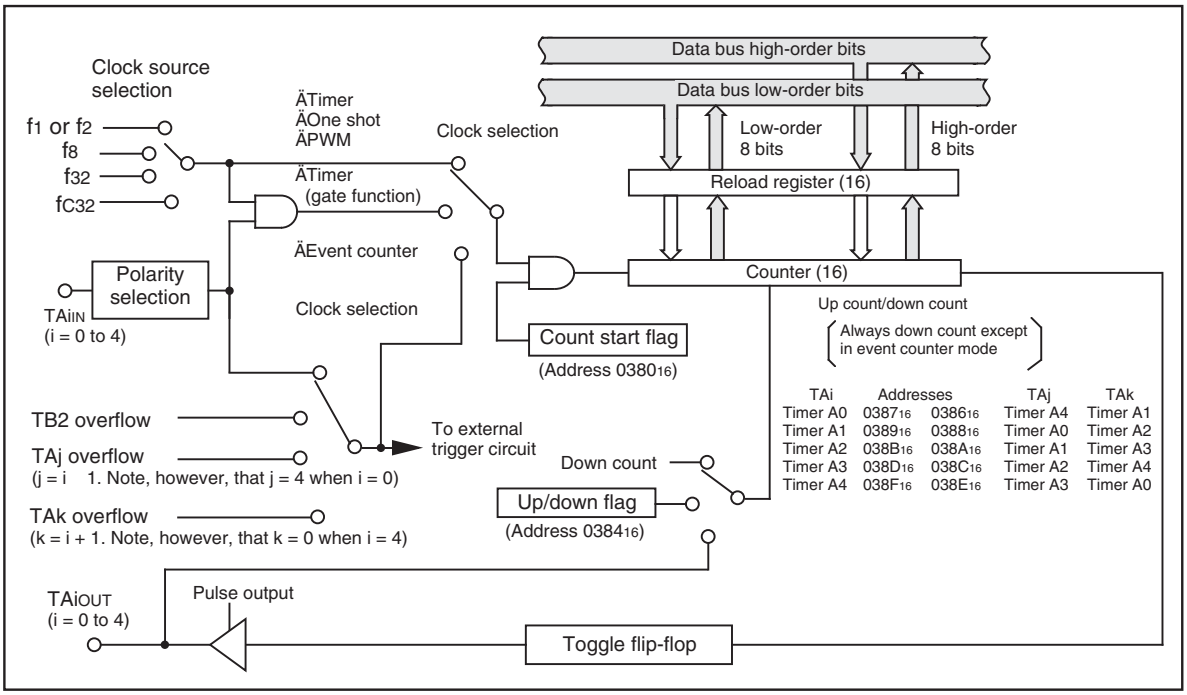


Figure 1.12.3. Block diagram of timer A

Timer Ai mode register

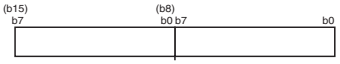
Symbol	Address	When reset
TAiMR(i=0 to 4)	039616 to 039A16	0016

b7	b6	b5	b4	b3	b2	b1	b0

Bit symbol	Bit name	Function	R	W
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	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 1.12.4. Timer A-related registers (1)

Timer Ai register (Note 1)

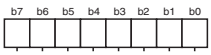


Symbol	Address	When reset
TA0	038716,038616	Indeterminate
TA1	038916,038816	Indeterminate
TA2	038B16,038A16	Indeterminate
TA3	038D16,038C16	Indeterminate
TA4	038F16,038E16	Indeterminate

Function	Values that can be set	R	W
· Timer mode Counts an internal count source	000016 to FFFF16	O	O
· Event counter mode Counts pulses from an external source or timer overflow	000016 to FFFF16	O	O
· One-shot timer mode Counts a one shot width	000016 to FFFF16 (Note 2,6)	X	O
· Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	000016 to FFFE16 (Note 3,4,6)	X	O
· Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	0016 to FE16 (High-order address) 0016 to FF16 (Low-order address) (Note 3,5,6)	X	O

Note 1: Read and write data in 16-bit units.
Note 2: When the timer Ai register is set to 000016 , the counter does not operate and the timer Ai interrupt request is not generated. When the pulse is set to output, the pulse does not output from the TAiOUT pin.
Note 3: When the timer Ai register is set to 000016 , the pulse width modulator does not operate and the output level of the TAiOUT pin remains L level, therefore the timer Ai interrupt request is not generated. This also occurs in the 8-bit pulse width modulator mode when the significant 8 high-order bits in the timer Ai register are set to 0016 .
Note 4: When the set value =n, PWM period and "H" width of PWM pule are as follows:
PWM perild: $(2^{16} - 1)$/ fi, "H" width of PWM pulse: n / fi
Note 5: When the set value of upper-address=n and lower-address=m, PWM period and "H" width of PWM pule are as follows:
PWM perild: $(2^{8} - 1)x(m+1)$/ fi, "H" width of PWM pulse: (m+1)n / fi
Note 6: Use MOV instruction to write to this register.

Count start flag



Symbol	Address	When reset
TABSR	038016	0016

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	O	O
TA1S	Timer A1 count start flag		O	O
TA2S	Timer A2 count start flag		O	O
TA3S	Timer A3 count start flag		O	O
TA4S	Timer A4 count start flag		O	O
TB0S	Timer B0 count start flag		O	O
TB1S	Timer B1 count start flag		O	O
TB2S	Timer B2 count start flag		O	O

Up/down flag (Note 1)

Symbol	Address	When reset
UDF	038416	0016

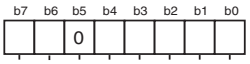
Bit symbol	Bit name	F unction	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	O	O
TA1UD	Timer A1 up/down flag		O	O
TA2UD	Timer A2 up/down flag		O	O
TA3UD	Timer A3 up/down flag		O	O
TA4UD	Timer A4 up/down flag		O	O
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled (Note 2) When not using the two-phase pulse signal processing function, set the select bit to 0	X	O
TA3P	Timer A3 two-phase pulse signal processing select bit		X	O
TA4P	Timer A4 two-phase pulse signal processing select bit		X	O

Note 1: Use MOV instruction to write to this register.
Note 2: Set the TAiIN and TAiOUT pins correspondent port direction registers to 0 .

Figure 1.12.5. Timer A-related registers (2)

One-shot start flag

Symbol	Address	When reset
ONSF	0382₁₆	00₁₆

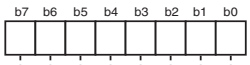


Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is 0	O	O
TA1OS	Timer A1 one-shot start flag		O	O
TA2OS	Timer A2 one-shot start flag		O	O
TA3OS	Timer A3 one-shot start flag		O	O
TA4OS	Timer A4 one-shot start flag		O	O
Reserved bit		Must always be set to 0	O	O
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on TA0IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	O	O
TA0TGH			O	O

Note: Set the corresponding port direction register to 0 .

Trigger select register

Symbol	Address	When reset
TRGSR	0383₁₆	00₁₆

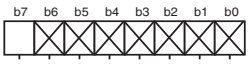


Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	O	O
TA1TGH			O	O
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2IN is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

Note: Set the corresponding port direction register to 0 .

Clock prescaler reset flag

Symbol	Address	When reset
CPSRF	0381₁₆	0XXXXXXX₂



Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is 0)	O	O

Figure 1.12.6. Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.12.1.) Figure 1.12.7 shows the timer Ai mode register in timer mode.

Table 1.12.1. Specifications of timer mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TAiIN pin's input signal • Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TAiMR(i=0 to 4) Address: 039616 to 039A16 When reset: 0016

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAiOUT pin is a pulse output pin)	○	○
MR1	Gate function select bit	b4 b3 0 X (Note 2): Gate function not available (TAiIN pin is a normal port pin) 1 0 : Timer counts only when TAiIN pin is held L (Note 3) 1 1 : Timer counts only when TAiIN pin is held H (Note 3)	○	○
MR2			○	○
MR3	0 (Must always be 0 in timer mode)		○	○
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}	○	○
TCK1			○	○

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: The bit can be 0 or 1 .
Note 3: Set the corresponding port direction register to 0 .

Figure 1.12.7. Timer Ai mode register in timer mode

Event Counter Mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timers A2, A3, and A4 can count a single-phase and a two-phase external signal. Table 1.12.2 lists timer specifications when counting a single-phase external signal. Figure 1.12.8 shows the timer Ai mode register in event counter mode.

Table 1.12.3 lists timer specifications when counting a two-phase external signal. Figure 1.12.9 shows the timer Ai mode register in event counter mode.

Table 1.12.2. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAiIN pin (effective edge can be selected by software) • TB2 overflow, TAj overflow
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register contents before continuing counting (Note)
Divide ratio	1/ (FFFF16 - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer register	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload (Transferred to counter at next reload time)
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 Each time the timer overflows or underflows, the TAiOUT pin's polarity is reversed

Note: This does not apply when the free-run function is selected.

Timer Ai mode register
(When not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

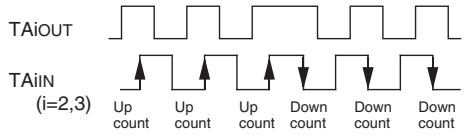
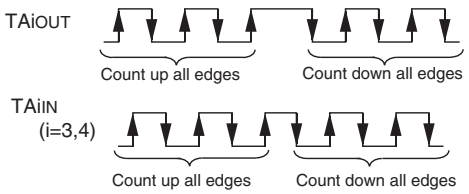
Symbol TAiMR(i = 0 to 4) Address 039616 to 039A16 When reset 0016

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (Note 1)	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 2) (TAiOUT pin is a pulse output pin)	○	○
MR1	Count polarity select bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 4)	○	○
MR3	0 (Must always be 0 in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Invalid when not using two-phase pulse signal processing Can be 0 or 1		○	○

Note 1: In event counter mode, the count source is selected by the event / trigger select bit (addresses 038216 and 038316).
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Valid only when counting an external signal.
Note 4: When an L signal is input to the TAiOUT pin, the downcount is activated. When H, the upcount is activated. Set the corresponding port direction register to 0.

Figure 1.12.8. Timer Ai mode register in event counter mode

Table 1.12.3. Timer 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 pin
Count operation	• Up count or down count can be selected by two-phase pulse signal • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note 1)
Divide ratio	1/ ($FFFF_{16}$ - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input (Set the TAiIN pin correspondent port direction register to "0".)
TAiOUT pin function	Two-phase pulse input (Set the TAiOUT pin correspondent port direction register to "0".)
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	• When counting stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function (Note 2)	• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on the TAiIN pin when input signal on the TAiOUT pin is "H".  • Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that the TAiIN pin goes "H" when the input signal on the TAiOUT pin is "H", the timer counts up rising and falling edges on the TAiOUT and TAiIN pins. If the phase relationship is such that the TAiIN pin goes "L" when the input signal on the TAiOUT pin is "H", the timer counts down rising and falling edges on the TAiOUT and TAiIN pins. 

Note 1: This does not apply when the free-run function is selected.

Note 2: Timer A3 alone can be selected. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

Timer Ai mode register
(When using two-phase pulse signal processing)

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

Symbol	Address	When reset
TAiMR(i = 2 to 4)	0398_{16} to $039A_{16}$	00_{16}

	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	0 (Must always be 0 when using two-phase pulse signal processing)		○	○
MR1	0 (Must always be 0 when using two-phase pulse signal processing)		○	○
MR2	1 (Must always be 1 when using two-phase pulse signal processing)		○	○
MR3	0 (Must always be 0 when using two-phase pulse signal processing)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse processing operation select bit (Note 1)(Note 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: This bit is valid for timer A3 mode register. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.
Note 2: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 0384_{16}) is set to "1". Also, always be sure to set the event/trigger select bit (addresses 0382_{16} and 0383_{16}) to '00_2'. And finally, set the port direction bits for TAiIN and TAiOUT to "0" (input mode).

Figure 1.12.9. Timer Ai mode register in event counter mode

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.12.4.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.12.12 shows the timer Ai mode register in one-shot timer mode.

Table 1.12.4. Timer specifications in one-shot timer mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down • When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TAi_{IN} pin function	Programmable I/O port or trigger input
TAi_{OUT} pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol: TAiMR(i = 0 to 4) Address: 0396_{16} to $039A_{16}$ When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	O	O
TMOD1			O	O
MR0	Pulse output function select bit	0 : Pulse is not output (TAi_{OUT} pin is a normal port pin) 1 : Pulse is output (Note 1) (TAi_{OUT} pin is a pulse output pin)	O	O
MR1	External trigger select bit (Note 2)	0 : Falling edge of TAi_{IN} pin's input signal (Note 3) 1 : Rising edge of TAi_{IN} pin's input signal (Note 3)	O	O
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select bits	O	O
MR3	0 (Must always be 0 in one-shot timer mode)		O	O
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}	O	O
TCK1			O	O

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: Valid only when the TAi_{IN} pin is selected by the event/trigger select bit (addresses 0382_{16} and 0383_{16}). If timer overflow is selected, this bit can be 1 or 0 .
Note 3: Set the corresponding port direction register to 0 .

Figure 1.12.12. Timer Ai mode register in one-shot timer mode

(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 1.12.5.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.12.13 shows the timer Ai mode register in pulse width modulation mode. Figure 1.12.14 shows the example of how a 16-bit pulse width modulator operates. Figure 1.12.15 shows the example of how an 8-bit pulse width modulator operates.

Table 1.12.5. Timer specifications in pulse width modulation mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM	High level width n / fi n : Set value • Cycle time(2^{16}-1) / fi fixed
8-bit PWM	• High level width × n (m+1) / fi n : values set to timer Ai register's high-order address • Cycle time (2^8-1) × (m+1) / fi m : values set to timer Ai register's low-order address
Count start condition	• External trigger is input • The timer overflows • The count start flag is set (= 1)
Count stop condition	• The count start flag is reset (= 0)
Interrupt request generation timing	PWM pulse goes "L"
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Symbol	Address	When reset
TAiMR(i=0 to 4)	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	O	O
TMOD1			O	O
MR0	1 (Must always be 1 in PWM mode)		O	O
MR1	External trigger select bit (Note 1)	0: Falling edge of TAiIN pin's input signal (Note 2) 1: Rising edge of TAiIN pin's input signal (Note 2)	O	O
MR2	Trigger select bit	0: Count start flag is valid 1: Selected by event/trigger select bits	O	O
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	O	O
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}	O	O
TCK1			O	O

Note 1: Valid only when the TAiIN pin is selected by the event/trigger select bit (addresses 0382_{16} and 0383_{16}). If timer overflow is selected, this bit can be 1 or 0.
Note 2: Set the corresponding port direction register to 0.

Figure 1.12.13. Timer Ai mode register in pulse width modulation mode

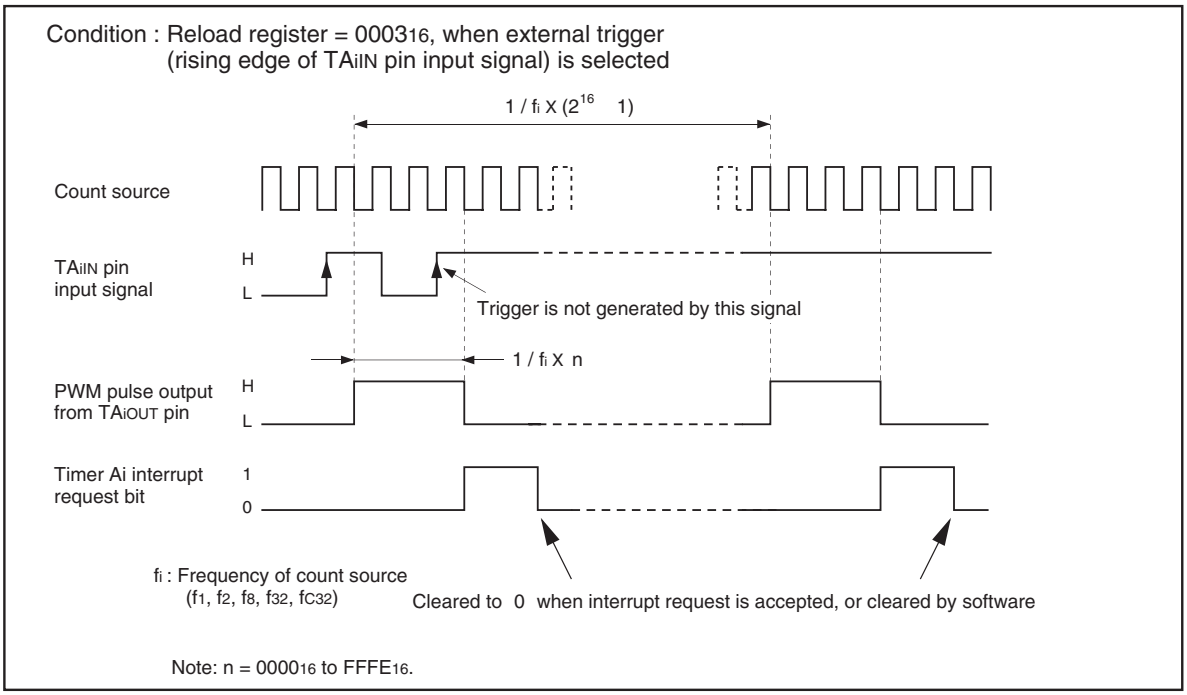


Figure 1.12.14. Example of how a 16-bit pulse width modulator operates

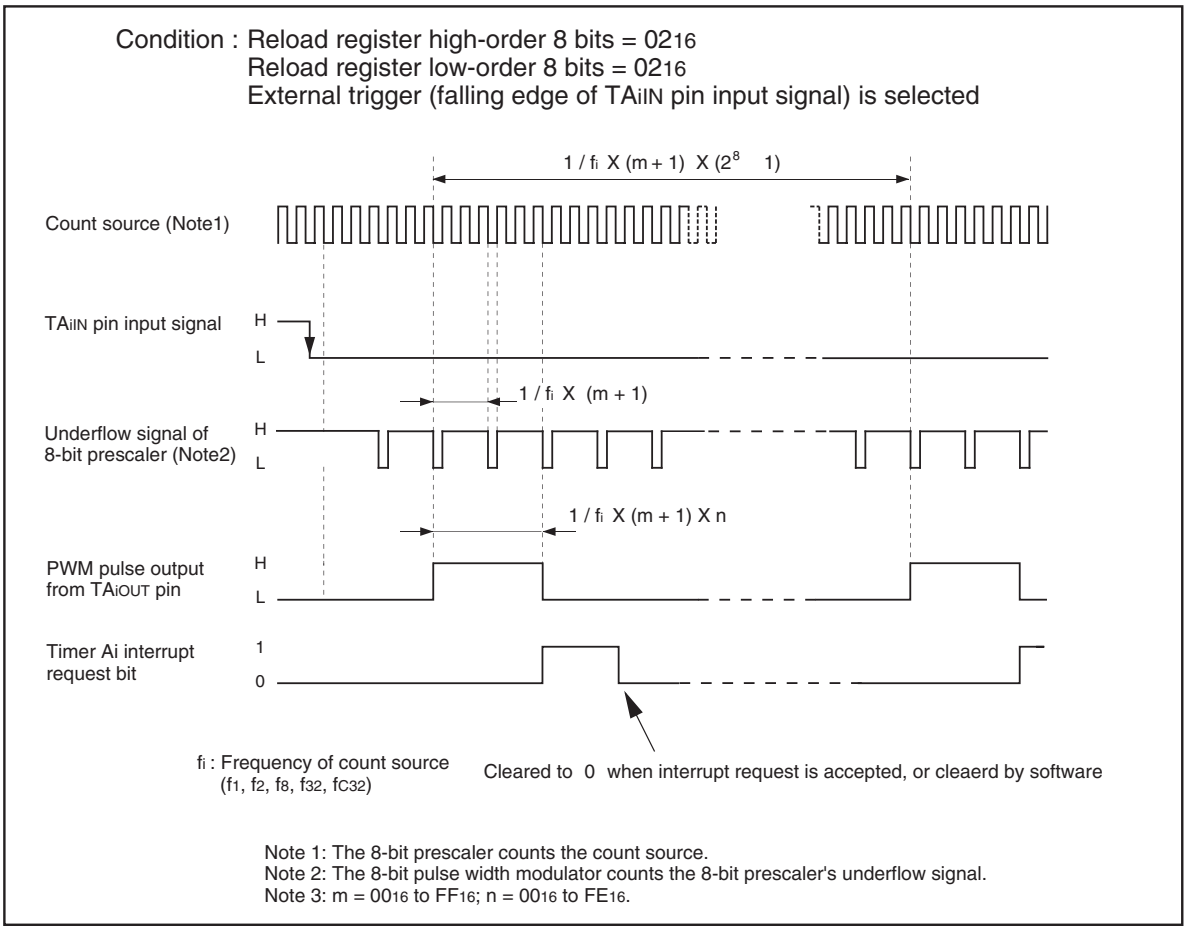


Figure 1.12.15. Example of how an 8-bit pulse width modulator operates

Timer B

Figure 1.13.16 shows the block diagram of timer B. Figures 1.13.17 and 1.13.18 show the timer B-related registers.

Use the timer Bi mode register (i = 0 to 2) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.

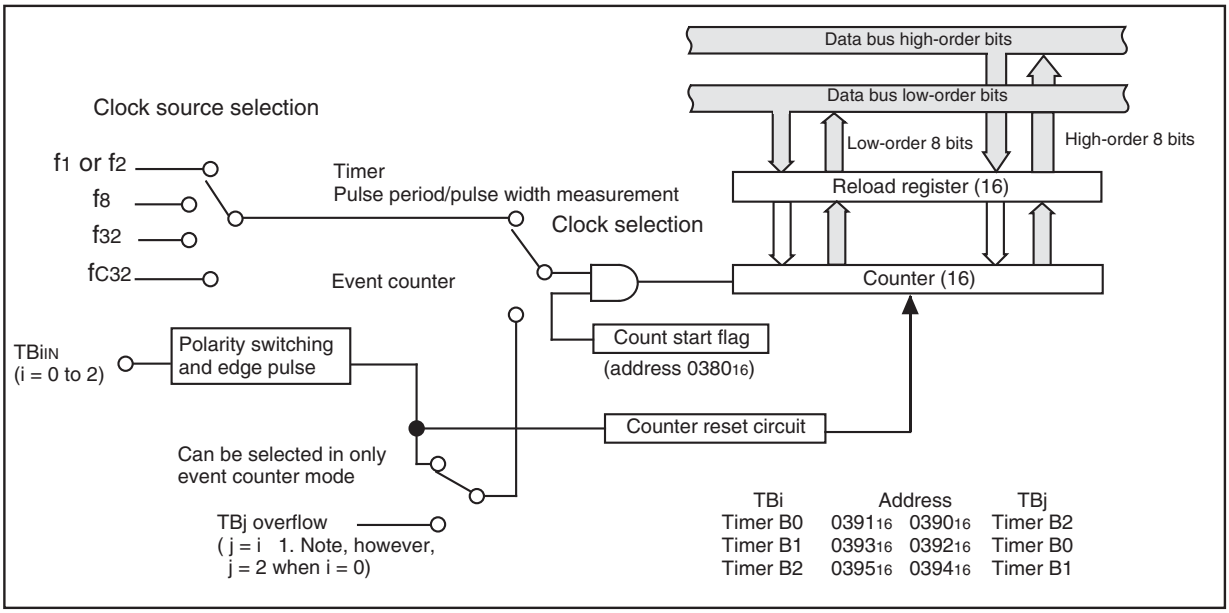


Figure 1.13.16. Block diagram of timer B

Timer Bi mode register

Symbol	Address	When reset
TBiMR(i = 0 to 2)	039B16 to 039D16	00XX00002

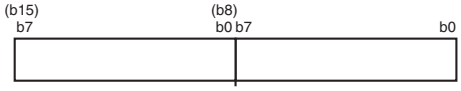
b7	b6	b5	b4	b3	b2	b1	b0

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Must not be set.	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1)	○
			× (Note 2)	×
MR3			○	×
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Note 1: Timer B0.
Note 2: Timer B1, timer B2.

Figure 1.13.17. Timer B-related registers (1)

Timer Bi register (Note)

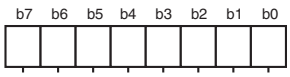


Symbol	Address	When reset
TB0	039116, 039016	Indeterminate
TB1	039316, 039216	Indeterminate
TB2	039516, 039416	Indeterminate

Function	Values that can be set	R	W
ÄTimer mode Counts the timer's period	000016 to FFFF16	○	○
ÄEvent counter mode Counts external pulses input or a timer overflow	000016 to FFFF16	○	○
ÄPulse period / pulse width measurement mode Measures a pulse period or width	——	○	×

Note: Read and write data in 16-bit units.

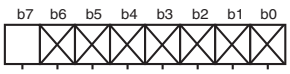
Count start flag



Symbol	Address	When reset
TABSR	038016	0016

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

Clock prescaler reset flag



Symbol	Address	When reset
CPSRF	038116	0XXXXXXX2

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is 0)	○	○

Figure 1.13.18. Timer B-related registers (2)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.13.6.) Figure 1.13.19 shows the timer Bi mode register in timer mode.

Table 1.13.6. Timer specifications in timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBi_{IN} pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol: TBiMR(i=0 to 2) Address: $039B_{16}$ to $039D_{16}$ When reset: $00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be 0 or 1		○	○
MR1			○	○
MR2	0 (Must always be 0 in timer mode ; i = 0)		○ (Note 1)	○
	Nothing is assiigned (i = 1, 2) . In an attempt to write to this bit, write 0 . The value, if read, turns out to be indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. In an attempt to write to this bit, write 0 . The value, if read in timer mode, turns out to be indeterminate.		○	×
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}	○	○
TCK1			○	○

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

Figure 1.13.19. Timer Bi mode register in timer mode

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 1.13.7.) Figure 1.13.20 shows the timer Bi mode register in event counter mode.

Table 1.13.7. Timer specifications in event counter mode

Item	Specification
Count source	• External signals input to TBiIN pin • Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol: TBiMR(i=0 to 2) Address: 039B16 to 039D16 When reset: 00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity select bit (Note 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 : Must not be set.	○	○
MR1			○	○
MR2	0 (Must always be 0 in event counter mode; i = 0)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2) . In an attempt to write to this bit, write 0 . The value, if read, turns out to be indeterminate.		× (Note 3)	×
MR3	Invalid in event counter mode. In an attempt to write to this bit, write 0 . The value, if read in event counter mode, turns out to be indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be 0 or 1 .		○	○
TCK1	Event clock select	0 : Input from TBiIN pin (Note 4) 1 : TBj overflow (j = i 1; however, j = 2 when i = 0)	○	○

Note 1: Valid only when input from the TBiIN pin is selected as the event clock. If timer's overflow is selected, this bit can be 0 or 1 .
Note 2: Timer B0.
Note 3: Timer B1, timer B2. .
Note 4: Set the corresponding port direction register to 0 .

Figure 1.13.20. Timer Bi mode register in event counter mode

(3) Pulse period/pulse width measurement mode

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 1.13.8.) Figure 1.13.21 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 1.13.22 shows the operation timing when measuring a pulse period. Figure 1.13.23 shows the operation timing when measuring a pulse width.

Table 1.13.8. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Up count • Counter value "0000_{16}" is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	• When measurement pulse's effective edge is input (Note 1) • When an overflow occurs. (Simultaneously, the timer Bi overflow flag changes to "1". Assume that the count start flag condition is "1" and then the timer Bi overflow flag becomes "1". If the timer Bi mode register has a write-access after next count cycle of the timer from the above condition, the timer Bi overflow flag becomes "0".)
TBi_{IN} pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.
Note 2: The value read out from the timer Bi register is indeterminate until the second effective edge is input after the timer has started counting.

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Symbol: TBiMR(i=0 to 2) Address: $039B_{16}$ to $039D_{16}$ When reset: $00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Must not be set.	○	○
MR1			○	○
MR2	0 (Must always be 0 in pulse period/pulse width measurement mode; i = 0)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2). In an attempt to write to this bit, write 0 . The value, if read, turns out to be indeterminate.		× (Note 3)	×
MR3	Timer Bi overflow flag (Note 1)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
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}	○	○
TCK1			○	○

Note 1: It is indeterminate when reset. Assume that the count start flag condition is 1 and then the timer Bi overflow flag becomes 1 . If the timer Bi mode register has a write access after next count cycle of the timer from the above condition, the timer Bi overflow flag becomes 0 . This flag cannot be set to 1 by software.
Note 2: Timer B0.
Note 3: Timer B1, timer B2.

Figure 1.13.21. Timer Bi mode register in pulse period/pulse width measurement mode

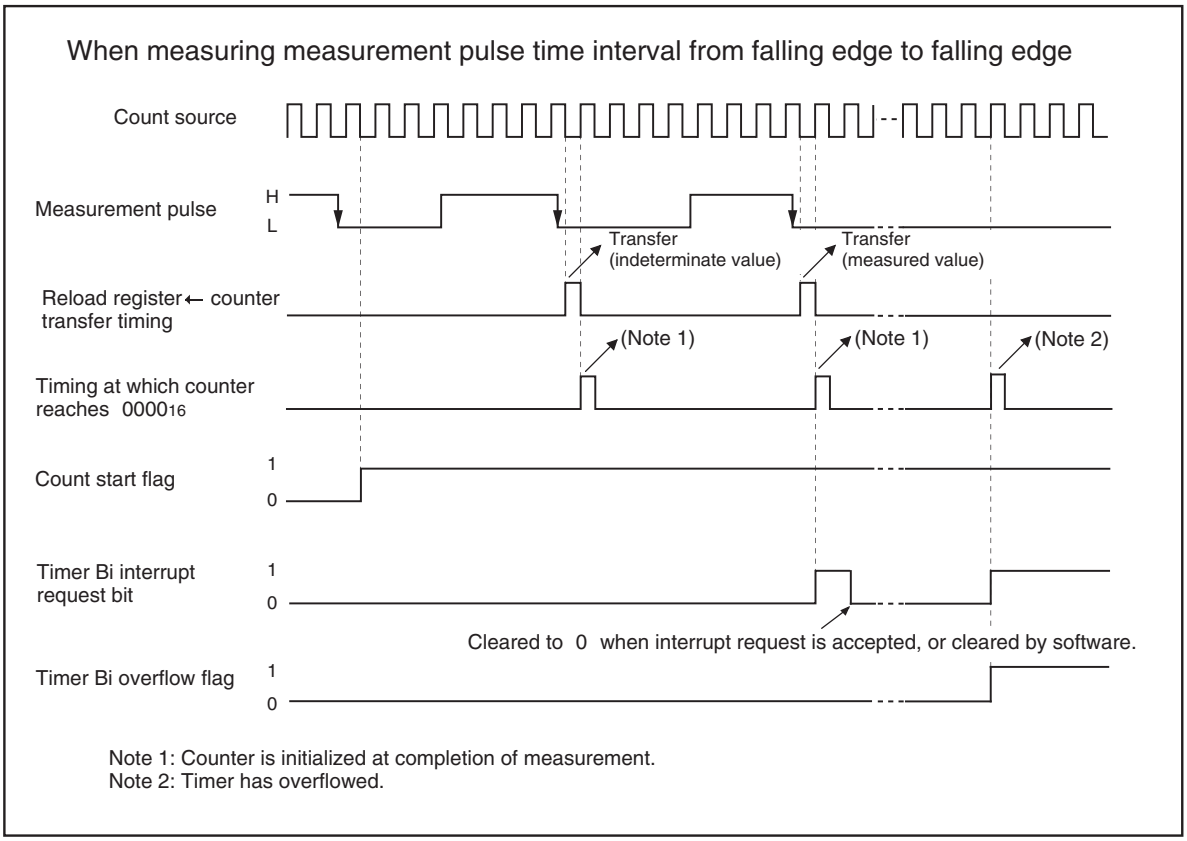


Figure 1.13.22. Operation timing when measuring a pulse period

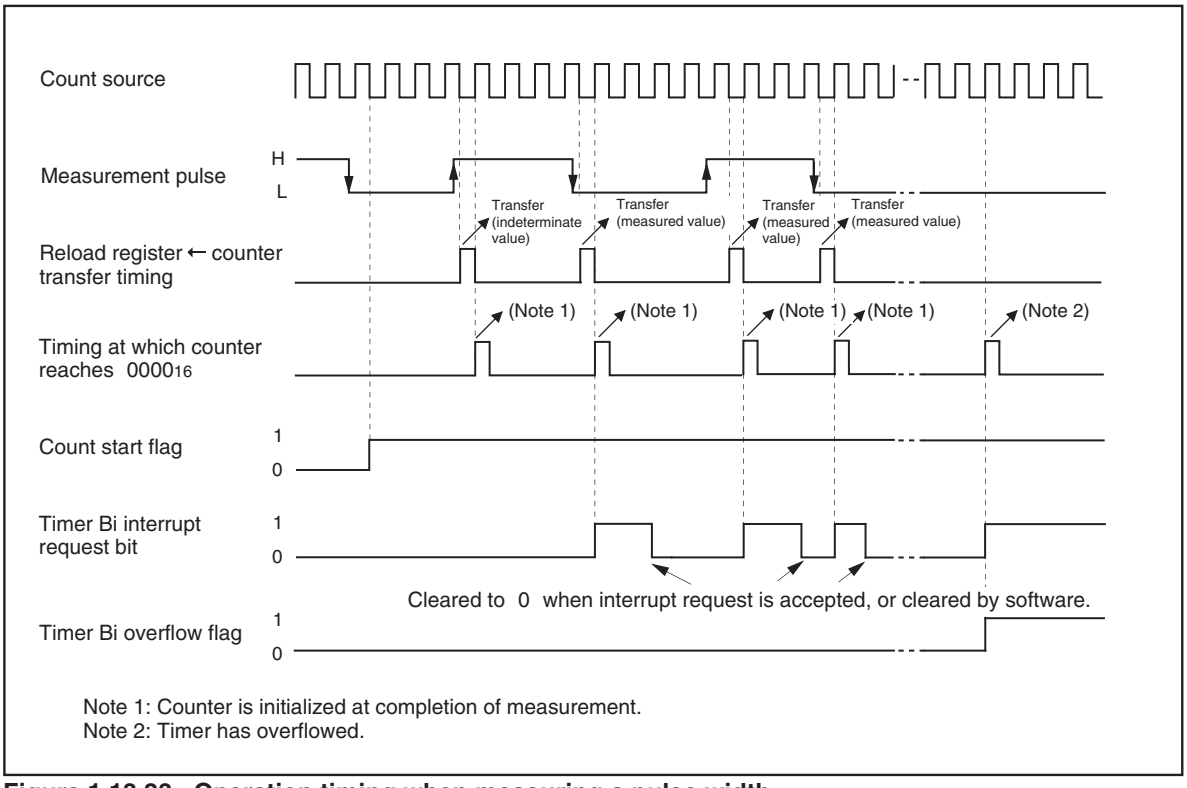
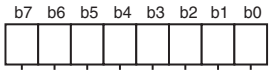


Figure 1.13.23. Operation timing when measuring a pulse width

Three-Phase Motor Control Timer Function

Use of more than one built-in timer A and timer B provides the means of outputting three-phase motor driving waveforms. This function requires that P85 become $\overline{SD}$, making it important that P85 not be used for general purpose I/O (GPIO). If the $\overline{SD}$ feature is not needed, then P85/$\overline{SD}$ must always be driven high.

Three-phase PWM control register 0 (Note 4)



Symbol: INVC0 | Address: 034816 | When reset: 0016

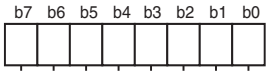
Bit symbol	Bit name	Description	R	W
INV00	Effective interrupt output polarity select bit	0: A timer B2 interrupt occurs when the timer A1 reload control signal is 1 . 1: A timer B2 interrupt occurs when the timer A1 reload control signal is 0 . (Note 3)	○	○
INV01	Effective interrupt output specification bit (Note 2)	0: Not specified. 1: Selected by the INV00 bit. (Note 3)	○	○
INV02	Mode select bit (Note 4)	0: Normal mode 1: Three-phase PWM output mode	○	○
INV03	Output control bit (Note 9)	0: Output disabled (Note 10) 1: Output enabled	○	○
INV04	Positive and negative phases concurrent L output disable function enable bit	0: Feature disabled 1: Feature enabled	○	○
INV05	Positive and negative phases concurrent L output detect flag	0: Not detected yet 1: Already detected (Note 5)	○	○
INV06	Modulation mode select bit	0: Triangular wave modulation mode (Note 6) 1: Sawtooth wave modulation mode (Note 7)	○	○
INV07	Software trigger bit	0: Ignored 1: Trigger generated (Note 8)	○	○

Note 1: Set bit 1 of the protect register (address 000A16) to 1 before writing to this register.
Note 2: Set bit 1 of this register to 1 after setting timer B2 interrupt occurrences frequency set counter.
Note 3: Effective only in three-phase mode 1(Three-phase PWM control register's bit 1 = 1).
Note 4: Selecting three-phase PWM output mode causes the dead time timer, the U, V, W phase output control circuits, and the timer B2 interrupt frequency set circuit works.
Note 5: No value other than 0 can be written.
Note 6: The dead time timer starts in synchronization with the falling edge of timer Ai output. The data transfer from the three-phase buffer register to the three-phase output shift register is made only once in synchronization with the transfer trigger signal after writing to the three-phase output buffer register.
Note 7: The dead time timer starts in synchronization with the falling edge of timer A output and with the transfer trigger signal. The data transfer from the three-phase output buffer register to the three-phase output shift register is made with respect to every transfer trigger.
Note 8: The value, when read, is 0 .
Note 9: The INV03 bit is set to 0 in the following cases:
- When reset.
- When positive and negative go active simultaneously while INV04 bit is 1 .
- When set to 0 in a program.
- When input on the $\overline{SD}$ pin changes state from H to L . (The INV03 bit cannot be set to 1 when $\overline{SD}$ input is L .)

Note 10: When INV03 = 1 (three-phase motor control timer output enabled) P80, P81, P72, P73, P74, and P75 function as U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$.

Figure 1.14.1. Registers related to timers for three-phase motor control (1)

Three-phase PWM control register 1 (Note 1)



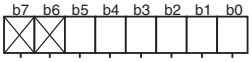
Symbol: INVC1 | Address: 034916 | When reset: 0016

Bit symbol	Bit name	D escription	R	W
INV10	Timer Ai start trigger signal select bit	0: Timer B2 overflow signal 1: Timer B2 overflow signal, signal for writing to timer B2	○	○
INV11	Timer A1-1, A2-1, A4-1 control bit	0: Three-phase mode 0 1: Three-phase mode 1	○	○
INV12	Dead time timer count source select bit	0 : f_1 or f_2 1 : f_1 divided by 2, or f_2 divided by 2	○	○
INV13	Carrier wave detect flag (Note 2)	0: Rising edge of triangular waveform 1: Falling edge of triangular waveform (Note 4)	○	—
INV14	Output polarity control bit	0 : Low active 1 : High active	○	○
INV15	Dead time invalid bit	0: Dead time valid bit 1: Dead time invalid bit	○	○
INV16	Dead time timer trigger select bit	0: Triggers from corresponding timer 1: Rising edge of corresponding phase output (Note 3)	○	○
$\overline{\text{(b7)}}$	Reserved bit	This bit should be set to "0".	○	○

Note 1: Set bit 1 of the protect register (address 000A16) to 1 before writing to this register.
Note 2: INV13 is valid only in triangular waveform mode (INV06=0) and three-phase mode (INV11=1).
Note 3:Usually set to 1 .
Note 4: When INV14="1", INV13="0" is falling edge of triangular waveform and INV13="1" is rising edge of triangular waveform.

Figure 1.14.2. Registers related to timers for three-phase motor control (2)

Three-phase output buffer register 0

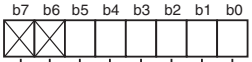


Symbol: IDB0 | Address: 034A16 | When reset: 0016

Bit Symbol	Bit name	Function	R	W
DU0	U phase output buffer 0	Setting in U phase output buffer 0	(Note)	○
DUB0	$\overline{U}$ phase output buffer 0	Setting in $\overline{U}$ phase output buffer 0	(Note)	○
DV0	V phase output buffer 0	Setting in V phase output buffer 0	(Note)	○
DVB0	$\overline{V}$ phase output buffer 0	Setting in $\overline{V}$ phase output buffer 0	(Note)	○
DW0	W phase output buffer 0	Setting in W phase output buffer 0	(Note)	○
DWB0	$\overline{W}$ phase output buffer 0	Setting in $\overline{W}$ phase output buffer 0	(Note)	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .			—	—

Note: When executing read instruction of this register, the contents of three-phase shift register is read out.

Three-phase output buffer register 1

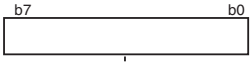


Symbol: IDB1 | Address: 034B16 | When reset: 0016

Bit Symbol	Bit name	Function	R	W
DU1	U phase output buffer 1	Setting in U phase output buffer 1	(Note)	○
DUB1	$\overline{U}$ phase output buffer 1	Setting in $\overline{U}$ phase output buffer 1	(Note)	○
DV1	V phase output buffer 1	Setting in V phase output buffer 1	(Note)	○
DVB1	$\overline{V}$ phase output buffer 1	Setting in $\overline{V}$ phase output buffer 1	(Note)	○
DW1	W phase output buffer 1	Setting in W phase output buffer 1	(Note)	○
DWB1	$\overline{W}$ phase output buffer 1	Setting in $\overline{W}$ phase output buffer 1	(Note)	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .			—	—

Note: When executing read instruction of this register, the contents of three-phase shift register is read out.

Dead time timer (Note)

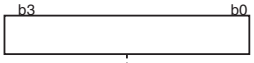


Symbol: DTT | Address: 034C16 | When reset: Indeterminate

Function		Values that can be set	R	W
Set dead time timer	1	to 255	—	○

Note: Use MOV instruction to write to this register.

Timer B2 interrupt occurrences frequency set counter (Note 1, 2, 3)



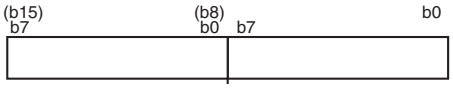
Symbol: ICTB2 | Address: 034D16 | When reset: Indeterminate

Function	Values that can be set	R	W
Set occurrence frequency of timer B2 interrupt request	1 to 15	—	○

Note 1: In setting 1 to bit 1 (INV01) - the effective interrupt output specification bit - of three-phase PWM control register 0, do not change the B2 interrupt occurrences frequency set counter to deal with the timer function for three-phase motor control.
Note 2: Do not write at the timing of an overflow occurrence in timer B2.
Note 3: Use MOV instruction to write to this register.

Figure 1.14.3. Registers related to timers for three-phase motor control (3)

Timer Ai (i=1, 2, 4) register (Notes 1, 4)

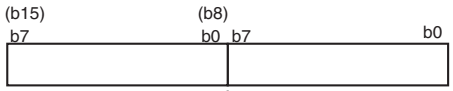


Symbol	Address	When reset
TA1	0389$_{16}$,0388$_{16}$	Indeterminate
TA2	038B$_{16}$,038A$_{16}$	Indeterminate
TA4	038F$_{16}$,038E$_{16}$	Indeterminate

Function	Values that can be set	R	W
· Timer mode Counts an internal count source	0000$_{16}$ to FFFF$_{16}$	○	○
· One-shot timer mode Counts a one shot width	0000$_{16}$ to FFFF$_{16}$ (Note 2, 3)	×	○

Note 1: Read and write data in 16-bit units.
Note 2: When the timer Ai register is set to "0000$_{16}$", the counter does not operate and a timer Ai interrupt does not occur.
Note 3: Use MOV instruction to write to this register.
Note 4: Do not write to these registers synchronously with a timer B2 underflow.

Timer Ai-1 (i=1, 2, 4) register (Notes 1, 2)

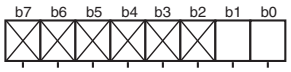


Symbol	Address	When reset
TA11	0343$_{16}$,0342$_{16}$	Indeterminate
TA21	0345$_{16}$,0344$_{16}$	Indeterminate
TA41	0347$_{16}$,0346$_{16}$	Indeterminate

Function	Values that can be set	R	W
Counts an internal count source	0000$_{16}$ to FFFF$_{16}$	○	○

Note 1: Read and write data in 16-bit units.
Note 2: Do not write to these registers synchronously with a timer B2 underflow.

Timer B2 special mode register



Symbol	Address	When reset
TB2SC	039E$_{16}$	XXXXXX00$_2$

Bit symbol	Bit name	Function	R	W
PWCOM	Timer B2 reload timing switching bit	0 : Next underflow 1 : Synchronized rising edge of triangular wave	○	○
IVPCR1	Three phase output port $\overline{SD}$ control bit 1 (Note 1,2,3)	0 : $\overline{SD}$ pin input port Hi-z disabled 1 : $\overline{SD}$ pin input port Hi-z enabled	○	○
Nothing is assigned. When write, set 0 . When read, its content is 0 .			—	—

Note 1: Set the protect register (address 000A16) to "1" before writing to this register.
Note 2: Related pins are P8$_0$, P8$_1$, P7$_2$, P7$_3$, P7$_4$, and P7$_5$, regardless of function assigned.
Note 3: Select P8$_5$ as input port prior to setting IVPCR1 to "1".

Figure 1.14.4. Registers related to timers for three-phase motor control (4)

Trigger select register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
TRGSR	0383₁₆	00₁₆

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

Note: Set the corresponding port direction register to 0 .

Count start flag

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
TABSR	0380₁₆	00₁₆

Bit symbol	Bit name	F unction	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

Figure 1.14.5. Registers related to timers for three-phase motor control (5)

Three-phase motor driving waveform output mode (three-phase PWM output mode)

Setting “1” in the mode select bit (bit 2 at 0348_{16}) shown in Figure 1.14.1 - causes three-phase PWM output mode that uses four timers A1, A2, A4, and B2 to be selected. As shown in Figure 1.14.6, set timers A1, A2, and A4 in one-shot timer mode, set the trigger in timer B2, and set timer B2 in timer mode using the respective timer mode registers.

Timer Ai mode register

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

Symbol	Address	When reset
TA1MR	039716	0016
TA2MR	039816	0016
TA3MR	$039A16$	0016

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 (Must always be 0 in three-phase PWM output mode)	○	○
MR1	External trigger select bit	Invalid in three-phase PWM output mode	○	○
MR2	Trigger select bit	1 : Selected by event/trigger select register	○	○
MR3	0 (Must always be 0 in one-shot timer mode)		○	○
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}	○	○
TCK1			○	○

Timer B2 mode register

b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

Symbol	Address	When reset
TB2MR	$039D16$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be 0 or 1		○	○
MR1			○	○
MR2	0 (Must always be 0 in timer mode)		○	○
MR3	Invalid in timer mode. In an attempt to write to this bit, write "0". When read in timer mode, its content is indeterminate.		○	×
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}	○	○
TCK1			○	○

Figure 1.14.6. Timer mode registers in three-phase PWM output mode

Figure 1.14.7 shows the block diagram for three-phase PWM output mode. When selecting output polarity “L” active in three-phase PWM output mode, the positive-phase waveforms (U phase, V phase, and W phase) and negative waveforms (U phase, V phase, and W phase), six waveforms in total, are output from $P8_0$, $P8_1$, $P7_2$, $P7_3$, $P7_4$, and $P7_5$ as active on the “L” level. Of the timers used in this mode, timer A4 controls the U phase and U phase, timer A1 controls the V phase and V phase, and timer A2 controls the W phase and W phase respectively; timer B2 controls the periods of one-shot pulse output from timers A4, A1, and A2.

In outputting a waveform, dead time can be set so as to cause the “L” level of the positive waveform output (U phase, V phase, and W phase) not to lap over the “L” level of the negative waveform output (U phase, V phase, and W phase).

To set short circuit time, use three 8-bit timers sharing the reload register for setting dead time. A value from 1 through 255 can be set as the count of the timer for setting dead time. The timer for setting dead time works as a one-shot timer. If a value is written to the dead time timer ($034C_{16}$), the value is written to the reload register shared by the three timers for setting dead time.

Any of the timers for setting dead time takes the value of the reload register into its counter, if a start trigger comes from its corresponding timer, and performs a down count in line with the clock source selected by the dead time timer count source select bit (bit 2 at 0349_{16}). The timer can receive another trigger again before the workings due to the previous trigger are completed. In this instance, the timer performs a down count from the reload register’s content after its transfer, provoked by the trigger, to the timer for setting dead time.

Since the timer for setting dead time works as a one-shot timer, it starts outputting pulses if a trigger comes; it stops outputting pulses as soon as its content becomes 00_{16}, and waits for the next trigger to come.

The positive waveforms (U phase, V phase, and W phase) and the negative waveforms (U phase, V phase, and W phase) in three-phase PWM output mode are output from respective ports by means of setting “1” in the output control bit (bit 3 at 0348_{16}). Setting “0” in this bit causes the ports to be the state of set by port direction register. This bit can be set to “0” not only by use of the applicable instruction, but by entering a falling edge in the NMI terminal or by resetting. Also, if “1” is set in the positive and negative phases concurrent L output disable function enable bit (bit 4 at 0348_{16}) causes one of the pairs of U phase and U phase, V phase and V phase, and W phase and W phase concurrently go to “L”, as a result, the port becomes the state of set by port direction register.

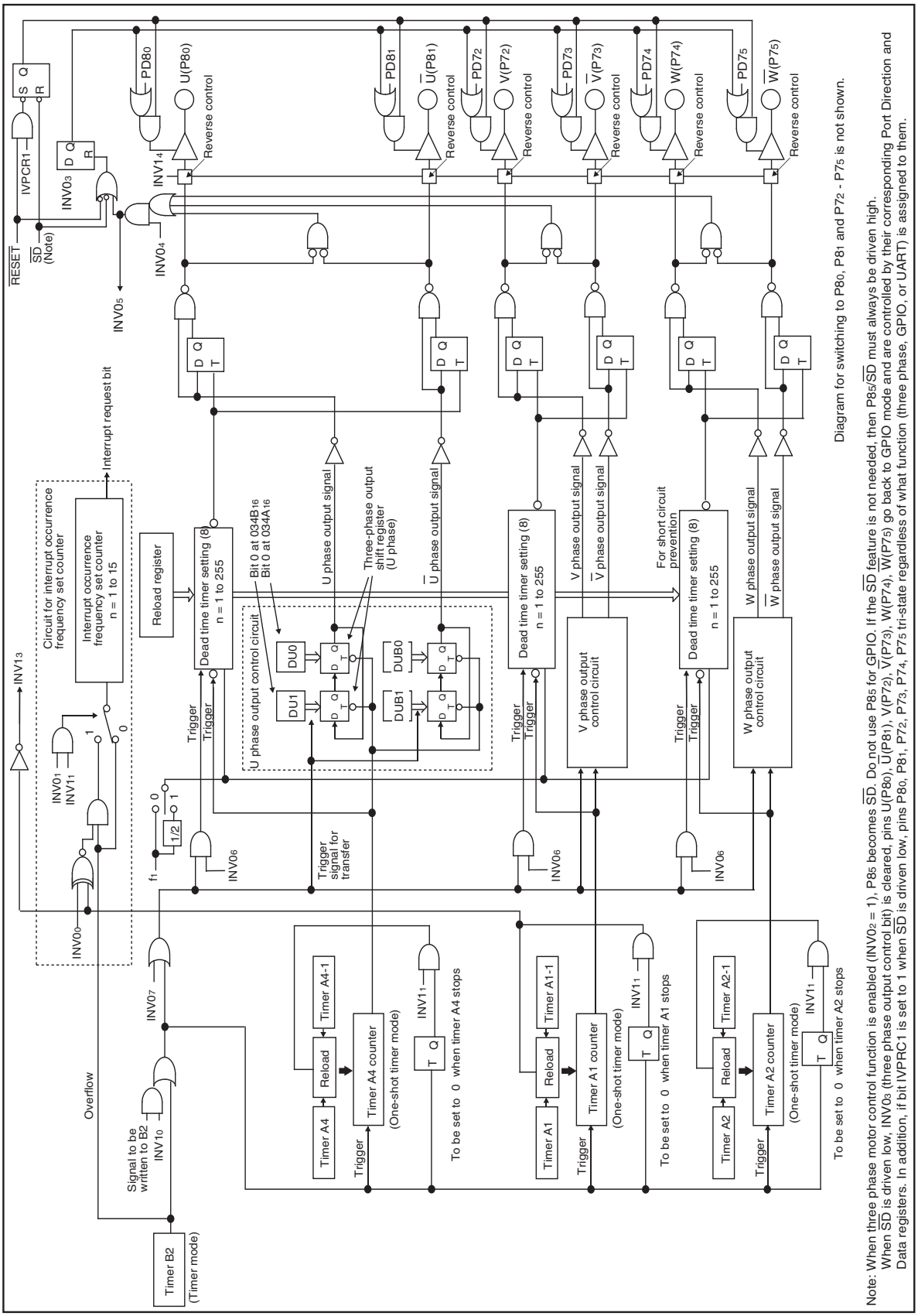


Figure 1.14.7. Block diagram for three-phase PWM output mode

Triangular wave modulation

To generate a PWM waveform of triangular wave modulation, set “0” in the modulation mode select bit (bit 6 at 034816). Also, set “1” in the timers A4-1, A1-1, A2-1 control bit (bit 1 at 034916). In this mode, each of timers A4, A1, and A2 has two timer registers, and alternately reloads the timer register’s content to the counter every time timer B2 counter’s content becomes 000016. If “0” is set to the effective interrupt output specification bit (bit 1 at 034816), the frequency of interrupt requests that occur every time the timer B2 counter’s value becomes 000016 can be set by use of the timer B2 counter (034D16) for setting the frequency of interrupt occurrences. The frequency of occurrences is given by (setting; setting • 0).

Setting “1” in the effective interrupt output specification bit (bit 1 at 034816) provides the means to choose which value of the timer A1 reload control signal to use, “0” or “1”, to cause timer B2’s interrupt request to occur. To make this selection, use the effective interrupt output polarity selection bit (bit 0 at 034816).

An example of U phase waveform is shown in Figure 1.14.8, and the description of waveform output workings is given below. Set “1” in DU0 (bit 0 at 034A16). And set “0” in DUB0 (bit 1 at 034A16). In addition, set “0” in DU1 (bit 0 at 034B16) and set “1” in DUB1 (bit 1 at 034B16). Also, set “0” in the effective interrupt output specification bit (bit 1 at 034816) to set a value in the timer B2 interrupt occurrence frequency set counter. By this setting, a timer B2 interrupt occurs when the timer B2 counter’s content becomes 000016 as many as (setting) times. Furthermore, set “1” in the effective interrupt output specification bit (bit 1 at 034816), set “0” in the effective interrupt output polarity select bit (bit 0 at 034816) and set "1" in the interrupt occurrence frequency set counter (034D16). These settings cause a timer B2 interrupt to occur every other interval when the U phase output goes to “H”.

When the timer B2 counter’s content becomes 000016, timer A4 starts outputting one-shot pulses. In this instance, the content of DU1 (bit 0 at 034B16) and that of DU0 (bit 0 at 034A16) are set in the three-phase output shift register (U phase), the content of DUB1 (bit 1 at 034B16) and that of DUB0 (bit 1 at 034A16) are set in the three-phase output shift register (U phase). After triangular wave modulation mode is selected, however, no setting is made in the shift register even though the timer B2 counter’s content becomes 000016.

The value of DU0 and that of DUB0 are output to the U terminal (P80) and to the U terminal (P81) respectively. When the timer A4 counter counts the value written to timer A4 (038F16, 038E16) and when timer A4 finishes outputting one-shot pulses, the three-phase shift register’s content is shifted one position, and the value of DU1 and that of DUB1 are output to the U phase output signal and to U phase output signal respectively. At this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the “L” level of the U phase waveform does not lap over the “L” level of the U phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the “H” level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register’s content changes from “1” to “0” by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, “0” already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the “L” level. When the timer B2 counter’s content becomes 000016, the timer A4 counter starts counting the value written to timer A4-1 (034716, 034616), and starts outputting one-shot pulses. When timer A4 finishes outputting one-shot pulses, the three-phase shift register’s content is shifted one position, but if the three-phase output shift register’s content changes from “0” to “1” as a result of the shift, the output level changes from “L” to “H” without waiting for the timer for setting dead time to finish outputting one-shot pulses. A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the U phase side is used, the workings in generating a U phase

waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the “L” level of the U phase waveform doesn’t lap over that of the U phase waveform, which has the opposite phase of the U phase waveform. The width of the “L” level too can be adjusted by varying the values of timer B2, timer A4, and timer A4-1. In dealing with the V and W phases, and V and W phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and U phases to generate an intended waveform.

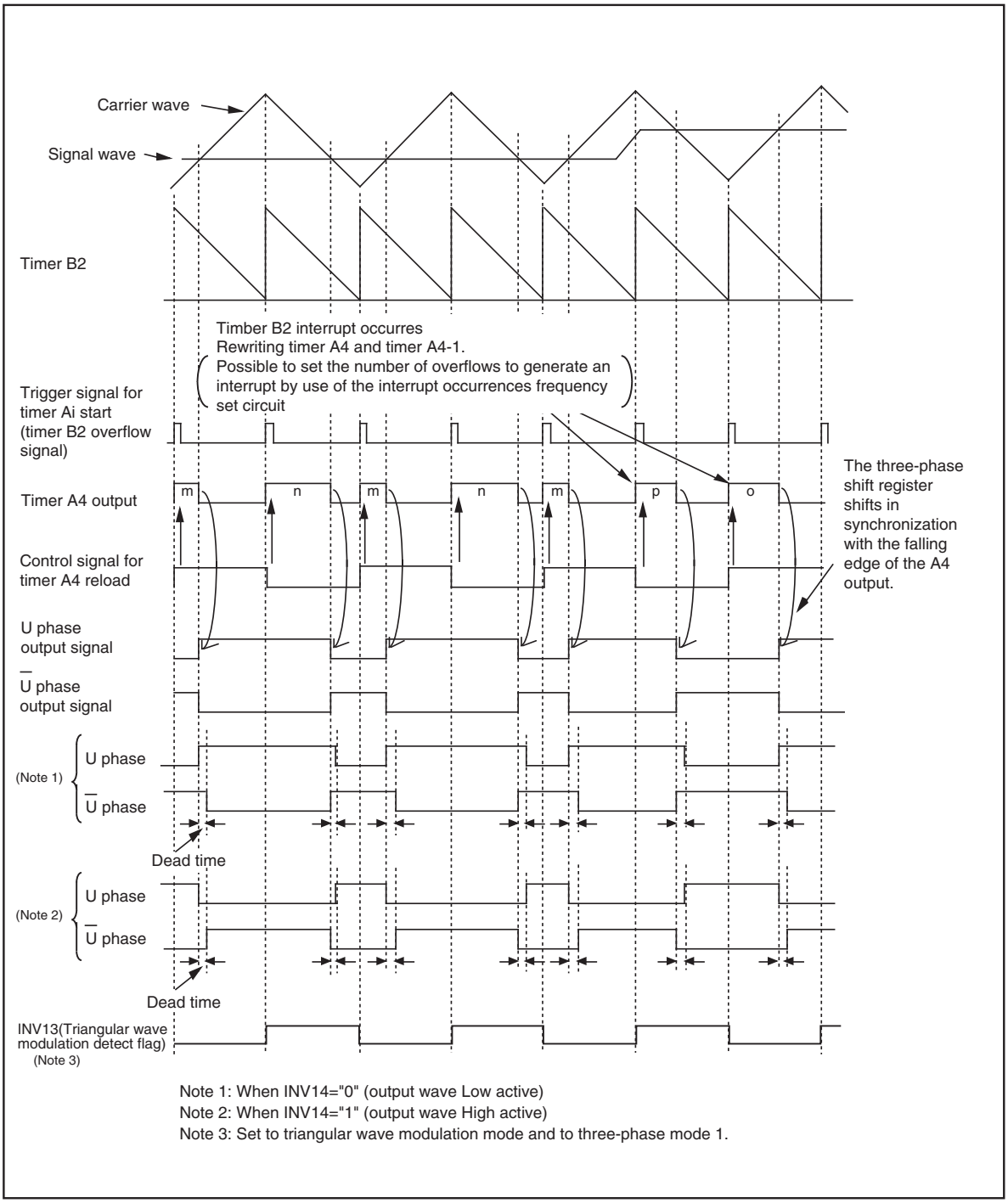


Figure 1.14.8. Timing chart of operation (1)

Assigning certain values to DU0 (bit 0 at 034A16) and DUB0 (bit 1 at 034A16), and to DU1 (bit 0 at 034B16) and DUB1 (bit 1 at 034B16) allows the user to output the waveforms as shown in Figure 1.14.9, that is, to output the U phase alone, to fix U phase to “H”, to fix the U phase to “H,” or to output the U phase alone.

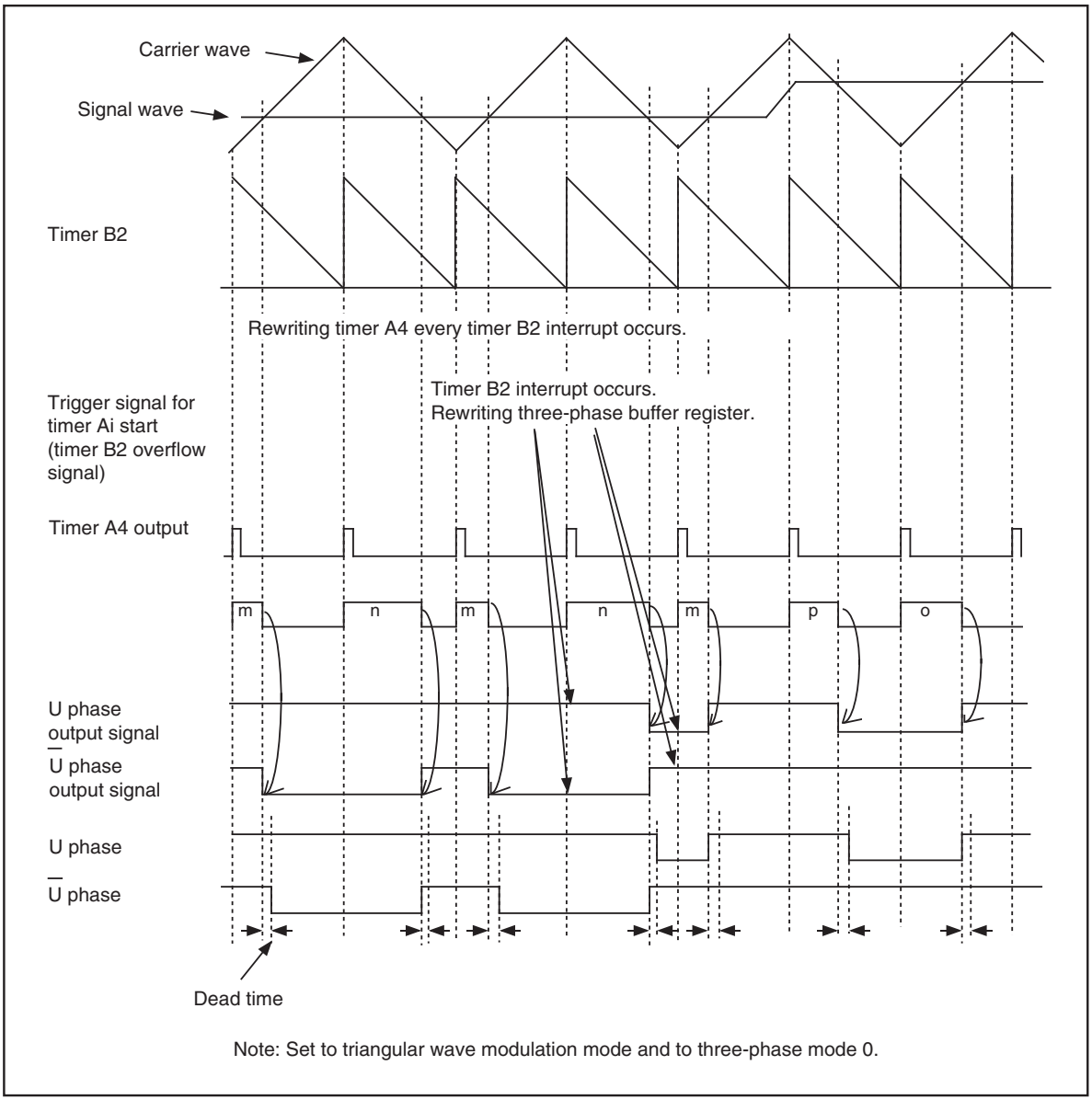


Figure 1.14.9. Timing chart of operation (2)

Figures 1.14.1 to 1.14.5 show registers related to timers for three-phase motor control.

Sawtooth modulation

To generate a PWM waveform of sawtooth wave modulation, set “1” in the modulation mode select bit (bit 6 at 0348_{16}). Also, set “0” in the timers A4-1, A1-1, and A2-1 control bit (bit 1 at 0349_{16}). In this mode, the timer registers of timers A4, A1, and A2 comprise conventional timers A4, A1, and A2 alone, and reload the corresponding timer register’s content to the counter every time the timer B2 counter’s content becomes 0000_{16}. The effective interrupt output specification bit (bit 1 at 0348_{16}) and the effective interrupt output polarity select bit (bit 0 at 0348_{16}) go nullified.

An example of U phase waveform is shown in Figure 1.14.10, and the description of waveform output workings is given below. Set “1” in DU0 (bit 0 at $034A_{16}$), and set “0” in DUB0 (bit 1 at $034A_{16}$). In addition, set “0” in DU1 (bit 0 at $034A_{16}$) and set “1” in DUB1 (bit 1 at $034A_{16}$).

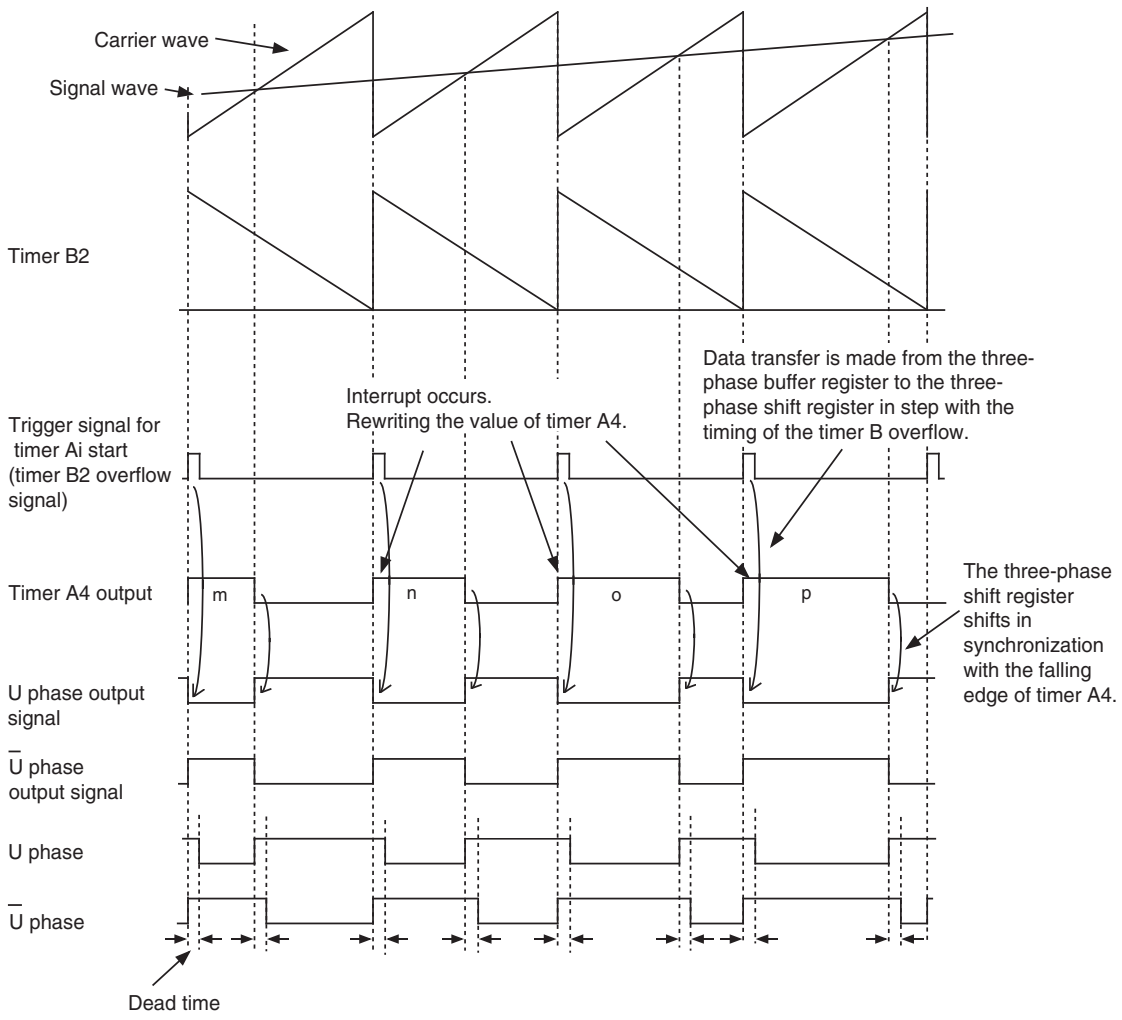
When the timber B2 counter’s content becomes 0000_{16}, timer B2 generates an interrupt, and timer A4 starts outputting one-shot pulses at the same time. In this instance, the contents of the three-phase buffer registers DU1 and DU0 are set in the three-phase output shift register (U phase), and the contents of DUB1 and DUB0 are set in the three-phase output shift register (U phase). After this, the three-phase buffer register’s content is set in the three-phase shift register every time the timer B2 counter’s content becomes 0000_{16}.

The value of DU0 and that of DUB0 are output to the U terminal ($P8_0$) and to the U terminal ($P8_1$) respectively. When the timer A4 counter counts the value written to timer A4 ($038F_{16}$, $038E_{16}$) and when timer A4 finishes outputting one-shot pulses, the three-phase output shift register’s content is shifted one position, and the value of DU1 and that of DUB1 are output to the U phase output signal and to the U output signal respectively. At this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the “L” level of the U phase waveform doesn’t lap over the “L” level of the U phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the “H” level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register’s content changes from “1” to “0 ”by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, 0 already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the “L” level. When the timer B2 counter’s content becomes 0000_{16}, the contents of the three-phase buffer registers DU1 and DU0 are set in the three-phase output shift register (U phase), and the contents of DUB1 and DUB0 are set in the three-phase output shift register (U phase) again.

A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the U phase side is used, the workings in generating a U phase waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the “L” level of the U phase waveform doesn’t lap over that of the U phase waveform, which has the opposite phase of the U phase waveform. The width of the “L” level too can be adjusted by varying the values of timer B2 and timer A4. In dealing with the V and W phases, and V and W phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and U phases to generate an intended waveform.

Setting “1” both in DUB0 and in DUB1 provides a means to output the U phase alone and to fix the U phase output to “H” as shown in Figure 1.14.11.

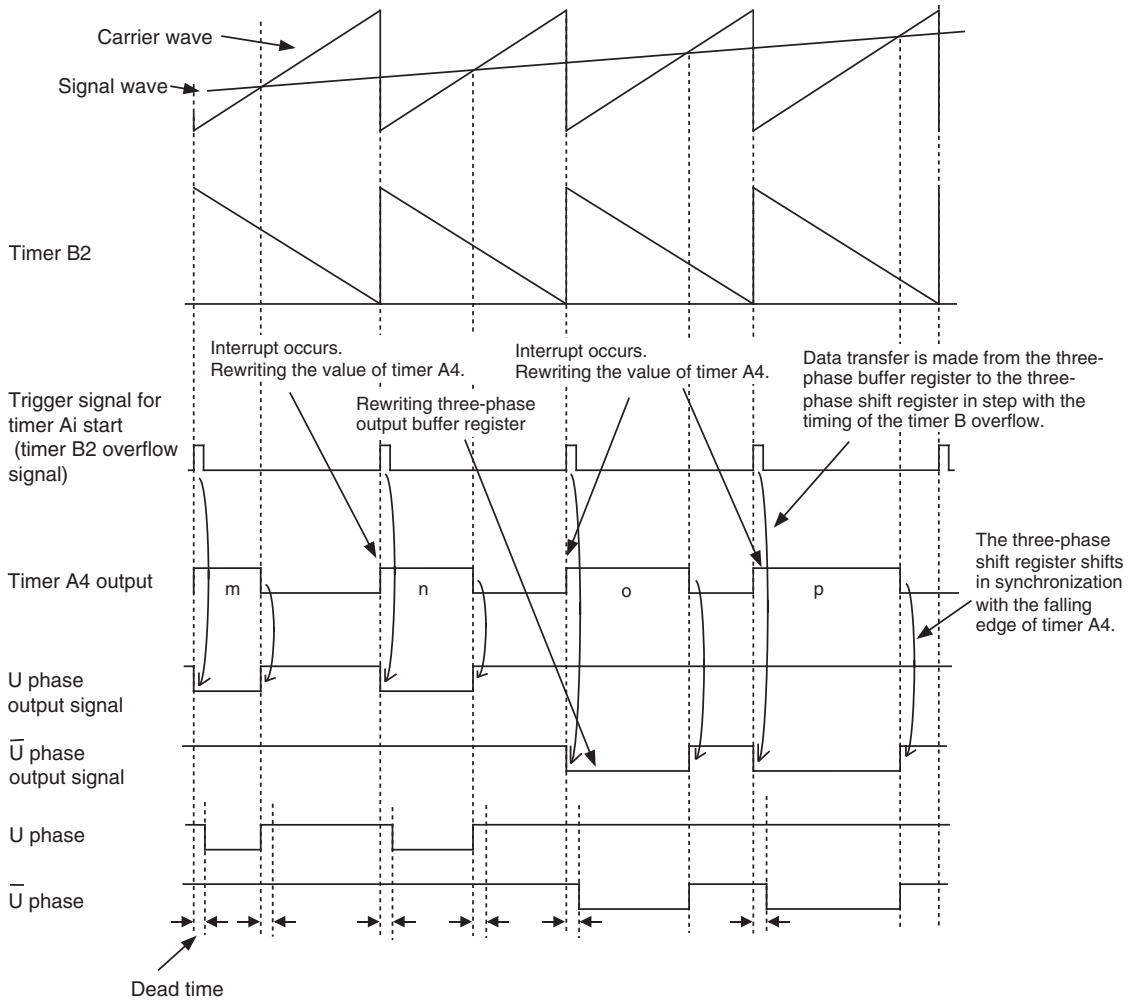
A carrier wave of sawtooth waveform



Note: Set to sawtooth modulation mode and to three-phase mode 0.

Figure 1.14.10. Timing chart of operation (3)

A carrier wave of sawtooth waveform



Note: Set to sawtooth modulation mode and to three-phase mode 0.

Figure 1.14.11. Timing chart of operation (4)

Serial I/O

Serial I/O is configured as three channels: UART0, UART1, and UART2.

UARTi (i = 0 to 2)

UART0, UART1 and UART2 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 1.15.1 shows the block diagram of UART0, UART1 and UART2. Figures 1.15.2 and 1.15.3 show the block diagram of the transmit/receive unit.

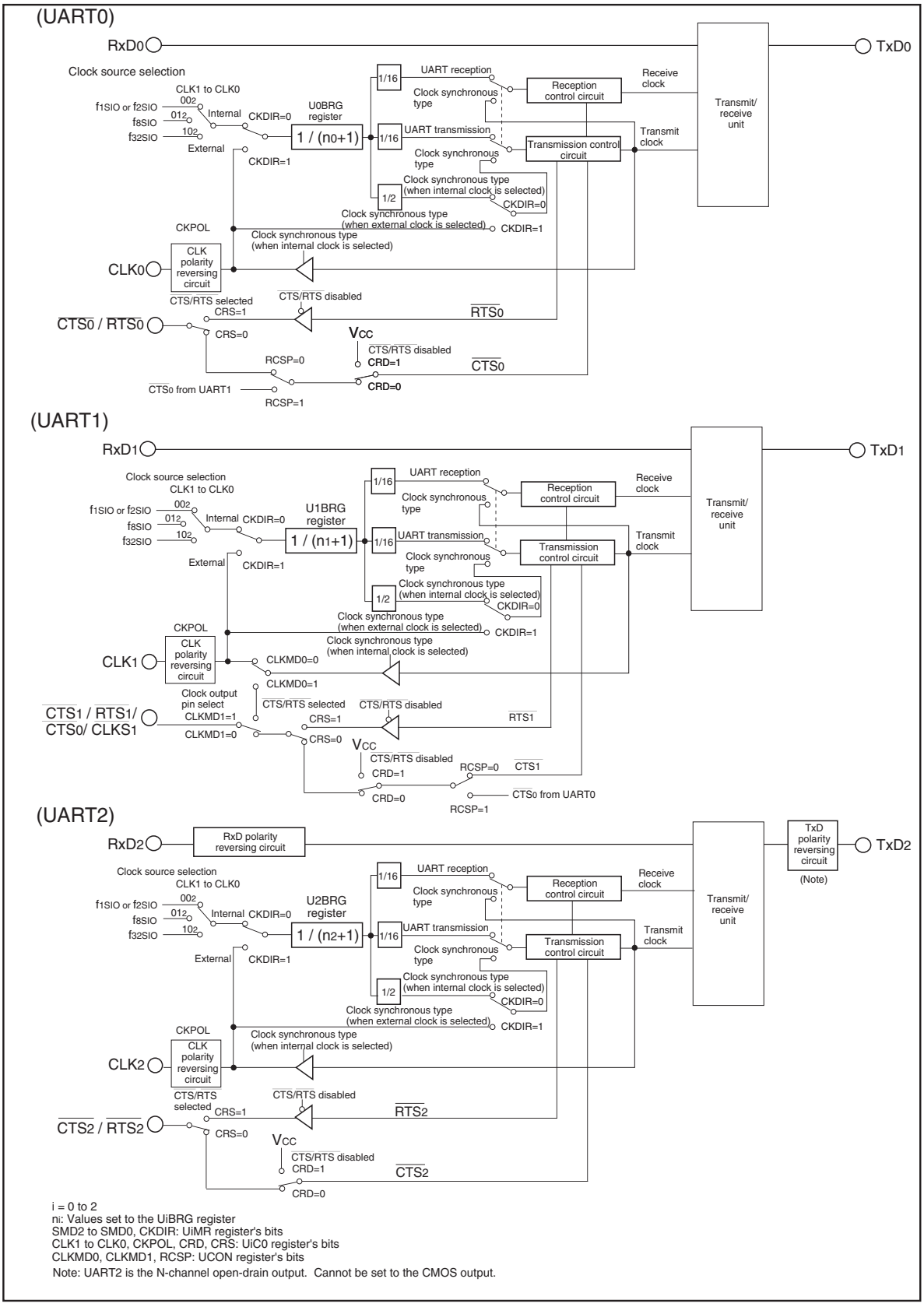
UARTi (i = 0 to 2) has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses $03A0_{16}$, $03A8_{16}$ and 0378_{16}) determine whether UARTi is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART0, UART1 and UART2 have almost the same functions. UART2, in particular, is used for the SIM (Subscriber Identity Module) interface with some extra settings added in clock-asynchronous serial I/O mode.

Table 1.15.1 shows the comparison of functions of UART0 through UART2, and Figures 1.15.4 to 1.15.10 show the registers related to UARTi.

Table 1.15.1. Comparison of functions of UART0 through UART2

Function	UART0	UART1	UART2
CLK polarity selection	Supported (Note 1)	Supported (Note 1)	Supported (Note 1)
LSB first / MSB first selection	Supported (Note 1)	Supported (Note 1)	Supported (Note 2)
Continuous receive mode selection	Supported (Note 1)	Supported (Note 1)	Supported (Note 1)
Transfer clock output from multiple pins selection	Not Supported	Supported (Note 1)	Not Supported
Serial data logic switch	Not Supported	Not Supported	Supported (Note 3)
TxD, RxD I/O polarity switch	Not Supported	Not Supported	Supported
TxD, RxD port output format	CMOS output/Nch OD	CMOS output/Nch OD	N-channel open-drain output
Parity error signal output	Not Supported	Not Supported	Supported (Note 3)
Bus collision detection	Not Supported	Not Supported	Supported

Note 1: Only when in clock synchronous serial I/O mode.
Note 2: Only when in clock synchronous serial I/O mode and 8-bit UART mode.
Note 3: Used for SIM interface.



i = 0 to 2
ni: Values set to the UiBRG register
SMD2 to SMD0, CKDIR: UiMR register's bits
CLK1 to CLK0, CKPOL, CRD, CRS: UiC0 register's bits
CLKMD0, CLKMD1, RCSP: UCON register's bits
Note: UART2 is the N-channel open-drain output. Cannot be set to the CMOS output.

Figure 1.15.1. Block diagram of UARTi (i = 0 to 2)

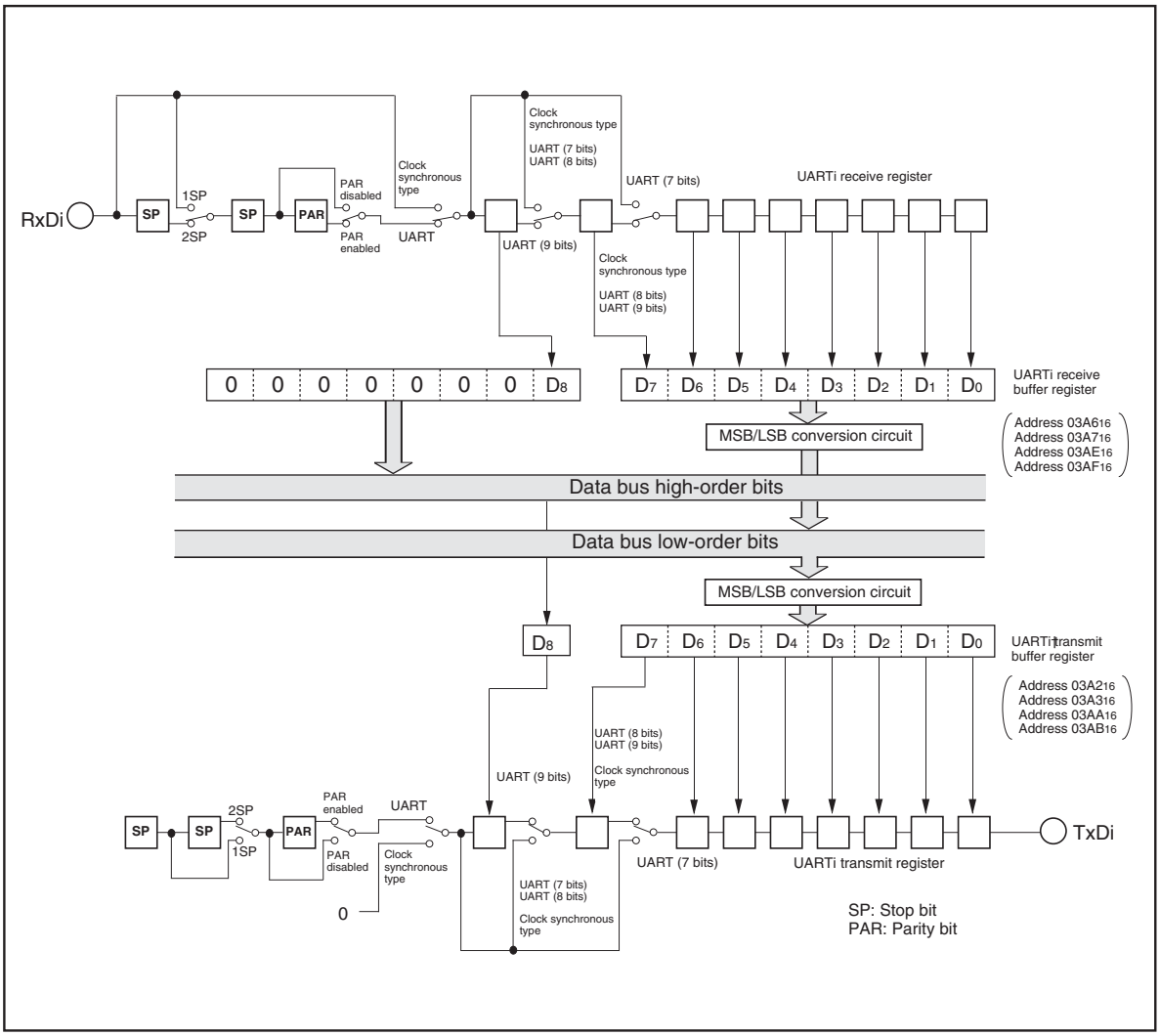


Figure 1.15.2. Block diagram of UARTi (i = 0, 1) transmit/receive unit

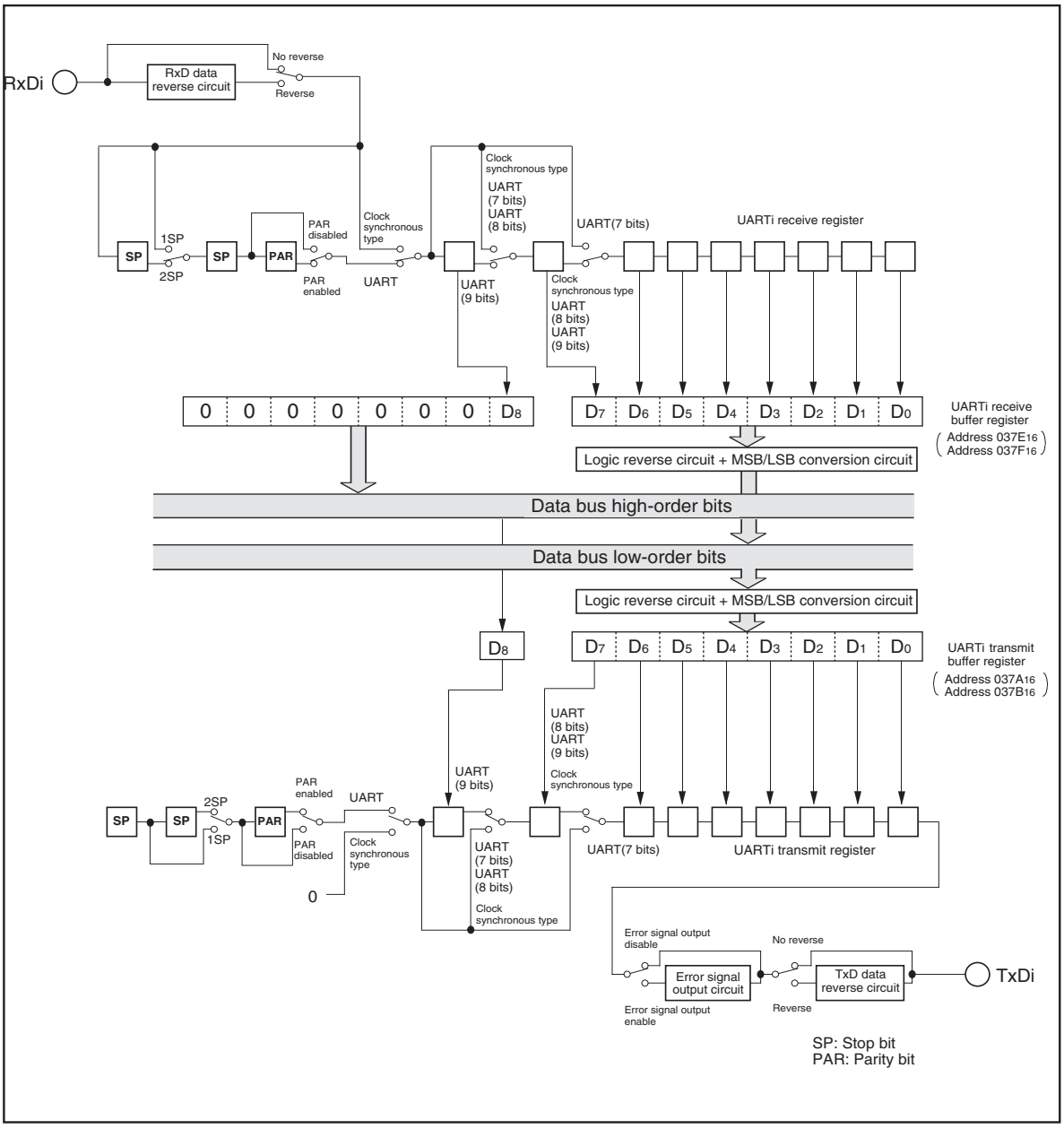
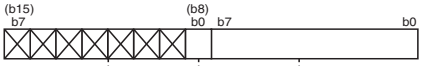


Figure 1.15.3. Block diagram of UART2 transmit/receive unit

UARTi transmit buffer register (Note)

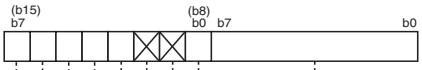


Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate
U2TB	$037B_{16}$, $037A_{16}$	Indeterminate

Function	R	W
Transmit data	×	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.	—	—

Note: Use MOV instruction to write to this register.

UARTi receive buffer register

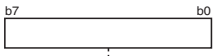


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate
U2RB	$037F_{16}$, $037E_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	———	Receive data	Receive data	○	×
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .				—	—
ABT	Arbitration lost detecting flag (Note 2)	0 : Not detected 1 : Detected	Invalid	○	○
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note 1)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note 1)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note 1)	Invalid	0 : No error 1 : Error found	○	×

Note 1: Bits 15 through 12 are set to 0 when the serial I/O mode select bit (bits 2 to 0 at addresses $03A0_{16}$, $03A8_{16}$ and 0378_{16}) are set to 000_2 or the receive enable bit is set to 0 .
(Bit 15 is set to 0 when bits 14 to 12 all are set to 0 .) Bits 14 and 13 are also set to 0 when the lower byte of the UARTi receive buffer register (addresses $03A6_{16}$, $03AE_{16}$ and $037E_{16}$) is read out.
Note 2: Arbitration lost detecting flag is allocated to U2RB and only "0" must be written. Nothing is assigned in bit 11 of U0RB and U1RB. Set to "0" when writing. If read, value is "0".

UARTi bit rate generator (Note 1, 2)



Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate
U2BRG	0379_{16}	Indeterminate

Function	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	×	○

Note 1: Write a value to this register while transmit/receive halts.
Note 2: Use MOV instruction to write to this register.

Figure 1.15.4. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol: UiMR(i=0,1) — Address: 03A0$_{16}$, 03A8$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bits	Must be fixed to 001	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1		b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited		○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock (Note 1)	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
Reserved		Must always be "0"		○	○

UARTi transmit/receive control register 0

b7	b6	b5	b4	b3	b2	b1	b0

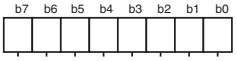
Symbol: UiC0(i=0,1) — Address: 03A4$_{16}$, 03AC$_{16}$ — When reset: 08$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bits	b1 b0 0 0 : f_{1SIO} or f_{2SIO} is selected (Note 2) 0 1 : f_{8SIO} is selected 1 0 : f_{32SIO} is selected 1 1 : Inhibited	b1 b0 0 0 : f_{1SIO} or f_{2SIO} is selected (Note 2) 0 1 : f_{8SIO} is selected 1 0 : f_{32SIO} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{CTS}/\overline{RTS}$ function select bit	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 3)	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 3)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{CTS}/\overline{RTS}$ disable bit	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : $\overline{CTS}/\overline{RTS}$ function disabled ($P6_0$ and $P6_4$ function as programmable I/O port)	0 : $\overline{CTS}/\overline{RTS}$ function enabled 1 : $\overline{CTS}/\overline{RTS}$ function disabled ($P6_0$ and $P6_4$ function as programmable I/O port)	○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open-drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open-drain output	○	○
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	Must always be "0"	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Must always be "0"	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: Periperal clock select register bit PCLK1 is used to select between f_{1SIO} and f_{2SIO}.
Note 3: The settings of the corresponding port register and port direction register are invalid.

Figure 1.15.5. Serial I/O-related registers (2)

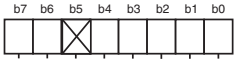
UART2 transmit/receive mode register



Symbol U2MR | Address 037816 | When reset 0016

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 0 : Serial I/O invalid 0 0 1 : Serial I/O 0 1 0 : I²C mode 0 1 1 : ⎫ : ⎬ Must not be set 1 1 1 : ⎭	b2 b1 b0 0 0 0 : Serial I/O invalid 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long Except the above : Must not be set	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 1)	0 : Internal clock 1 : External clock (Note 1)	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = 1 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
IOPOL	TxD, RxD I/O polarity reverse bit	0 : No reverse 1 : Reverse Usually set to 0	0 : No reverse 1 : Reverse Usually set to 0	○	○

UART2 transmit/receive control register 0



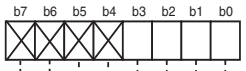
Symbol U2C0 | Address 037C16 | When reset 0816

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f1SIO or f2SIO is selected (Note 2) 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Must not be set.	b1 b0 0 0 : f1SIO or f2SIO is selected (Note 2) 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Must not be set.	○	○
CLK1				○	○
CRS	CTS/RTS function select bit	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 3)	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 3)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	CTS/RTS disable bit	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (P60, P64 and P73 function as programmable I/O port	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (P60, P64 and P73 function as programmable I/O port)	○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be "0".				–	–
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	Must always be "0"	○	○
UFORM	Transfer format select bit (Note 4)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: Peripheral clock select register bit PCLK1 is used to select between f1SIO and f2SIO.
Note 3: The settings of the corresponding port register and port direction register are invalid.
Note 4: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.

Figure 1.15.6. Serial I/O-related registers (3)

UARTi transmit/receive control register 1 (i=0, 1)



Symbol: UiC1(i=0,1) | Address: 03A5$_{16}$,03AD$_{16}$ | When reset: 02$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	O	O
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	O	X
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	O	O
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	O	X
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .				–	–

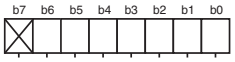
UART2 transmit/receive control register 1

Symbol: U2C1 | Address: 037D$_{16}$ | When reset: 02$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	O	O
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	O	X
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	O	O
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	O	X
U2IRS	UART2 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	O	O
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Must always be "0"	O	O
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	0 : No reverse 1 : Reverse	O	O
U2ERE	Error signal output enable bit	Must be fixed to 0	0 : Output disabled 1 : Output enabled	O	O

Figure 1.15.7. Serial I/O-related registers (4)

UART transmit/receive control register 2

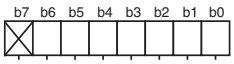


Symbol: UCON | Address: 03B0$_{16}$ | When reset: X0000000$_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Must always be 0	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Must always be 0	○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = 1 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note)	0 : Normal mode (CLK output is CLK1 only) 1 : Transfer clock output from multiple pins function selected	Must always be 0	○	○
RCSP	Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ bit	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ shared pin 1 : $\overline{\text{CTS}}$/ $\overline{\text{RTS}}$ separated	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ shared pin 1 : $\overline{\text{CTS}}$/ $\overline{\text{RTS}}$ separated	○	○
Nothing is assigned. In an attempt to write to this bit, write 0 . The value, if read, turns out to be indeterminate.				–	–

Note: When using multiple pins to output the transfer clock, the following requirements must be met:
* UART1 internal/external clock select bit (bit 3 at address 03A8$_{16}$) = 0 .

UART2 special mode register



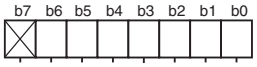
Symbol: U2SMR | Address: 037716 | When reset: X0000000$_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
IICM	I^{2}C mode select bit	0 : Normal mode 1 : I^{2}C mode	Must always be 0	○	○
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	Must always be 0	○	○
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected	Must always be 0	○	○ (Note1)
$\overline{\text{(b3)}}$	Reserved bit	Must always be 0.	Must always be 0.	○	○
ABSCS	Bus collision detect sampling clock select bit	Must always be 0	0 : Rising edge of transfer clock 1 : Underflow signal of timer Aj (Note 2)	○	○
ACSE	Auto clear function select bit of transmit enable bit	Must always be 0	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition select bit	Must always be 0	0 : Ordinary 1 : Falling edge of RxD$_i$	○	○
Nothing is assigned. In an attempt to write to this bit, write 0 . The value, if read, turns out to be indeterminate.				–	–

Note 1: Nothing but 0 may be written.
Note 2: Underflow signal of timer A0 in UART2.

Figure 1.15.8. Serial I/O-related registers (5)

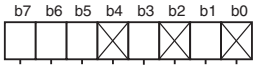
UART2 special mode register 2



Symbol U2SMR2 | Address 0376 16 | When reset X0000000 2

Bit symbol	Bit name	Function	R	W
IICM2	I2C mode select bit 2	Refer to Table 1.15.14	O	O
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	O	O
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	O	O
ALS	SDA output stop bit	0 : Disabled 1 : Enabled	O	O
STAC	UARTi initialization bit	0 : Disabled 1 : Enabled	O	O
SWC2	SCL wait output bit 2	0: UART2 clock 1: 0 output	O	O
SDHI	SDA output disable bit	0: Enabled 1: Disabled (high impedance)	O	O
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			—	—

UART2 special mode register 3



Symbol U2SMR3 | Address 0375 16 | When reset 000X0X0X 2

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			—	—
CKPH	Clock phase set bit	0 : Without clock delay 1 : With clock delay	O	O
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			—	—
NODC	Clock output select bit	0 : CLKi is CMOS output 1 : CLKi is N-channel open drain output	O	O
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be indeterminate.			—	—
DL0	SDAi (TxDi) digital delay setup bit (Note 1, Note 2)	b7 b6 b5 0 0 0 : Without delay 0 0 1 : 1 to 2 cycle(s) of BRG count source 0 1 0 : 2 to 3 cycles of BRG count source 0 1 1 : 3 to 4 cycles of BRG count source 1 0 0 : 4 to 5 cycles of BRG count source 1 0 1 : 5 to 6 cycles of BRG count source 1 1 0 : 6 to 7 cycles of BRG count source 1 1 1 : 7 to 8 cycles of BRG count source	O	O
DL1			O	O
DL2			O	O

Note 1 : These bits are used for SDAi (TxDi)output digital delay when using UARTi for I2C interface. Otherwise,must set to 000 2 .
Note 2 : The amount of delay varies with the load on SCL and SDA pins. Also, when using an external clock, the amount of delay increases by about 100 ns, so be sure to take this into account when using the device.

Figure 1.15.9. Serial I/O-related registers (6)

UART2 special mode register 4

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
U2SMR4	0374 16	00 16

Bit symbol	Bit name	Function	R	W
STAREQ	Start condition generate bit (Note)	0 : Clear 1 : Start	○	○
RSTAREQ	Restart condition generate bit (Note)	0 : Clear 1 : Start	○	○
STPREQ	Stop condition generate bit (Note)	0 : Clear 1 : Start	○	○
STSPSEL	SCL,SDA output select bit	0 : Ordinal block 1 : Start/stop condition generate block	○	○
ACKD	ACK data bit	0 : ACK 1 : NACK	○	○
ACKC	ACK data output enable bit	0 : SI/O data output 1 : ACKD output	○	○
SCLHI	SCL output stop enable bit	0 : Disabled 1 : Enabled	○	○
SWC9	Final bit L hold enable bit	0 : SCL L hold disabled 1 : SCL L hold enabled	○	○

Note: When start condition is generated,these bits automatically become 0 .

Figure 1.15.10. Serial I/O-related registers (7)

Clock Synchronous Serial I/O Mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 1.15.2 and 1.15.3 list the specifications of the clock synchronous serial I/O mode. Figure 1.15.11 shows the UARTi transmit/receive mode register.

Table 1.15.2. Specifications of clock synchronous serial I/O mode (1)

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock 037816	• When internal clock is selected (bit 3 at addresses $03A016$, $03A816$, 037816 = "0") : fi/ 2(n+1) (Note 1) fi = f1SIO, f2SIO, f8SIO, f32SIO • When external clock is selected (bit 3 at addresses $03A016$, $03A816$, = "1") : Input from CLKi pin
Transmission/reception control	• CTS function, RTS function, CTS and RTS function disabled: selectable
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at addresses $03A516$, $03AD16$, $037D16$) = "1" – Transmit buffer empty flag (bit 1 at addresses $03A516$, $03AD16$, $037D16$) = "0" – When CTS function selected, CTS input level = "L" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses $03A416$, $03AC16$, $037C16$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses $03A416$, $03AC16$, $037C16$) = "1": CLKi input level = "L"
Reception start condition	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at addresses $03A516$, $03AD16$, $037D16$) = "1" – Transmit enable bit (bit 0 at addresses $03A516$, $03AD16$, $037D16$) = "1" – Transmit buffer empty flag (bit 1 at addresses $03A516$, $03AD16$, $037D16$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses $03A416$, $03AC16$, $037C16$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses $03A416$, $03AC16$, $037C16$) = "1": CLKi input level = "L"
Interrupt request generation timing UARTi	• When transmitting – Transmit interrupt cause select bit (bits 0, 1 at address $03B016$, bit 4 at address $037D16$) = "0": Interrupts requested when data transfer from transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bits 0, 1 at address $03B016$, bit 4 at address $037D16$) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving – Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 2) Generated 7 clock periods after the device started receiving the next data before reading out the contents of the UARTi receive buffer register.

Note 1: "n" denotes the value 0016 to $FF16$ that is set in the UART bit rate generator.

Note 2: If an overrun error occurs, the UARTi receive buffer will have the next data written in. In addition, the UARTi receive interrupt request bit does not change.

Table 1.15.3. Specifications of clock synchronous serial I/O mode (2)

Item	Specification
Select function	• CLK polarity selection Whether transmit data output/input timing is at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled automatically after the receive buffer register is read • Switching serial data logic (UART2) Whether to invert data (1's complement) when writing to the transmission buffer register or reading the reception buffer register can be selected. • TxD, RxD I/O polarity reverse (UART2) This function inverts the TxD port output and RxD port input. All I/O data level is reversed. • Transfer clock output from multiple pins selection (UART1) UART1 transfer clock can be chosen by software to be output from one of the two pins set

Table 1.15.4. Registers used in clock synchronous serial I/O mode and the register values set

Register	Bit	Function
UiTB(Note3)	0 to 7	Set transmission data
UiRB(Note3)	0 to 7	Reception data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set a transfer rate
UiMR(Note3)	SMD2 to SMD0	Set to "001$_2$"
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the CTS or RTS function
	NCH	Select TxDi pin output mode (Note 2)
	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 (Note 1)	Select the source of UART2 transmit interrupt
	U2RRM (Note 1)	Set this bit to "1" to use continuous receive mode
	UiLCH	Set this bit to "1" to use inverted data logic
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 2	Set to "0"
	NODC	Select clock output mode
	4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the 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 = 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 CTS$_0$ signal from the P6$_4$ pin
	7	Set to "0"

Note 1: Set the U0C1 and U1C1 register bit 4 and bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are in the UCON register.

Note 2: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

Note 3: Not all register bits are described above. Set those bits to "0" when writing to the registers in clock synchronous serial I/O mode.

i=0 to 2

UARTi transmit/receive mode register (i=0 to 1)

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

Symbol: UiMR(i=0 to 1) — Address: 03A0$_{16}$, 03A8$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
Reserved	Must always be "0".		○	○

Note: Set the corresponding port direction register to "0".

UART2 transmit/receive mode register

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

Symbol: U2MR — Address: 0378 16 — When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 2)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note 1)	0 : No reverse 1 : Reverse	○	○

Note 1: Usually set to 0 .
Note 2: Set the corresponding port direction register to 0 .

Figure 1.15.11. UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 1.15.5 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions when the transfer clock output from multiple pins function is not selected. 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 is selected, this pin is in floating state.)

Table 1.15.5. Input/output pin functions in clock synchronous serial I/O mode
(when transfer clock output from multiple pins is not selected)

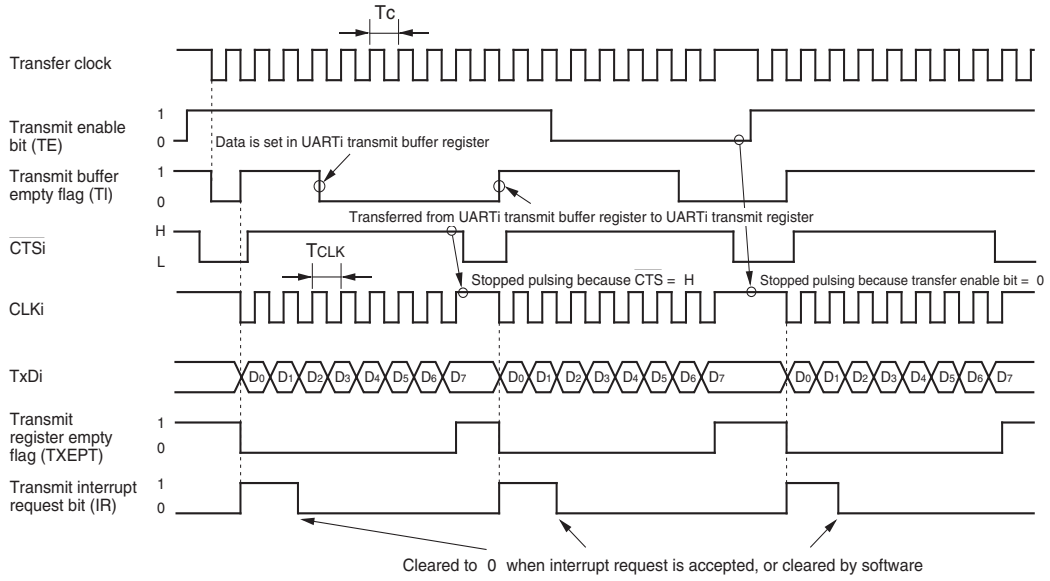
Pin name	Function	Method of selection
TxDi (P6$_3$, P6$_7$, P7$_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	Port P6$_2$, P6$_6$ and P7$_1$ direction register (bits 2 and 6 at address 03EE$_{16}$, bit 1 at address 03EF$_{16}$)= 0 (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$) = 0
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$) = 1 Port P6$_1$, P6$_5$ and P7$_2$ direction register (bits 1 and 5 at address 03EE$_{16}$, bit 2 at address 03EF$_{16}$) = 0
$\overline{CTS}$i/$\overline{RTS}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{CTS}$ input	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 0 CTS/RTS function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 0 Port P6$_0$, P6$_4$ and P7$_3$ direction register (bits 0 and 4 at address 03EE$_{16}$, bit 3 at address 03EF$_{16}$) = 0
	$\overline{RTS}$ output	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 0 $\overline{CTS}$/$\overline{RTS}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 1
	Programmable I/O port	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 1

Table 1.15.6. P6$_4$ pin function in clock synchronous serial I/O mode

Pin function	Bit set value					
	U1C0 register		UCON register			PD6 register
	CRD	CRS	RCSP	CLKMD1	CLKMD0	PD6_4
P6$_4$	1	—	0	0	—	Input: 0, Output: 1
$\overline{CTS}_1$	0	0	0	0	—	0
$\overline{RTS}_1$	0	1	0	0	—	—
$\overline{CTS}_0$(Note1)	0	0	1	0	~~0~~	
CLKS$_1$	—	—	—	1(Note 2)	1	—

Note 1: In addition to this, set the U0C0 register's CRD bit to "0" ($\overline{CTS}_0$/$\overline{RTS}_0$ enabled) and the U0 C0 register's CRS bit to "1" ($\overline{RTS}_0$ selected).
Note 2: When the CLKMD1 bit = 1 and the CLKMD0 bit = 0, the following logic levels are output:
- High if the U1C0 register's CLKPOL bit = 0
- Low if the U1C0 register's CLKPOL bit = 1

• Example of transmit timing (when internal clock is selected)

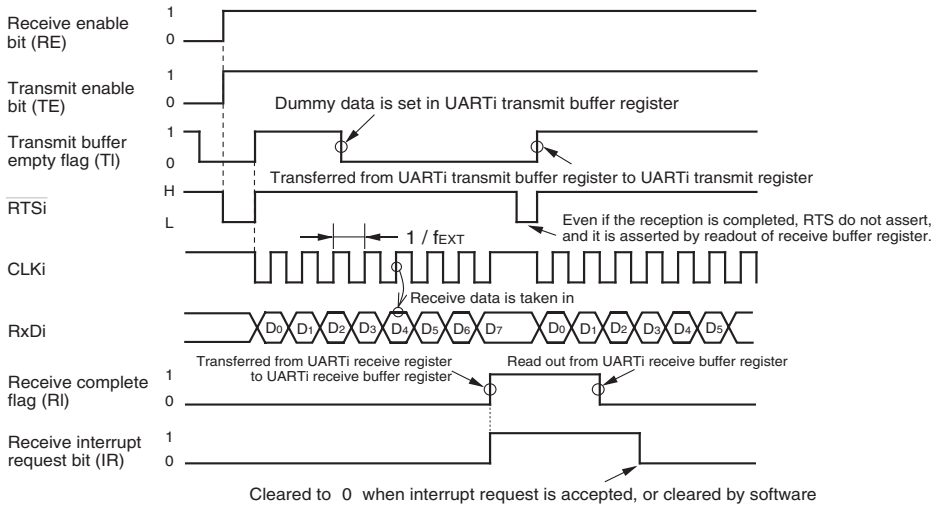


Shown in ( ) are bit symbols.

The above timing applies to the following settings:
* Internal clock is selected.
* CTS function is selected
* CLK polarity select bit = 0 .
* Transmit interrupt cause select bit = 0 .

Tc = TCLK = 2(n + 1) / fi
fi: frequency of BRGi count source (f1, f8, f32)
n: value set to BRGi

• Example of receive timing (when external clock is selected)



Shown in ( ) are bit symbols.

The above timing applies to the following settings:
* External clock is selected
* RTS function is selected
* CLK polarity select bit = 0 .

fEXT: frequency of external clock

Meet the following conditions are met when the CLK input before data reception = H
* Transmit enable bit→ 1
* Receive enable bit→ 1
* Dummy data write to UARTi transmit buffer register

Figure 1.15.12. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 1.15.13, the CLK polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) allows selection of the polarity of the transfer clock.

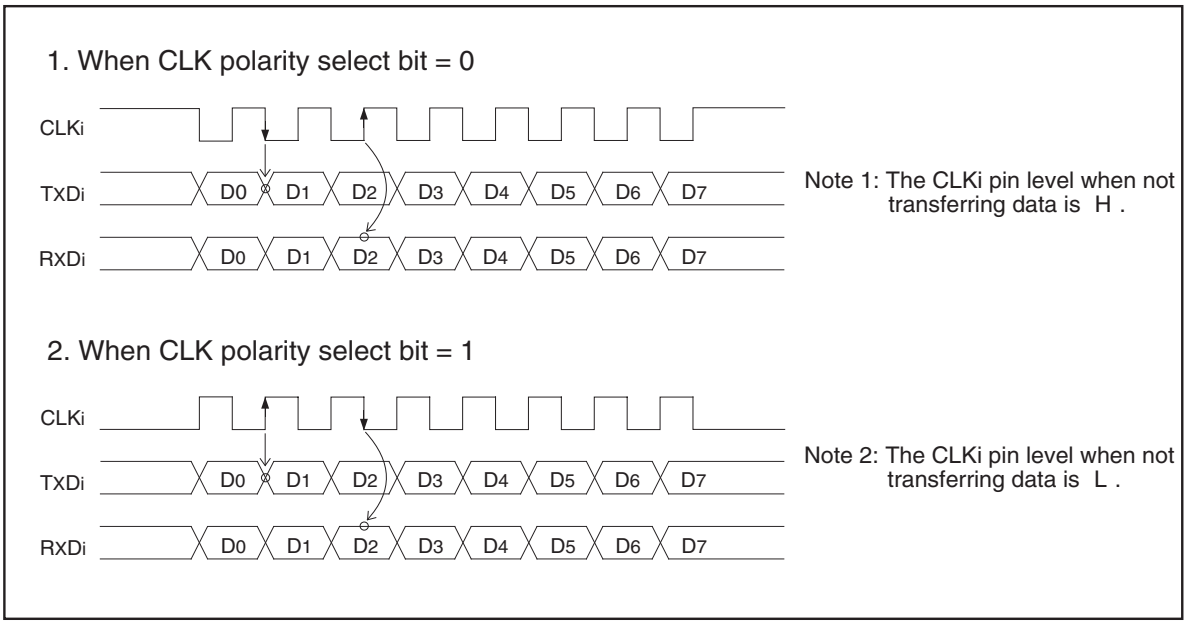


Figure 1.15.13. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.15.14, when the transfer format select bit (bit 7 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".

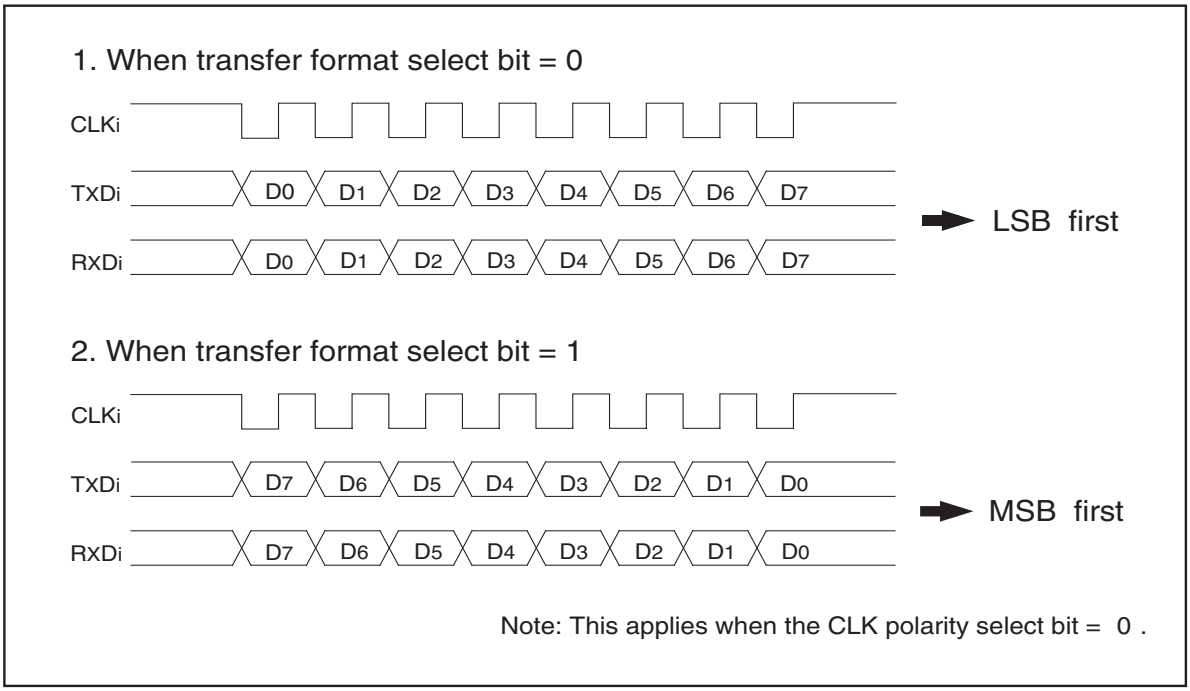


Figure 1.15.14. Transfer format

(c) Transfer clock output from multiple pins function (UART1)

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address 03B0$_{16}$). (See Figure 1.15.15.) The multiple pins function is valid only when the internal clock is selected for UART1.

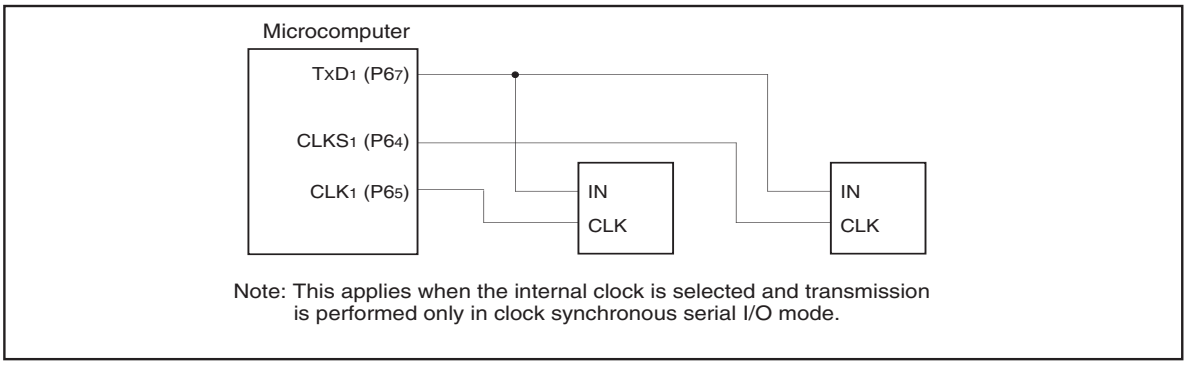


Figure 1.15.15. The transfer clock output from the multiple pins function usage

(d) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address 03B0$_{16}$, bit 5 at address 037D$_{16}$) is set to "1", the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the SIO simultaneously goes to a receive enable state without having to write dummy data to the transmit buffer register back again.

(e) Serial data logic switch function (UART2)

When the data logic select bit (bit6 at address 037D$_{16}$) = "1", and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 1.15.16 shows the example of serial data logic switch timing.

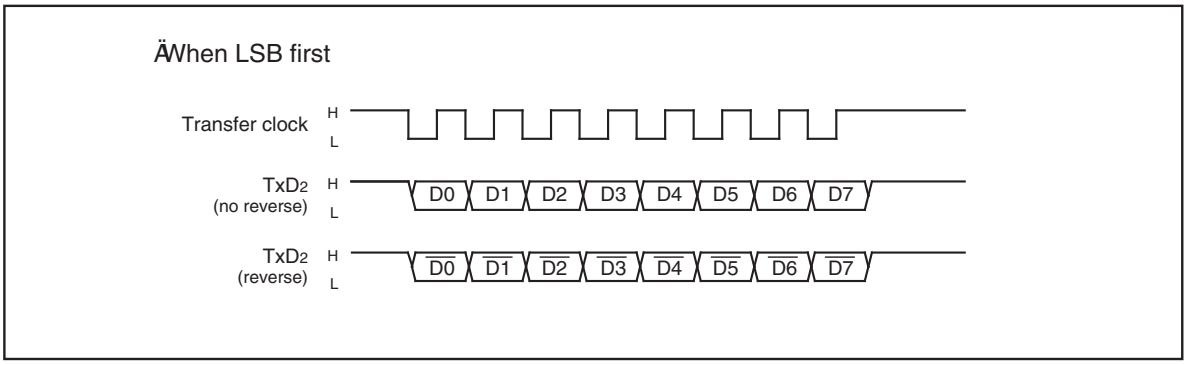


Figure 1.15.16. Serial data logic switch timing

(f) $\overline{CTS}/\overline{RTS}$ 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. To use this function, set the register bits as shown below.

- U0C0 register's CRD bit = 0 (enables UART0 $\overline{CTS}/\overline{RTS}$)
- U0C0 register's CRS bit = 1 (outputs UART0 $\overline{RTS}$)
- U1C0 register's CRD bit = 0 (enables UART1 $\overline{CTS}/\overline{RTS}$)
- U1C0 register's CRS bit = 0 (inputs UART1 $\overline{CTS}$)
- UCON register's RCSP bit = 1 (inputs $\overline{CTS}_0$ from the $P6_4$ pin)
- UCON register's CLKMD1 bit = 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.

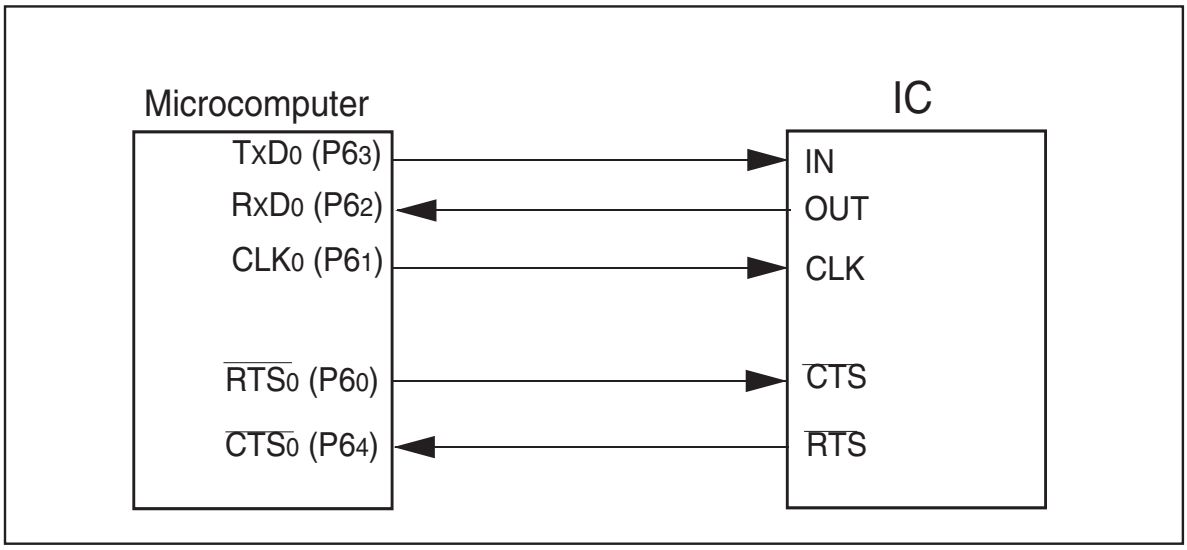


Figure 1.15.17. $\overline{CTS}/\overline{RTS}$ separate function usage

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 1.15.7 and 1.15.8 list the specifications of the UART mode. Figure 1.15.18 shows the UARTi transmit/receive mode register.

Table 1.15.7. Specifications of UART Mode (1)

Item	Specification
Transfer data format	• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ = "0") : fi/16 (n+1) (Note 1) fi = f_{1SIO}, f_{2SIO}, f_{8SIO}, f_{32SIO} • When external clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ ="1") : f_{EXT}/16(n+1) (Note 1) (Note 2)
Transmission/reception control	• CTS function, RTS function, CTS and RTS function disabled: selectable
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "1" - Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "0" - When CTS function selected, CTS input level = "L"
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = "1" - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0,1 at address 03B0$_{16}$, bit4 at address 037D$_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed - Transmit interrupt cause select bits (bits 0, 1 at address 03B0$_{16}$, bit4 at address 037D$_{16}$) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving - Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 3) Generated 6.5, 7, or 8.5 clock periods after the device started receiving the next data before reading out the contents of the UARTi receive buffer register. • 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's in parity and character bits does not match the number of 1's set • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered

Note 1: 'n' denotes the value 00$_{16}$ to FF$_{16}$ that is set to the UARTi bit rate generator.

Note 2: f_{EXT} is input from the CLKi pin.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit does not change.

Table 1.15.8. Specifications of UART Mode (2)

Item	Specification
Select function	• Serial data logic switch (UART2) This function is reversing logic value of transferring data. Start bit, parity bit and stop bit are not reversed. • TxD, RxD I/O polarity switch (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

Table 1.15.9 Registers used in UART mode and the register values set

Register	Bit	Function
UiTB	0 to 8	Set transmission data (Note 1)
UiRB	0 to 8	Reception data can be read (Note 1)
	OER,FER,PER,SUM	Error flag
UiBRG	---	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	Select the TxD/RxD input/output polarity
UiC0	CLK0, CLK1	Select the count source for the UiBRG register
	CRS	Select CTS or RTS to use
	TXEPT	Transmit register empty flag
	CRD	Enable or disable the CTS or RTS function
	NCH	Select TxDi pin output mode (Note 2)
	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 (Note 2)	Select the source of UART2 transmit interrupt
	U2RRM (Note 2)	Set to "0"
	UiLCH	Set this bit to "1" to use inverted data logic
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
UCON	U0IRS, U1IRS	Select the source of UART0/UART1 transmit interrupt
	U0RRM, U1RRM	Set to "0"
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1	Set to "0"
	RCSP	Set this bit to "1" to accept as input the UART0 CTS$_0$ signal from the P6$_4$ pin
	7	Set to "0"

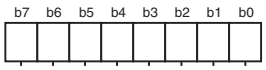
Note 1: The bits used for transmit/receive data are as follows: Bit 0 to bit 6 when transfer data is 7 bits long; bit 0 to bit 7 when transfer data is 8 bits long; bit 0 to bit 8 when transfer data is 9 bits long.

Note 2: Set the U0C1 and U1C1 registers bit 4 to bit 5 to "0". The U0IRS, U1IRS, U0RRM and U1RRM bits are included in the UCON register.

Note 3: TxD2 pin is N channel open-drain output. Set the U2C0 register's NCH bit to "0".

i=0 to 2

UARTi transmit / receive mode registers



Symbol: UiMR(i=0,1) | Address: $03A0_{16}$, $03A8_{16}$ | When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	O	O
SMD1			O	O
SMD2			O	O
CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock (Note)	O	O
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	O	O
PRY	Odd / even parity select bit	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	O	O
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	O	O
Reserved	Must always be "0".		O	O

Note: Set the corresponding port direction register to "0".

UART2 transmit / receive mode register

Symbol: U2MR | Address: 0378_{16} | When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	O	O
SMD1			O	O
SMD2			O	O
CKDIR	Internal / external clock select bit	Must always be fixed to "0"	O	O
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	O	O
PRY	Odd / even parity select bit	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	O	O
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	O	O
IOPOL	TxD, RxD I/O polarity reverse bit (Note)	0 : No reverse 1 : Reverse	O	O

Note: Usually set to "0".

Figure 1.15.18. UARTi transmit/receive mode register in UART mode

Table 1.15.10 lists the functions of the input/output pins 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 is selected, this pin is in floating state.)

Table 1.15.10. Input/output pin functions in UART mode

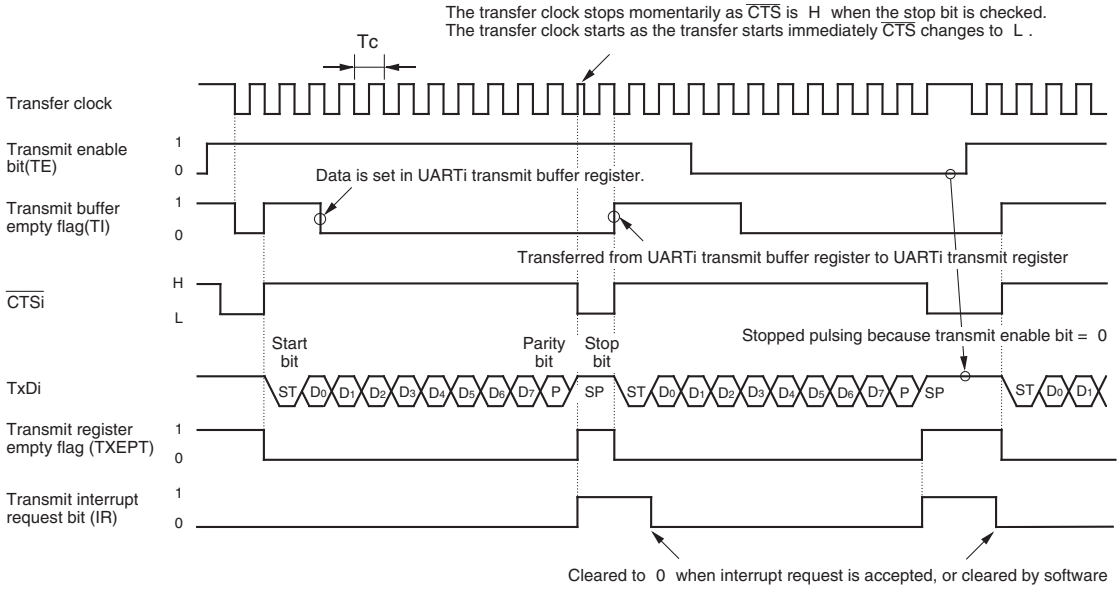
Pin name	Function	Method of selection
TxDi (P6$_3$, P6$_7$, P7$_0$)	Serial data output	
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	Port P6$_2$, P6$_6$ and P7$_1$ direction register (bits 2 and 6 of address 03EE$_{16}$, bit 1 of address 03EF$_{16}$)= 0 (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Programmable I/O port	Internal/external clock select bit (bit 3 of address 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$) = 0
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$) = 1 Port P6$_1$, P6$_5$ and P7$_2$ direction register (bits 1 and 5 of address 03EE$_{16}$) = 0 (Do not use external for UART2)
$\overline{CTS}$i/$\overline{RTS}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{CTS}$ input	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 of address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 0 $\overline{CTS}$/$\overline{RTS}$ function select bit (bit 2 of address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 0 Port P6$_0$, P6$_4$ and P7$_3$ direction register (bits 0 and 4 of address 03EE$_{16}$, bit 3 at address 03EF$_{16}$) = 0
	$\overline{RTS}$ output	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 of address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 0 $\overline{CTS}$/$\overline{RTS}$ function select bit (bit 2 of address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 1
	Programmable I/O port	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 of address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = 1

Table 1.15.11. P6$_4$ pin function in UART mode

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
$\overline{CTS}_1$	0	0	0	0	0
$\overline{RTS}_1$	0	1	0	0	—
$\overline{CTS}_0$ (Note)	0	0	1	0	0

Note: In addition to this, set the U0C0 register's CRD bit to "0" ($\overline{CTS}_0$/$\overline{RTS}_0$ enabled) and the U0C0 register's CRS bit to "1" ($\overline{RTS}_0$ selected).

• Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)

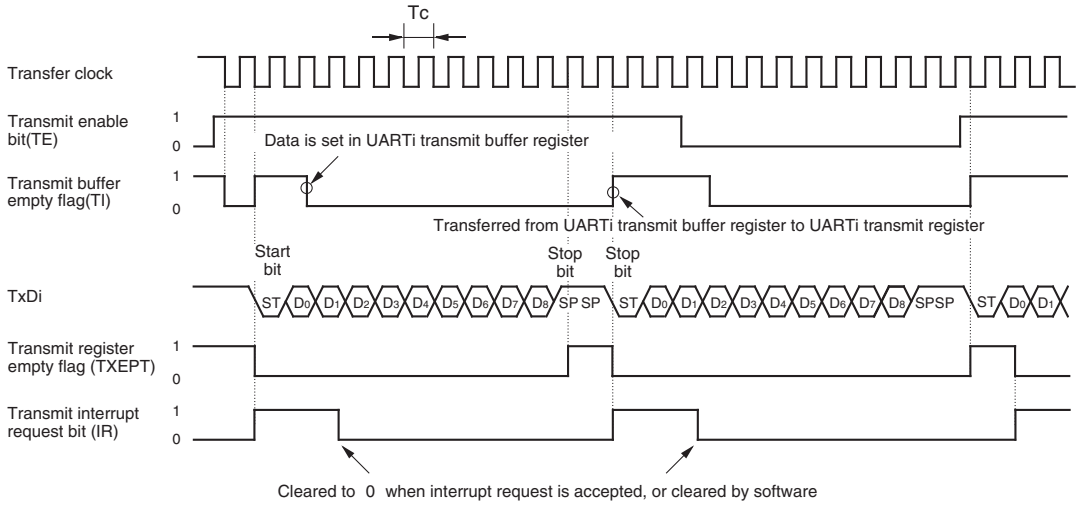


Shown in ( ) are bit symbols.

The above timing applies to the following settings :
* Parity is enabled.
* One stop bit.
* CTS function is selected.
* Transmit interrupt cause select bit = 1 .

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

• Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)



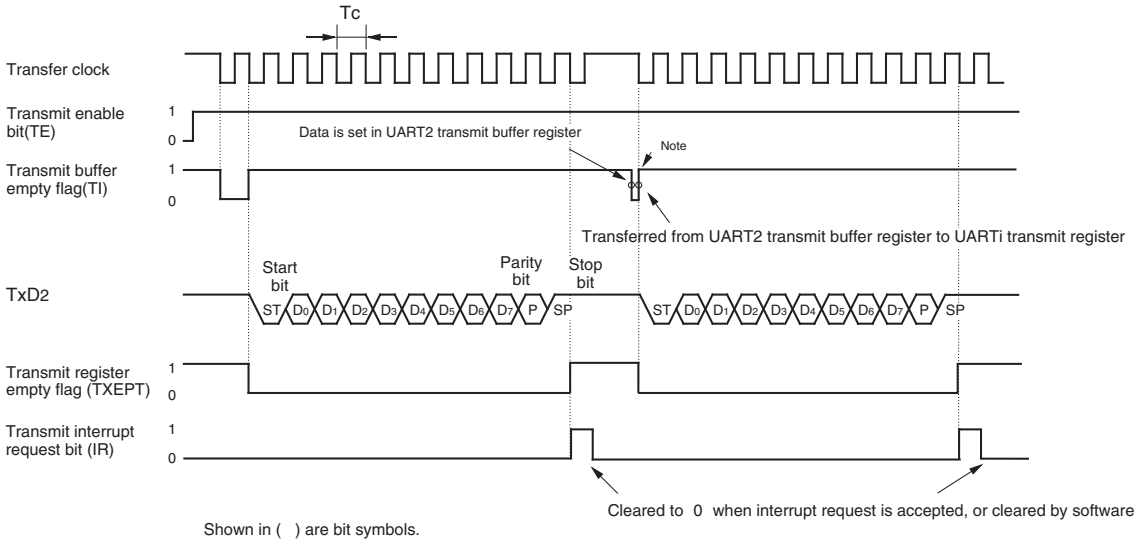
Shown in ( ) are bit symbols.

The above timing applies to the following settings :
* Parity is disabled
* Two stop bits
* CTS function is disabled
* Transmit interrupt cause select bit = 0 .

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 1.15.19. Typical transmit timing in UART mode (UART0, UART1)

• Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)



Shown in ( ) are bit symbols.

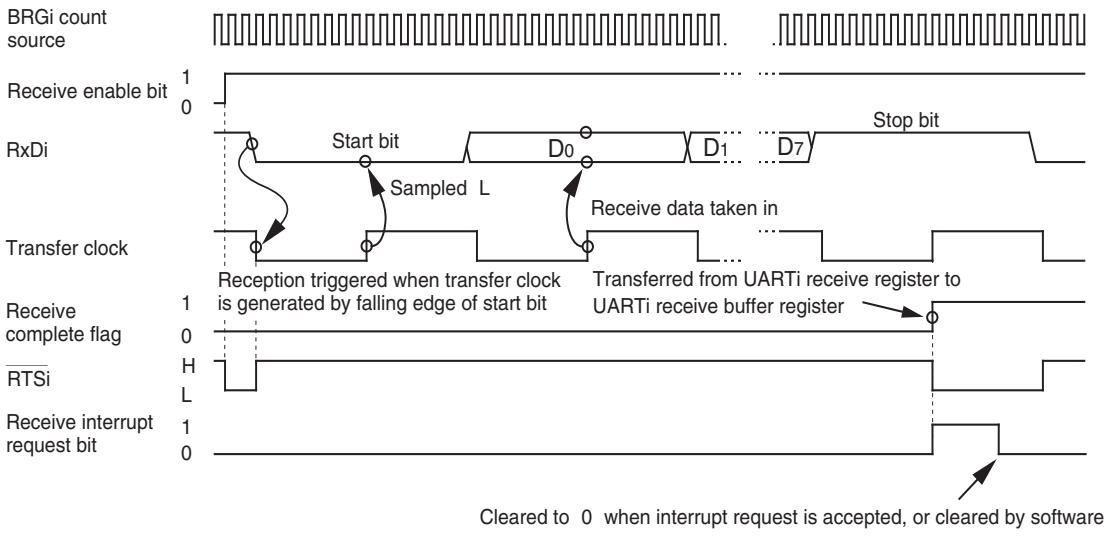
The above timing applies to the following settings :
* Parity is enabled.
* One stop bit.
* Transmit interrupt cause select bit = 1 .

Tc = 16 (n + 1) / fi
fi : frequency of BRG2 count source (f1SIO, f2SIO, f8SIO, f32SIO)
n : value set to BRG2

Note: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.

Figure 1.15.20. Typical transmit timing in UART mode (UART2)

• Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)



The above timing applies to the following settings :
* Parity is disabled.
* One stop bit.
* RTS function is selected.

Figure 1.15.21. Typical receive timing in UART mode

(a) Function for switching serial data logic (UART2)

When the data logic select bit (bit 6 of address 037D16) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 1.15.22 shows the example of timing for switching serial data logic.

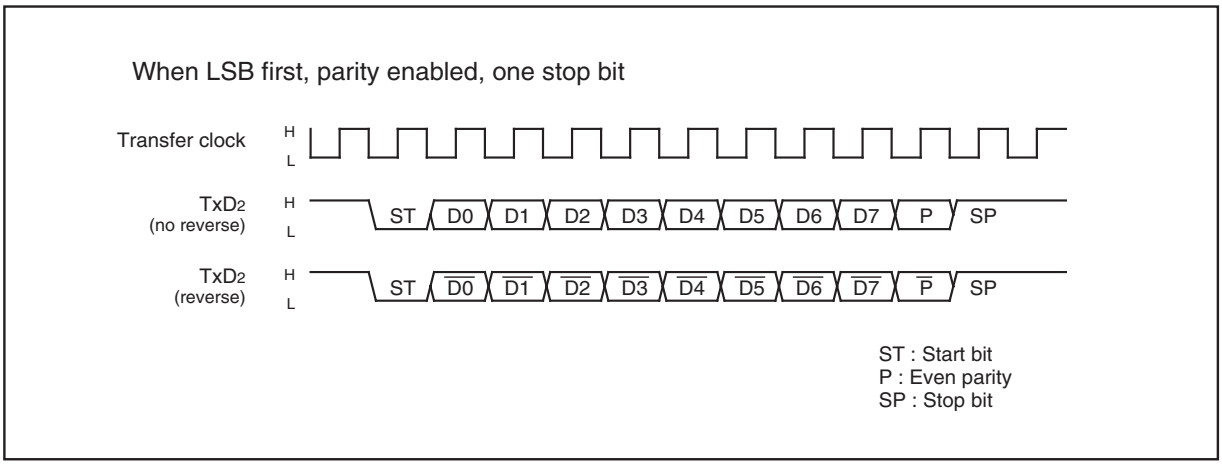


Figure 1.15.22. Timing for switching serial data logic (UART2)

(b) TxD, RxD I/O polarity reverse function (UART2)

This function reverses/inverts TxD pin output and RxD pin input. The level of any data to be input or output (including the start bit, stop bit(s), and parity bit) is reversed. Set this function to "0" (not reversed/inverted) for normal use.

(c) Bus collision detection function (UART2)

This function samples the output level of the TxD pin and the input level of the RxD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Figure 1.15.23 shows the example of detection timing of a bus collision (in UART mode).

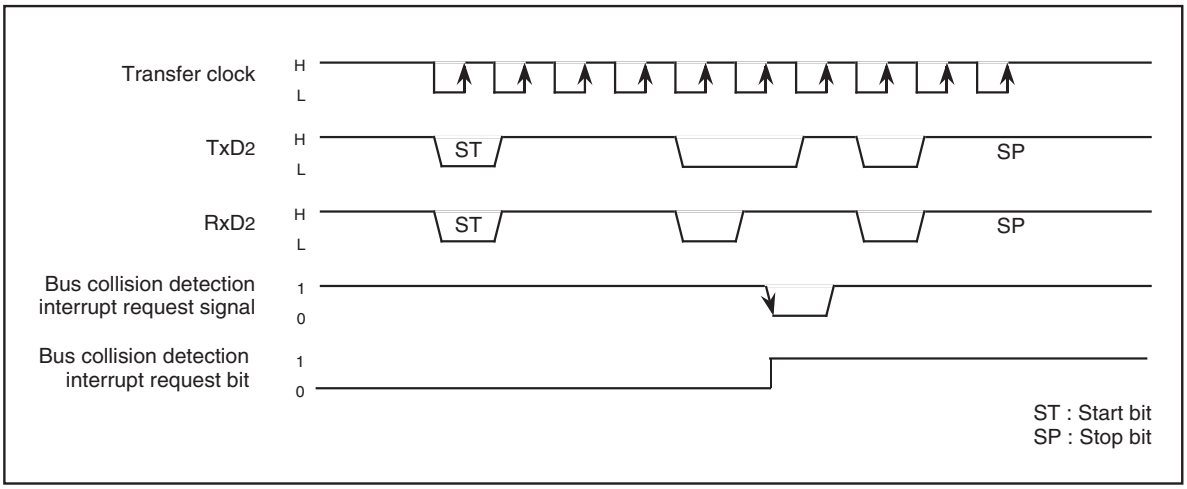


Figure 1.15.23. Detection timing of a bus collision (in UART mode)

Clock-Asynchronous Serial I/O Mode (used for SIM interface)

The SIM (Subscriber Identity Module) interface is used for connecting the microcomputer with a memory card or the like; adding some extra settings in UART2 clock-asynchronous serial I/O mode allows the user to use this function. Table 1.15.12 shows the specifications of clock-asynchronous serial I/O mode (used for SIM interface).

Table 1.15.12. Specifications of clock-asynchronous serial I/O mode (used for SIM interface)

Item	Specification
Transfer data format	• Transfer data 8-bit UART mode (bit 2, bit1, bit 0 of address 0378_{16} = "101_2") • One stop bit (bit 4 of address 037816 = "0") • With the direct format chosen Set parity to "even" (bit 5, bit 6 of address 0378_{16} = "11_2") Set data logic to "direct" (bit 6 of address $037D_{16}$ = "0") Set transfer format to LSB (bit 7 of address $037C_{16}$ = "0") • With the inverse format chosen Set parity to "odd" (bit 5, bit 6 of address 0378_{16} = "01_2") Set data logic to "inverse" (bit 6 of address $037D_{16}$ = "1") Set transfer format to MSB (bit 7 of address $037C_{16}$ = "1")
Transfer clock	• With the internal clock chosen (bit 3 of address 0378_{16} = "0") : fi / 16 (n + 1) (Note 1) : fi = $f1_{SIO}$, $f2_{SIO}$, $f8_{SIO}$, $f32_{SIO}$ (Do not set external clock)
Transmission/reception control Other settings	• Disable the CTS and RTS function (bit 4 of address $037C_{16}$ = "1") • UART2 does not support sleep mode function. • Set transmission interrupt factor to "transmission completed" (bit 4 of address $037D_{16}$ = "1")
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 of address $037D_{16}$) = "1" - Transmit buffer empty flag (bit 1 of address $037D_{16}$) = "0"
Reception start condition	• To start reception, the following requirements must be met: - Reception enable bit (bit 2 of address $037D_{16}$) = "1" - Detection of a start bit
Interrupt request generation timing	• When transmitting When data transmission from the UART2 transmit register is completed (bit 4 of address $037D_{16}$ = "1") • When receiving When data transfer from the UART2 receive register to the UART2 receive buffer register is completed
Error detection	• Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 2) • Framing error (see the specifications of clock-asynchronous serial I/O) • Parity error (see the specifications of clock-asynchronous serial I/O) - On the reception side, an "L" level is output from the TXD2 pin by use of the parity error signal output function (bit 7 of address $037D_{16}$ = "1") when a parity error is detected - On the transmission side, a parity error is detected by the level of input to the RXD2 pin when a transmission interrupt occurs • The error sum flag (see the specifications of clock-asynchronous serial I/O)

Note 1: 'n' denotes the value 00_{16} to FF_{16} that is set to the UART2 bit rate generator.
Note 2: If an overrun error occurs, the UART2 receive buffer will have the next data written in. In addition, the UART2 receive interrupt request bit does not change.

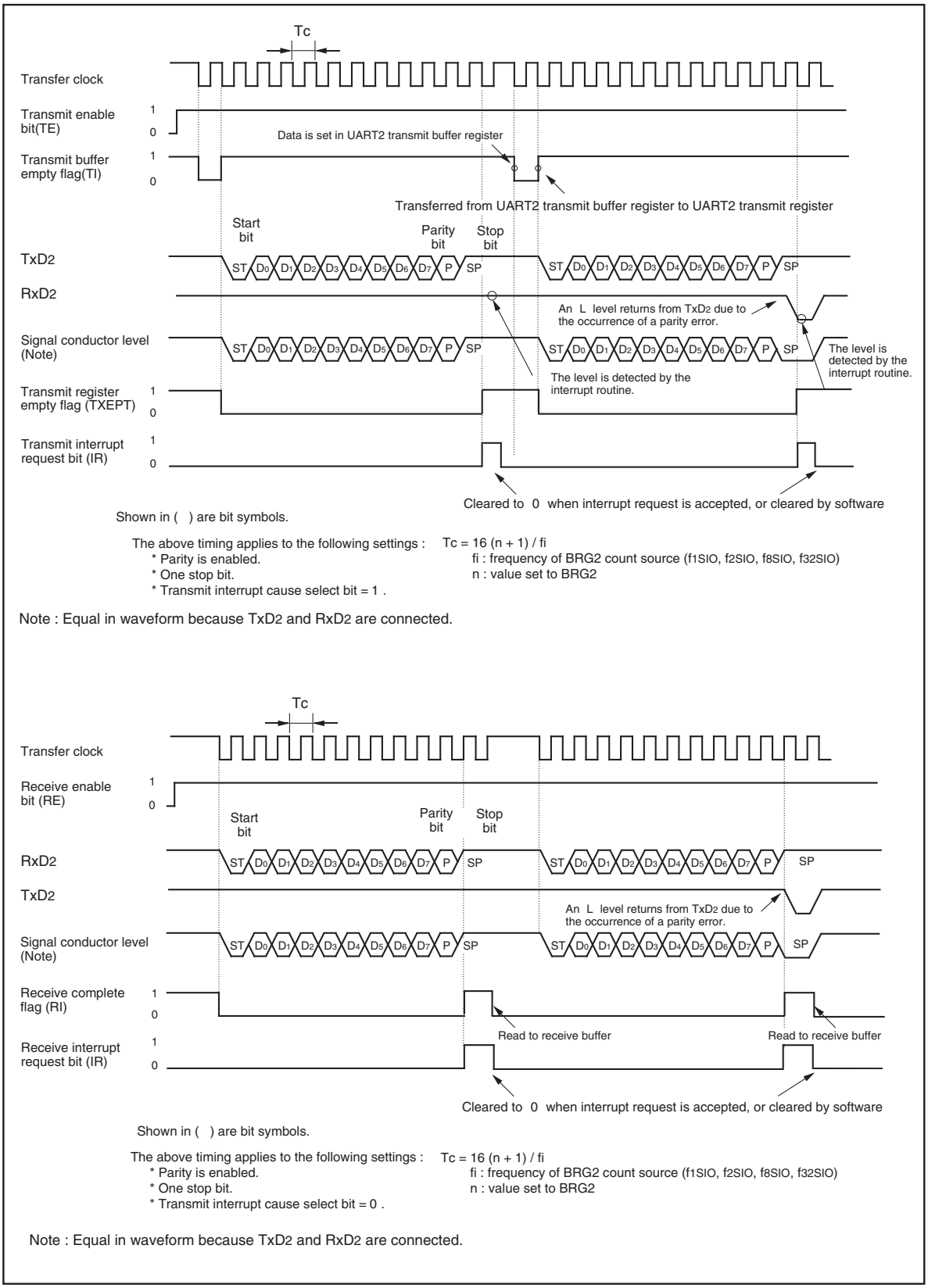


Figure 1.15.24. Typical transmit/receive timing in UART mode (used for SIM interface)

(a) Function to output a parity error signal

During reception, with the error signal output enable bit (bit 7 of address 037D16) assigned "1", you can output an "L" level from the TxD2 pin when a parity error is detected. And during transmission, comparing with the case in which the error signal output enable bit (bit 7 of address 037D16) is assigned "0", the transmission completion interrupt occurs in the half cycle later of the transfer clock. Therefore parity error signals can be detected by a transmission completion interrupt program. Figure 1.15.25 shows the output timing of the parity error signal.

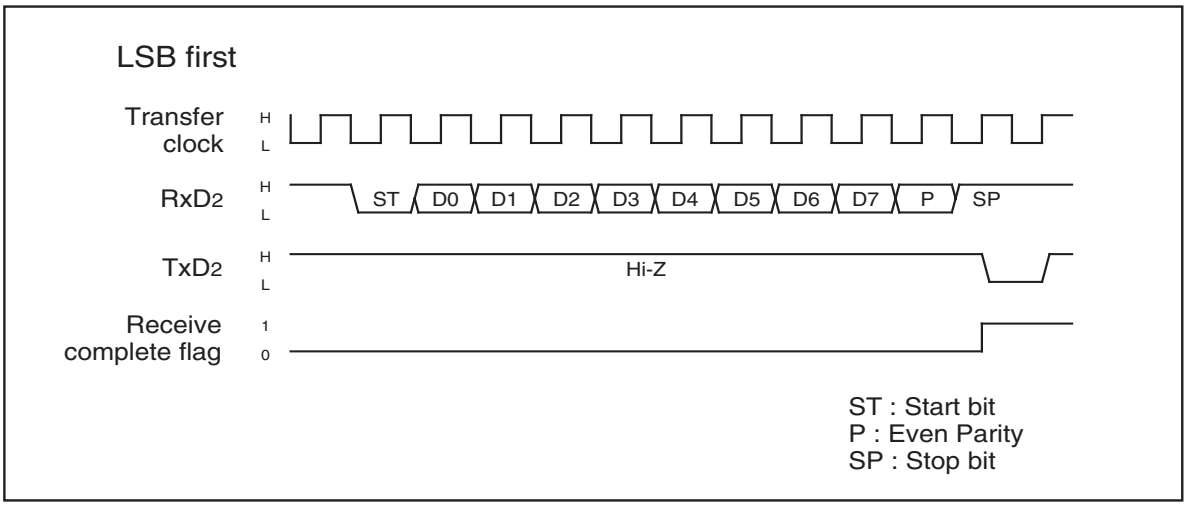


Figure 1.15.25. Output timing of the parity error signal

(b) Direct format/inverse format

Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, D0 data is output from TxD2. If you choose the inverse format, D7 data is inverted and output from TxD2.

Figure 1.15.26 shows the SIM interface format.

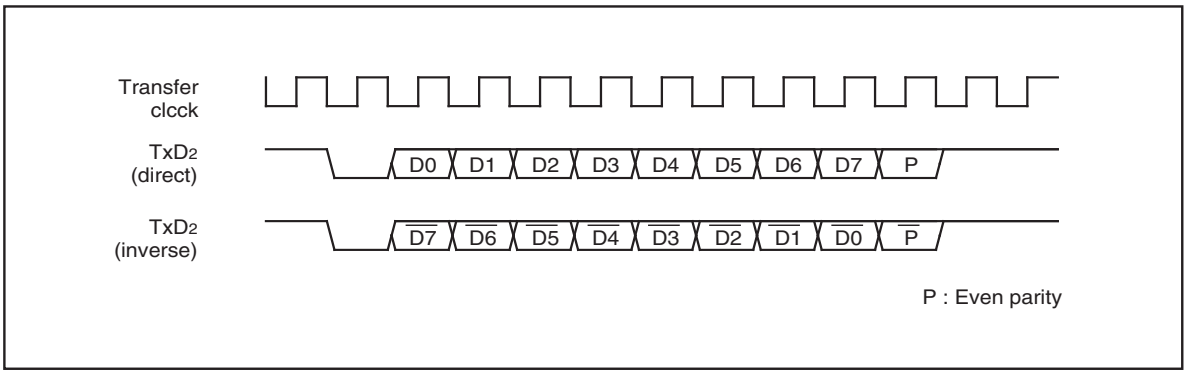


Figure 1.15.26. SIM interface format

Figure 1.15.27 shows the example of connecting the SIM interface. Connect TxD_2 and RxD_2 and apply pull-up.

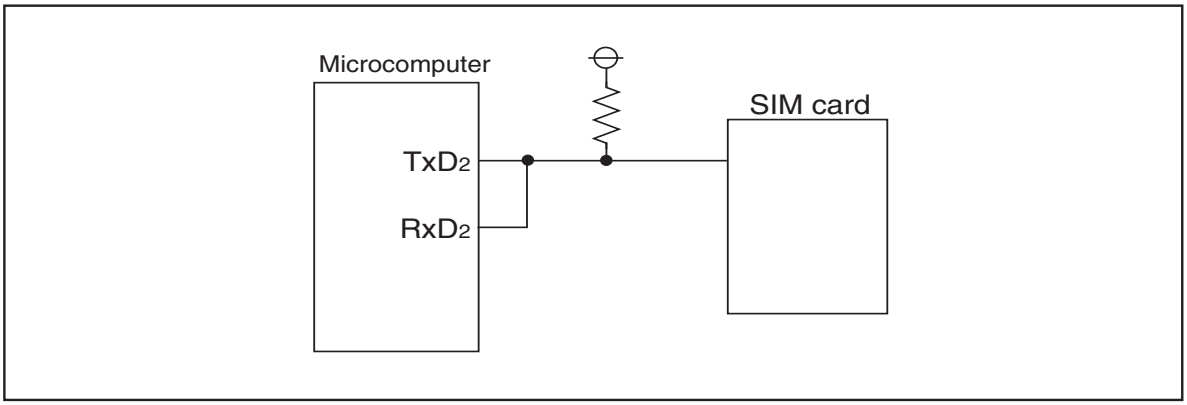


Figure 1.15.27. Connecting the SIM interface

UART2 Special Mode Register

The UART2 special mode register (address 037716) is used to control UART2 in various ways.

Bit 0 of the UART2 special mode register are used as the I^2C mode select bit.

Setting "1" in the I^2C mode select bit (bit 0) goes the circuit to achieve the I^2C bus (simplified I^2C bus) interface effective.

Table 1.15.13 shows the relation between the I^2C mode select bit and respective control workings.

Since this function uses clock-synchronous serial I/O mode, set this bit to "0" in UART mode.

Table 1.15.13. Features in I^2C mode

	Function	Normal mode	I^2C mode (Note 1)
1	Factor of interrupt number 10 (Note 2, 4)	Bus collision detection	Start condition detection or stop condition detection
2	Factor of interrupt number 15 (Note 2)	UART2 transmission	No acknowledgment detection (NACK)
3	Factor of interrupt number 16 (Note 2)	UART2 reception	Acknowledgment detection (ACK)
4	UART2 transmission output delay	Not delayed	Delayed
5	P_{70} at the time when UART2 is in use	TxD2 (output)	SDA (input/output) (Note 3)
6	P_{71} at the time when UART2 is in use	RxD2 (input)	SCL (input/output)
7	P_{72} at the time when UART2 is in use	CLK2	P_{72}
8	DMA1 factor at the time when 1101 is assigned to the DMA request factor selection bits	UART2 reception	Acknowledgment detection (ACK)
9	Noise filter width	15ns	200ns
10	Reading P_{71}	Reading the terminal when 0 is assigned to the direction register	Reading the terminal regardless of the value of the direction register
11	Initial value of UART2 output	H level (when 0 is assigned to the CLK polarity select bit)	The value set in latch P_{70} when the port is selected (Note 3)

Note 1: Make the settings given below when I^2C mode is in use.
Set 0 1 0 in bits 2, 1, 0 of the UART2 transmission/reception mode register.
Disable the RTS/CTS function. Choose the MSB First function.

Note 2: Follow the steps given below to switch from a factor to another.
1. Disable the interrupt of the corresponding number.
2. Switch from a factor to another.
3. Reset the interrupt request flag of the corresponding number.
4. Set an interrupt level of the corresponding number.

Note 3: Set an initial value of SDA transmission output when serial I/O is invalid.

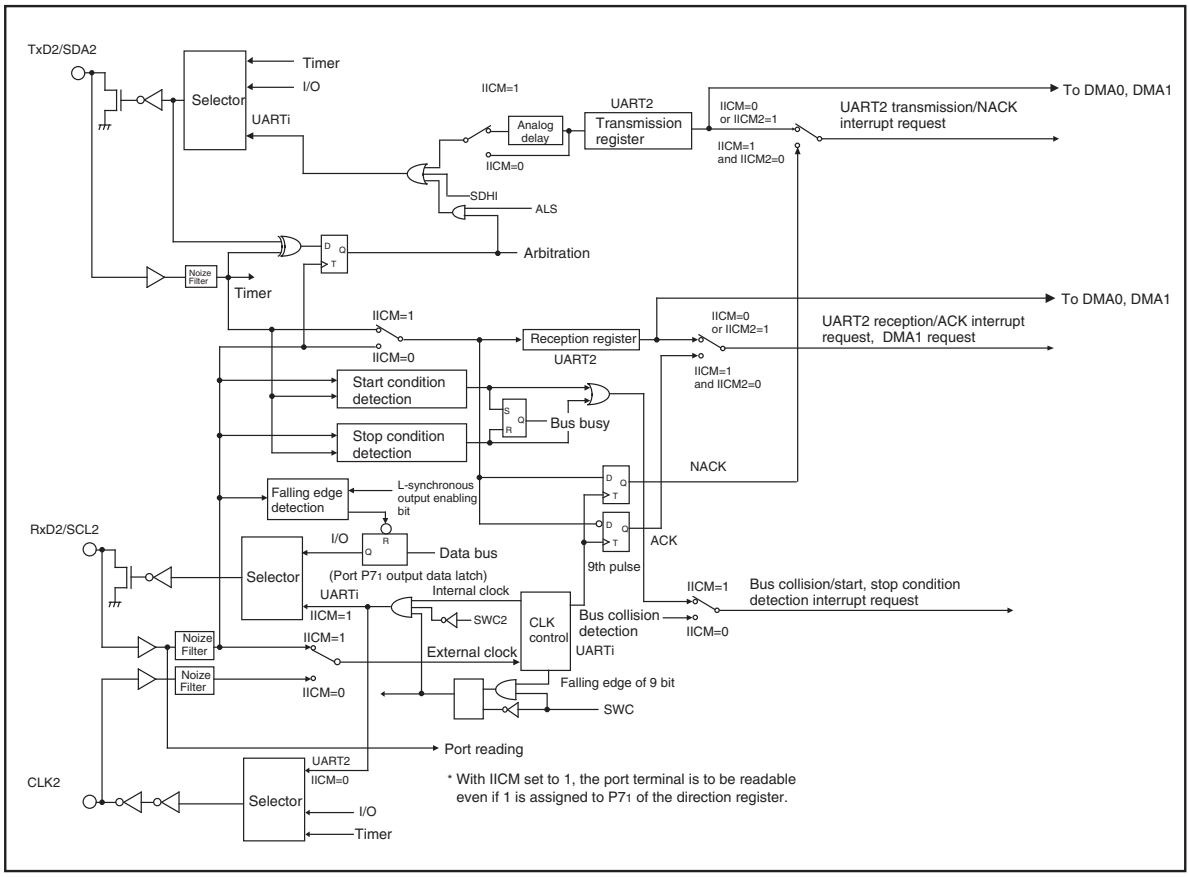


Figure 1.15.28. Functional block diagram for I²C mode

Figure 1.15.28 shows the functional block diagram for I²C mode.

Setting “1” in the I2C mode select bit (IICM) causes ports to work as data transmission-reception terminal SDAi, clock input-output terminal SCLi, and port respectively. A delay circuit is added to the SDA2 transmission output, so the SDA2 output changes after SCL2 fully goes to “L”.

An attempt to read Port (SCL2) results in getting the terminal’s level regardless of the content of the port direction register. The initial value of SDA2 transmission output in this mode goes to the value set in port.

The interrupt factors of the bus collision detection interrupt, UART2 transmission interrupt, and of UART2 reception interrupt turn to the start/stop condition detection interrupt, acknowledgment non-detection interrupt, and acknowledgment detection interrupt respectively.

The start condition detection interrupt refers to the interrupt that occurs when the falling edge of the SDA2 terminal is detected with the SCL2 terminal staying “H”. The stop condition detection interrupt refers to the interrupt that occurs when the rising edge of the SDA2 terminal is detected with the SCL2 terminal staying “H”. The bus busy flag (bit 2 of the UART2 special mode register) is set to “1” by the start condition detection, and set to “0” by the stop condition detection.

The acknowledgment non-detection interrupt refers to the interrupt that occurs when the SDA2 terminal level is detected still staying “H” at the rising edge of the 9th transmission clock. The acknowledgment detection interrupt refers to the interrupt that occurs when SDA2 terminal’s level is detected already went to “L” at the 9th transmission clock. Also, assigning (UART2 reception) to the DMA1 request factor select bits provides the means to start up the DMA transfer by the effect of acknowledgment detection.

Bit 1 of the UART2 special mode register is used as the arbitration lost detecting flag control bit. Arbitration means the act of detecting the nonconformity between transmission data and SDA2 terminal data at the timing of the SCL2 rising edge. This detecting flag is located at bit 11 of the UART2 reception buffer register, and “1” is set in this flag when nonconformity is detected. Use the arbitration lost detecting flag control bit to choose which way to use to update the flag, bit by bit or byte by byte. When setting this bit to “1” and updated the flag byte by byte if nonconformity is detected, the arbitration lost detecting flag is set to “1” at the falling edge of the 9th transmission clock.

If update the flag byte by byte, must judge and clear (“0”) the arbitration lost detecting flag after completing the first byte acknowledge detect and before starting the next one byte transmission.

Bit 3 of the UART2 special mode register is used as SCL2- and L-synchronous output enable bit. Setting this bit to “1” goes the port data register to “0” in synchronization with the SCL2 terminal level going to “L”.

Some other functions added are explained here. Figure 1.15.29 shows their workings.
Bit 4 of the UART2 special mode register is used as the bus collision detect sampling clock select bit. The bus collision detect interrupt occurs when the RxD2 level and TxD2 level do not match, but the nonconformity is detected in synchronization with the rising edge of the transfer clock signal if the bit is set to “0”. If this bit is set to “1”, the nonconformity is detected at the timing of the overflow of timer Aj rather than at the rising edge of the transfer clock.
Bit 5 of the UART2 special mode register is used as the auto clear function select bit of transmit enable bit. Setting this bit to “1” automatically resets the transmit enable bit to “0” when “1” is set in the bus collision detect interrupt request bit (nonconformity).
Bit 6 of the UART2 special mode register is used as the transmit start condition select bit. Setting this bit to “1” starts the TxD transmission in synchronization with the falling edge of the RxD terminal.

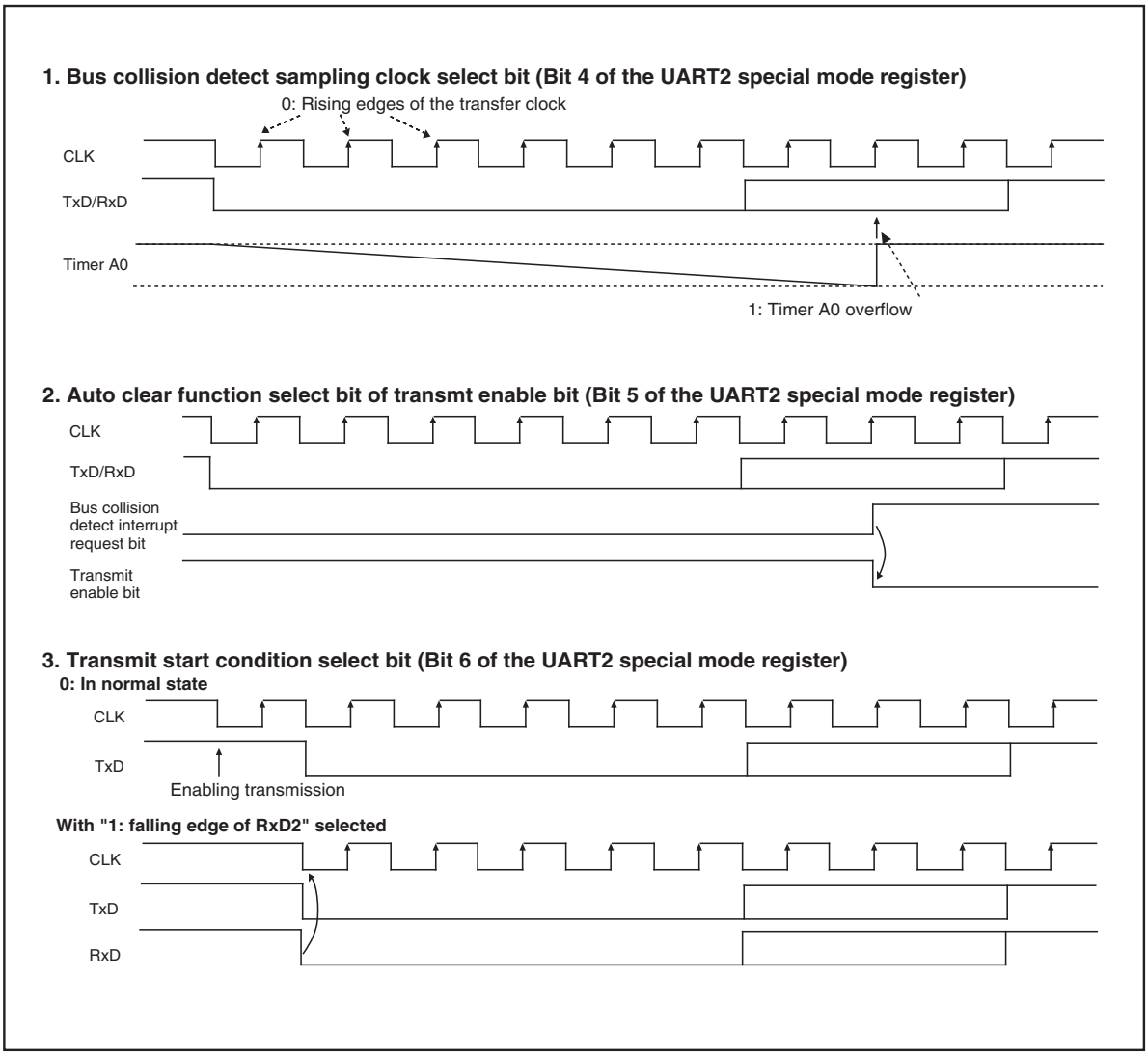


Figure 1.15.29. Some other functions added

UART2 Special Mode Register 2

UARTi special mode register 2 (address 037_{16}) is used to further control UART2 in I^2C mode.

Bit 0 of the UART2 special mode register 2 (address 037_{16}) is used as the I^2C mode select bit 2. Table 1.15.14 shows the types of control to be changed by I^2C mode select bit 2 when the I^2C mode select bit is set to "1". Table 1.15.15 shows the timing characteristics of detecting the start condition and the stop condition.

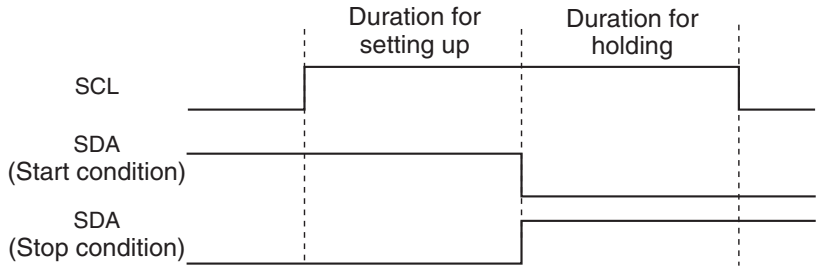
Table 1.15.14. Functions changed by I^2C mode select bit 2

	Function	IICM2 = 0	IICM2 = 1
1	Factor of interrupt number 15	No acknowledgment detection (NACK)	UART2 transmission (the rising edge of the final bit of the clock)
2	Factor of interrupt number 16	Acknowledgment detection (ACK)	UART2 reception (the falling edge of the final bit of the clock)
3	DMA1 factor at the time when 1101 is assigned to the DMA request factor selection bits.	Acknowledgment detection (ACK)	UART2 reception (the falling edge of the final bit of the clock)
4	Timing for transferring data from the UART2 reception shift register to the reception buffer.	The rising edge of the final bit of the reception clock	The falling edge of the final bit of the reception clock
5	Timing for generating a UART2 reception/ACK interrupt request	The rising edge of the final bit of the reception clock	The falling edge of the final bit of the reception clock

Table 1.15.15. Timing characteristics of detecting the start condition and the stop condition (Note 1)

3 to 6 cycles < duration for setting-up (Note)
3 to 6 cycles < duration for holding (Note)

Note: Cycles is in terms of the input oscillation frequency f(X_{IN}) of the main clock.



Bit 3 of the UARTi special mode register 2 are used as the SDAi output stop bit. Setting this bit to “1” causes an arbitration loss to occur, and the SDAi pin turns to high-impedance state at the instant when the arbitration lost detecting flag is set to “1”.

Bit 1 of the UART2 special mode register 2 are used as the clock synchronization bit. With this bit set to “1” at the time when the internal SCL2 is set to “H”, the internal SCL2 turns to “L” if the falling edge is found in the SCL2 pin; and the baud rate generator reloads the set value, and start counting within the “L” interval.
When the internal SCL2 changes from “L” to “H” with the SCL2 pin set to “L”, stops counting the baud rate generator, and starts counting it again when the SCL2 pin turns to “H”. Due to this function, the UART2 transmission-reception clock becomes the logical product of the signal flowing through the internal SCL2 and that flowing through the SCL2 pin. This function operates over the period from the moment earlier by a half cycle than falling edge of the UART2 first clock to the rising edge of the ninth bit. To use this function, choose the internal clock for the transfer clock.

Bit 2 of the UART2 special mode register 2 are used as the SCL2 wait output bit. Setting this bit to “1” causes the SCL2 pin to be fixed to “L” at the falling edge of the ninth bit of the clock. Setting this bit to “0” frees the output fixed to “L”.

Bit 4 of the UART2 special mode register 2 are used as the UART2 initialization bit. Setting this bit to “1”, and when the start condition is detected, the microcomputer operates as follows.
(1) The transmission shift register is initialized, and the content of the transmission register is transferred to the transmission shift register. This starts transmission by dealing with the clock entered next as the first bit. The UART2 output value, however, doesn’t change until the first bit data is output after the entrance of the clock, and remains unchanged from the value at the moment when the microcomputer detected the start condition.
(2) The reception shift register is initialized, and the microcomputer starts reception by dealing with the clock entered next as the first bit.
(3) The SCL2 wait output bit turns to “1”. This turns the SCL2 pin to “L” at the falling edge of the ninth bit of the clock.

Starting to transmit/receive signals to/from UART2 using this function doesn’t change the value of the transmission buffer empty flag. To use this function, choose the external clock for the transfer clock.
Bit 5 of the UART2 special mode register 2 are used as the SCL2 pin wait output bit 2. Setting this bit to “1” with the serial I/O specified allows the user to forcibly output an “1” from the SCL2 pin even if UART2 is in operation. Setting this bit to “0” frees the “L” output from the SCL2 pin, and the UART2 clock is input/output.

Bit 6 of the UART2 special mode register 2 are used as the SDA2 output disable bit. Setting this bit to “1” forces the SDA2 pin to turn to the high-impedance state. Refrain from changing the value of this bit at the rising edge of the UART2 transfer clock. There can be instances in which arbitration lost detecting flag is turned on.

UART2 Special Mode Register 3

Bit 1 of UART2 special mode register 3 (address 037516) are used to clock phase set bit. Figure1.15.9 shows UART2 special mode register 3.

When both the IIC mode select bit (bit 0 of UART2 special mode select register) and the IIC mode select bit 2 (bit 0 of U2SMR2 register) are “1”, functions changed by these bits are shown in table 1.15.16 and figure 1.15.30.

Bits 5 to 7 of UART2 special mode register 3 are SDA digital delay setting bits. By setting these bits, it is possible to turn the SDA delay OFF or set the BRG count source delay to 2 to 8 cycles.

Table 1.15.16. Functions changed by clock phase set bits

Function	CKPH = 0, IICM = 1, IICM2 = 1	CKPH = 1, IICM = 1, IICM2 = 1
SCL initial and last value	Initial value = H, last value = L	Initial value = L, last value = L
Transfer interrupt factor	Rising edge of 9th bit	Falling edge of 10th bit
Data transfer times from UART receive shift register to receive buffer register	Falling edge of 9th bit	Two times :falling edge of 9th bit and rising edge of 9th bit

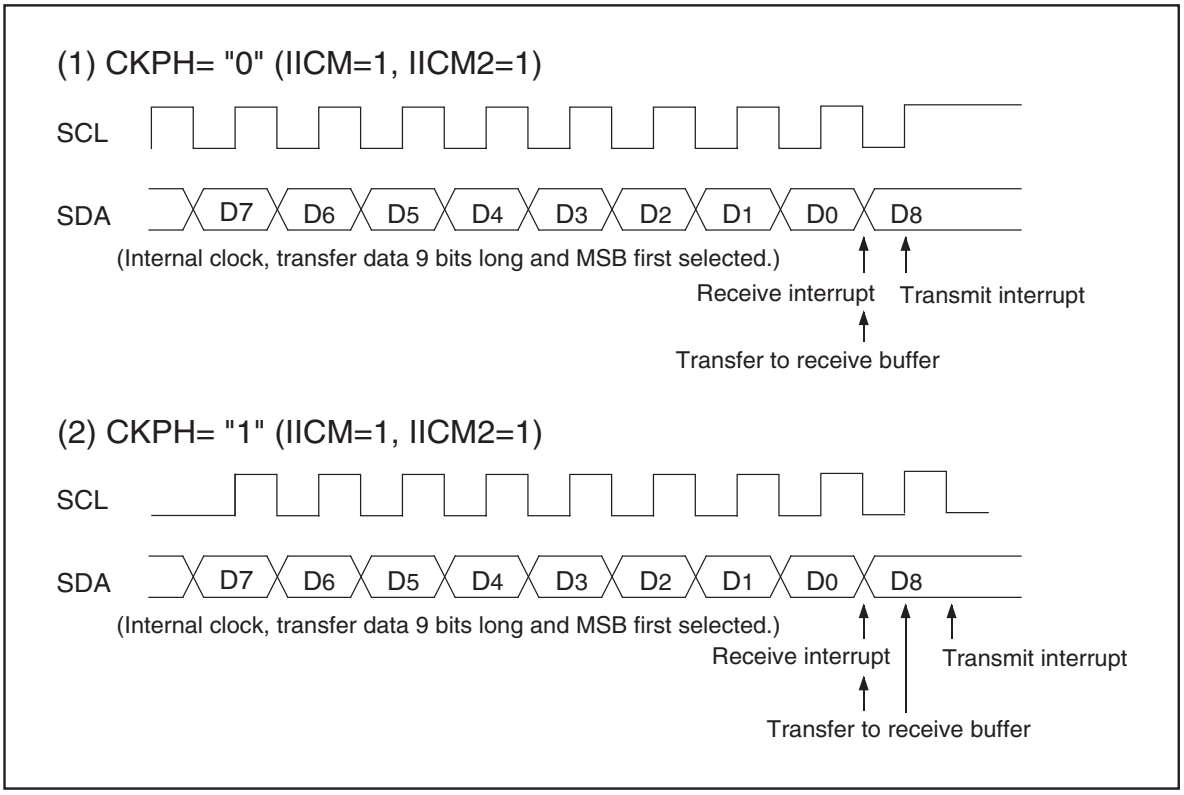


Figure 1.15.30. Functions changed by clock phase set bits

UART2 Special Mode Register 4

Bit 0 of UART2 special mode register 4 (address 0374$_{16}$) are used to start condition generate bit. When the SCL, SDA output select bit (bit 3 of U2SMR4 register) is “1” and this bit is “1”, then the start condition is generated.

Bit 1 of UART2 special mode register 4 are used to restart condition generate bit.
When the SCL, SDA output select bit (bit 3 of U2SMR4 register) is “1” and this bit is “1“, then the restart condition is generated.

Bit 2 of UART2 special mode register 4 are used to stop condition generate bit.
When the SCL, SDA output select bit (bit 3 of U2SMR4 register) is “1” and this bit is “1”, then the stop condition is generated.

Bit 3 of UART2 special mode register 4 are used to SCL, SDA output select bit.
Functions changed by these bits are shown in table 1.15.17 and figure 1.15.31.

Bit 4 of UART2 special mode register 4 are used to ACK data bit.
When the SCL, SDA output select bit (bit 3 of U2SMR4 register) is “0” and the ACK data output enable bit (bit 5 of U2SMR4 register) is “1”, then the content of ACK data bit is output to SDA pin.

Bit 5 of UART2 special mode register 4 are used to ACK data output enable bit.
When the SCL, SDA output select bit (bit 3 of U2SMR4 register) is “0” and this bit is “1”, then the content of ACK data bit is output to SDA pin.

Bit 6 of UART2 special mode register 4 are used to SCL output stop bit.
When this bit is “1”, SCL output is stopped at stop condition detection. (Hi-impedance status).

Bit 7 of UART2 special mode register 4 are used to SCL wait output bit.
When this bit is “1”, SCL output is fixed to "L" at falling edge of 10th bit of clock. When this bit is “0”, SCL output fixed to “L” is released.

Table 1.15.17. Functions changed by SCL, SDA output select bit

Function	STSPSEL = 0	STSPSEL = 1
SCL, SDA output	Output of SI/O control circuit	Output of start/stop conditioncontrol circuit
Start/stop condition interrupt factor	Start/stop condition detection	Completion of start/stop condition generation

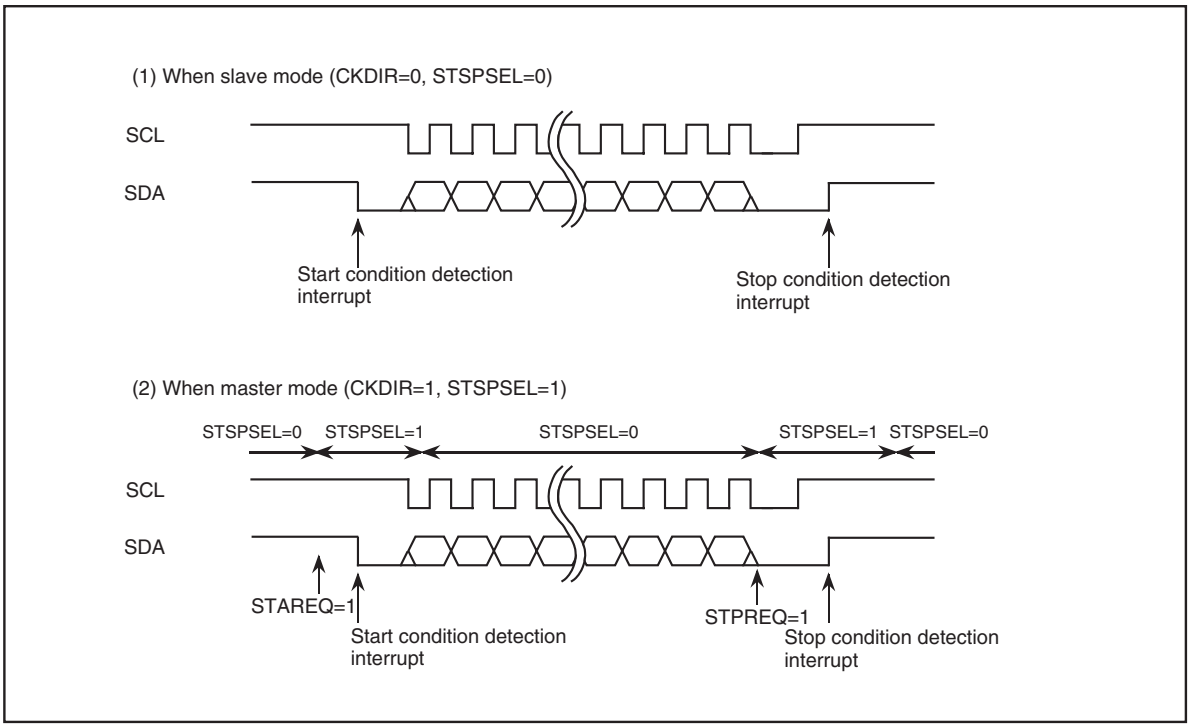


Figure 1.15.31 Functions changed by SCL, SDA output select bit

Serial Interface Special Function

UARTi can control communications on the serial bus (Figure 1.15.32). The master outputting the transfer clock transfers data to the slave inputting the transfer clock. In this case, in order to prevent a data collision on the bus, the master floats the output pin of other slaves/masters using the SSi input pins.

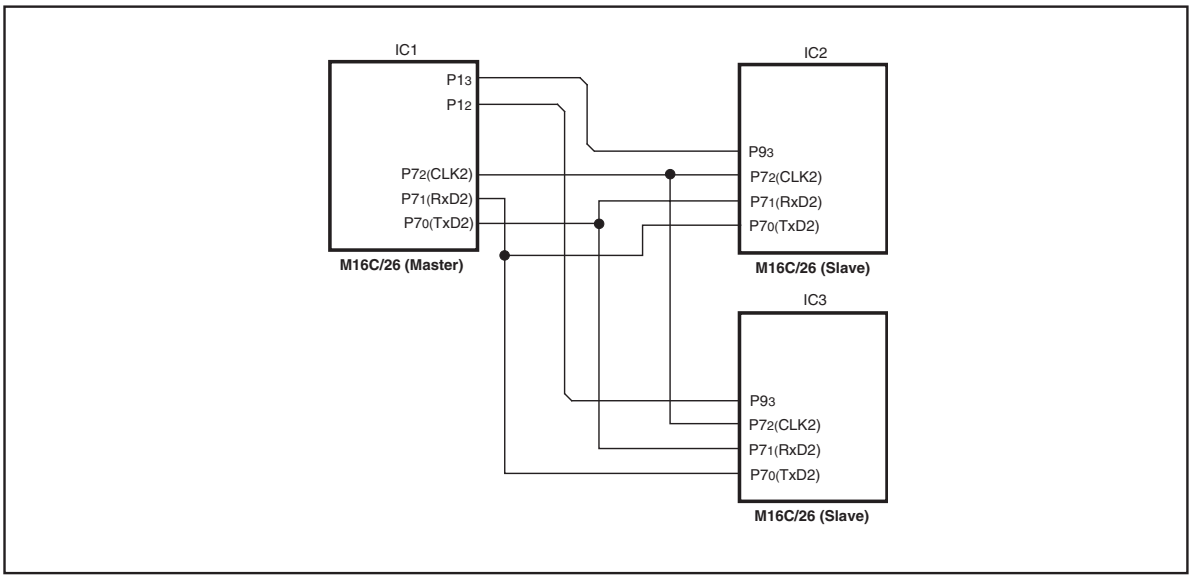


Figure 1.15.32. Programmable serial bus communication control example

• **Clock Phase Setting**

With bit 1 of UART2 special mode register 3 (addresses 036D16, 037116 and 037516) and bit 6 of UART2 transmission-reception control register 0 (addresses 03A416, 02AC16 and 037C16), four combinations of transfer clock phase and polarity can be selected.

Bit 6 of UART2 transmission-reception control register 0 sets transfer clock polarity, whereas bit 1 of U2SMR3 register sets transfer clock phase.

Transfer clock phase and polarity must be the same between the master and slave involved in the transfer.

(a) Master (Internal Clock)

Figure 1.15.33 shows the transmission and reception timing.

(b) Slave (External Clock)

- Figure 1.15.34 shows the timing when bit 1 of address 037516)=“0”
- Figure 1.15.35 shows the timing when bit 1 of address037516)=“1”

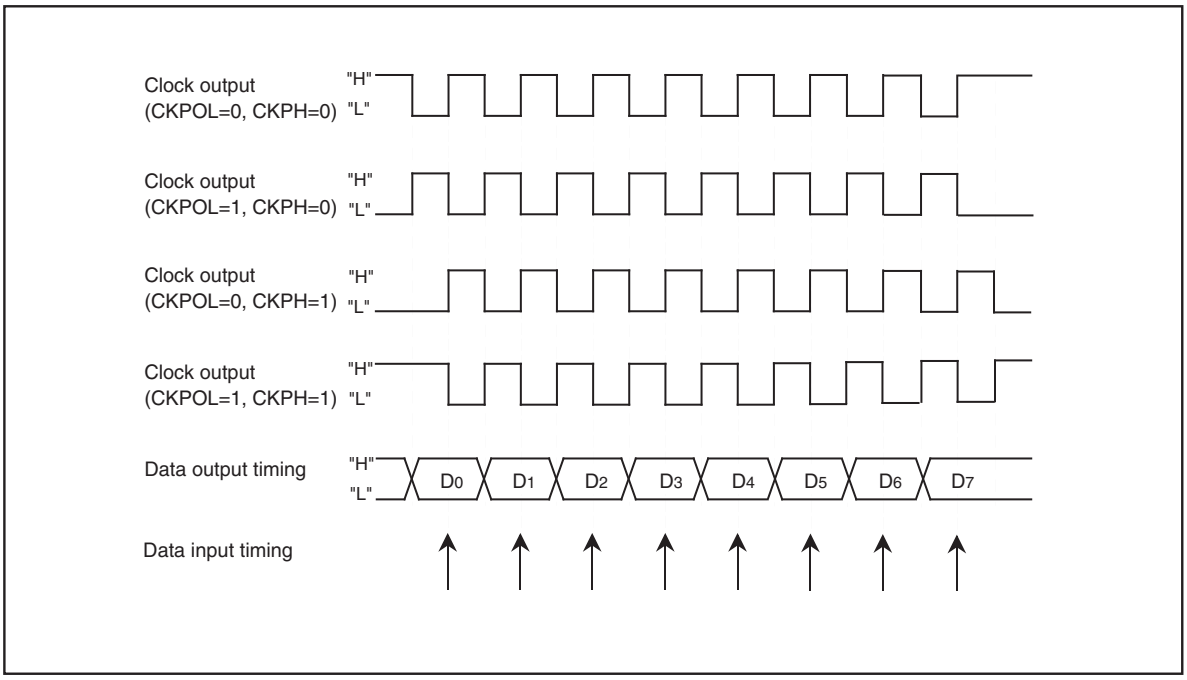


Figure 1.15.33. The transmission and reception timing in master mode (internal clock)

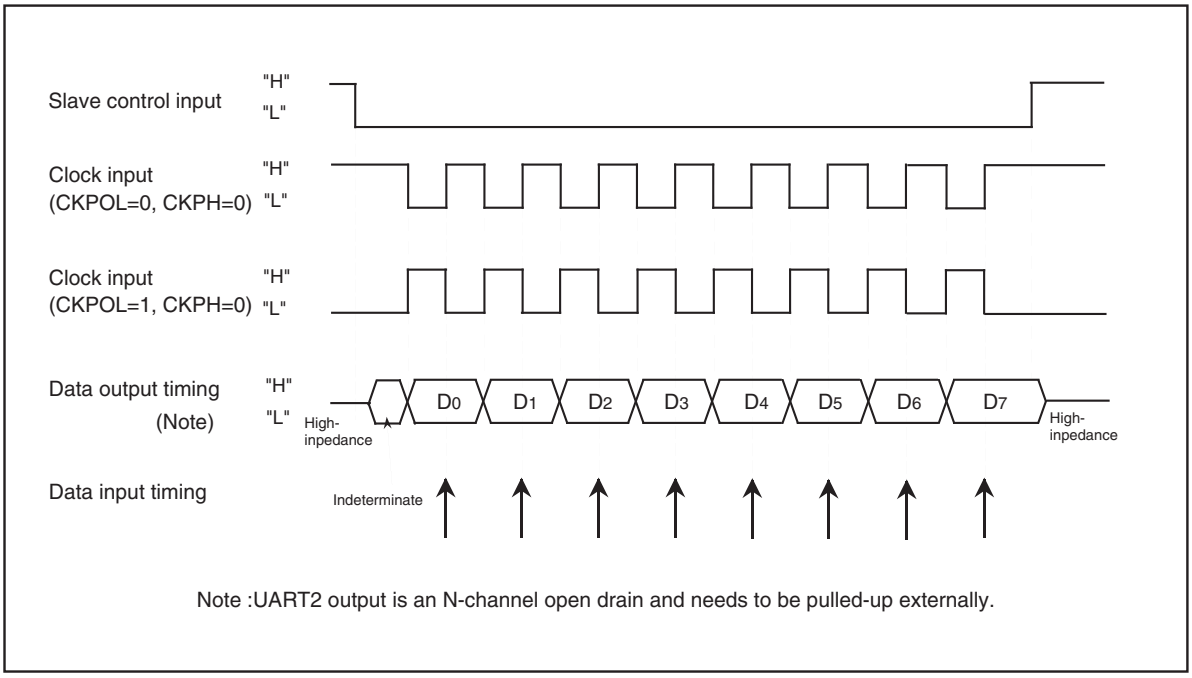


Figure 1.15.34. The transmission and reception timing (CKPH=0) in slave mode (external clock)

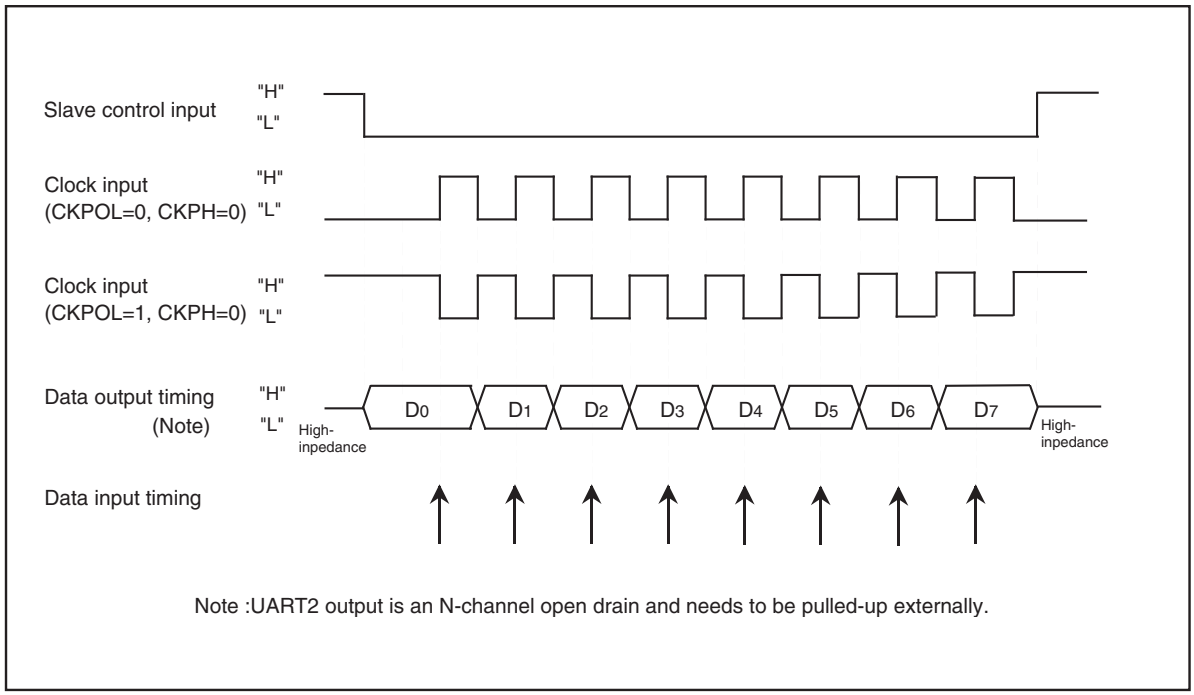


Figure 1.15.35. The transmission and reception timing (CKPH=1) in slave mode (external clock)

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P10_0$ to $P10_7$ also function as the analog signal input pins AN_0 to AN_7. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address $03D7_{16}$) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (V_{REF}) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of $03D7_{16}$ to connect V_{REF}.

The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses. Table 1.16.1 shows the performance of the A-D converter. Figure 1.16.1 shows the block diagram of the A-D converter, and Figures 1.16.2 and 1.16.3 show the A-D converter-related registers.

Table 1.16.1. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AV_{CC} (V_{CC})
Operating clock f_{AD} (Note 2) Resolution	f_{AD}, $f_{AD}/2$, $f_{AD}/3$, $f_{AD}/4$, $f_{AD}/6$, or $f_{AD}/12$ where $f_{AD}=f(X_{IN})$ 8-bit or 10-bit (selectable)
Integral nonlinearity error	When $AV_{CC} = V_{REF} = 5V$ • With 8-bit resolution: ±2LSB • With 10-bit resolution: ±3LSB When $AV_{CC} = V_{REF} = 3.3V$ • With 8-bit resolution: ±2LSB • With 10-bit resolution: ±5LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins A-D conversion start condition	8pins (AN_0 to AN_7) • Software trigger A-D conversion starts when the A-D conversion start flag changes to “1” • External trigger (can be retriggered) A-D conversion starts when the A-D conversion start flag is “1” and the $ADTRG/P1_5$ input (shared with INT3) changes from “H” to “L”
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 f_{AD} cycles, 10-bit resolution: 59 f_{AD} cycles • With sample and hold function 8-bit resolution: 28 f_{AD} cycles, 10-bit resolution: 33 f_{AD} cycles

Note 1: Does not depend on use of sample and hold function.
Note 2: Divide the f_{AD} if $f(X_{IN})$ exceeds 10MHz, and make ϕ_{AD} frequency equal to or less than 10MHz.
Without sample and hold function, set the ϕ_{AD} frequency to 250kHz min.
With the sample and hold function, set the ϕ_{AD} frequency to 1MHz min.

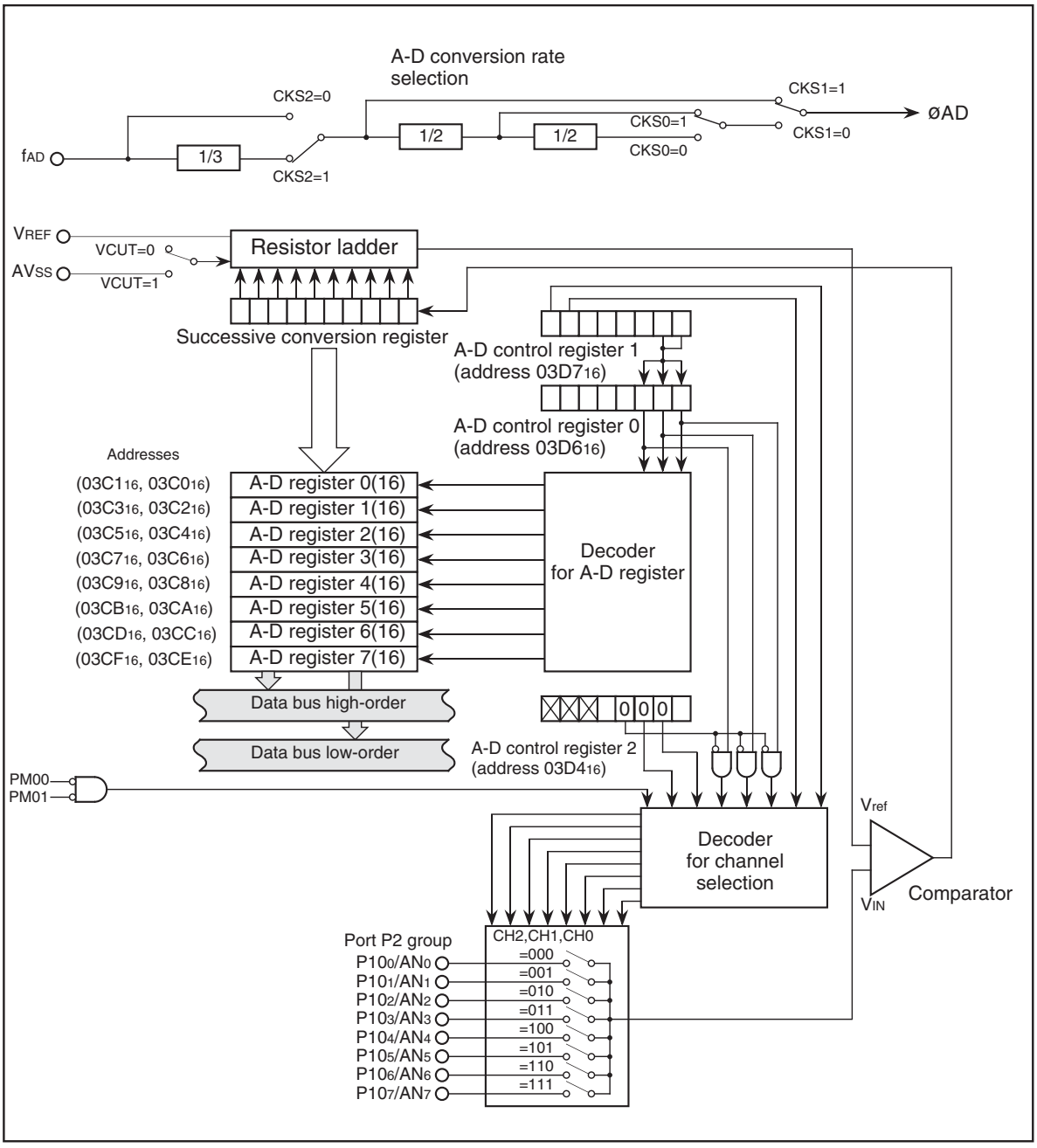
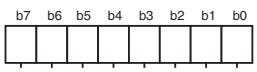


Figure 1.16.1. Block diagram of A-D converter

A-D control register 0 (Note 1)

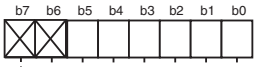


Symbol	Address	When reset
ADCON0	03D6₁₆	00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 (Note 2) Repeat sweep mode 1	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit	Refer to table 1.16.2	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note 1)



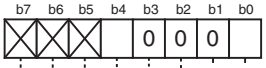
Symbol	Address	When reset
ADCON1	03D7₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins)	○	○
SCAN1		When repeat sweep mode 1 is selected b1 b0 0 0 : AN0 (1 pin) 0 1 : AN0, AN1 (2 pins) 1 0 : AN0 to AN2 (3 pins 1 1 : AN0 to AN3 (4 pins)	○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit	Refer to table 1.16.2	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure 1.16.2. A-D converter control registers (1)

A-D control register 2 (Note 1)



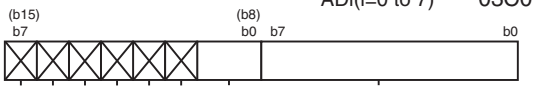
Symbol: ADCON2 | Address: 03D4₁₆ | When reset: 00₁₆

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
Reserved bits		Must always be set to 0	○	○
CKS2	Frequency select bit 2	Refer to table 1.16.2	○	○
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .			–	–

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register i

Symbol: ADi(i=0 to 7) | Address: 03C0₁₆ to 03CF₁₆ | When reset: Indeterminate



Function	R	W
Eight low-order bits of A-D conversion result	○	×
During 10-bit mode Two high-order bits of A-D conversion result	○	×
During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. In an attempt to write to these bits, write 0 . The value, if read, turns out to be 0 .	–	–

Figure 1.16.3. A-D converter control registers (2)

Table 1.16.2. Operation clock ϕ_{AD}

CKS0	CKS1	CKS2	ϕ_{AD}
0	0	0	$f_{AD}/4$
0	0	1	$f_{AD}/12$
0	1	0	f_{AD}
0	1	1	$f_{AD}/3$
1	0	0	$f_{AD}/2$
1	0	1	$f_{AD}/6$
1	1	0	f_{AD}
1	1	1	$f_{AD}/3$

Note: Divide the f_{AD} if $f(X_{IN})$ exceeds 10MHz, and make ϕ_{AD} frequency equal to or less than 10MHz.

(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bits , is used for one-shot A-D conversion. Table 1.16.3 shows the specifications of one-shot mode. Figure 1.16.4 shows the settings of the A-D control registers in one-shot mode.

Table 1.16.3. One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bits is used for one A-D conversion
Start condition	Writing "1" to the A-D conversion start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to "0" - except when external trigger is selected • Writing "0" to the A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	Select one from AN_0 to AN_7
Reading of A-D result	Read A-D register that corresponds to selected analog input pin

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0			

Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode (Note 2)	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{ADTRG}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	Refer to table 1.16.2	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pins again.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕	1			0		

Symbol ADCON1 — Address $03D7_{16}$ — When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in one-shot mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set to 0 when this mode is selected	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	Refer to table 1.16.2	○	○
VCUT	Vref connect bit (Note 2)	1 : Vref connected	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.16.4. A-D conversion control registers in one-shot mode

(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bits , is used for one-shot A-D conversion. Table 1.16.4 shows the specifications of repeat mode. Figure 1.16.5 shows the settings of the A-D control registers in repeat-shot mode.

Table 1.16.4. Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bits is used for one A-D conversion
Start condition	Writing "1" to the A-D conversion start flag
Stop condition	Writing "0" to the A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	Select one from AN_0 to AN_7
Reading of A-D result	Read A-D register that corresponds to selected analog input pin (at any time)

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	1			

Symbol ADCON0 — Address 03D6₁₆ — When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode (Note 2)	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{ADTRG}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	Refer to table 1.16.2	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pins again.

A-D control register 1 (Note 1)

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

Symbol ADCON1 — Address 03D7₁₆ — When reset 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in repeat mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set to 0 when this mode is selected	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	Refer to table 1.16.2	○	○
VCUT	Vref connect bit (Note 2)	1 : Vref connected	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.16.5. A-D conversion control registers in repeat mode

(3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bits, are used for one-by-one A-D conversion. Table 1.16.5 shows the specifications of single sweep mode. Figure 1.16.6 shows the settings of the A-D control registers in single sweep mode.

Table 1.16.5. Single sweep mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bits is used for one A-D conversion
Start condition	Writing "1" to the A-D conversion start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to "0" - except when external trigger is selected • Writing "0" to the A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN_0 - AN_1 (2 pins), AN_0 - AN_3 (4 pins), AN_0 - AN_5 (6 pins), or AN_0 - AN_7 (8 pins)
Reading of A-D result	Read A-D register that corresponds to selected analog input pin

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	0			

Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in single sweep mode	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	Refer to table 1.16.2	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕	1			0		

Symbol ADCON1 — Address $03D7_{16}$ — When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep mode is selected b1 b0 0 0 : AN_0, 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)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set to 0 when this mode is selected	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	Refer to table 1.16.2	○	○
VCUT	Vref connect bit (Note 2)	1 : Vref connected	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.16.6. A-D conversion control registers in single sweep mode

(4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bits , are used for repeat sweep A-D conversion. Table 1.16.6 shows the specifications of repeat sweep mode 0. Figure 1.16.7 shows the settings of the A-D control registers in repeat sweep mode 0.

Table 1.16.6. Repeat sweep mode 0 specifications

Item	Specification
Function	The pin selected by the analog input pin select bits is used for one A-D conversion
Start condition	Writing "1" to the A-D conversion start flag
Stop condition	Writing "0" to the A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN_0 - AN_1 (2 pins), AN_0 - AN_3 (4 pins), AN_0 - AN_5 (6 pins), or AN_0 - AN_7 (8 pins)
Reading of A-D result	Read A-D register that corresponds to selected analog input pin (at any time)

A-D control register 0 (Note)

b7 b6 b5 b4 b3 b2 b1 b0: 1 1

Symbol: ADCON0 — Address: $03D6_{16}$ — When reset: $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{ADTRG}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	Refer to table 1.16.2	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

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

Symbol: ADCON1 — Address: $03D7_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When repeat sweep mode 0 is selected b1 b0 0 0 : AN_0, 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)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set to 0 when this mode is selected	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	Refer to table 1.16.2	○	○
VCUT	Vref connect bit (Note 2)	1 : Vref connected	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.16.7. A-D conversion control registers in repeat sweep mode 0

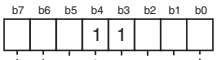
(5) Repeat sweep mode 1

In repeat sweep mode 1, all pins are used for A-D conversion with emphasis on the pin or pins selected using the A-D sweep pin select bits. Table 1.16.7 shows the specifications of repeat sweep mode 1. Figure 1.16.8 shows the settings of the A-D control registers in repeat sweep mode 1.

Table 1.16.7. Repeat sweep mode 1 specifications

Item	Specification
Function	The pin selected by the analog input pin select bits is used for one A-D conversion
Start condition	Writing "1" to the A-D conversion start flag
Stop condition	Writing "0" to the A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	With emphasis on these pins: AN_0, AN_0 - AN_1 (2 pins), AN_0 - AN_2 (3 pins), AN_0 - AN_3 (4 pins)
Reading of A-D result	Read A-D register that corresponds to selected analog input pin (at any time)

A-D control register 0 (Note)

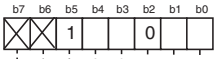


Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 1	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	Refer to table 1.16.2	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)



Symbol ADCON1 — Address $03D7_{16}$ — When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set to 1 when this mode is selected	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	Refer to table 1.16.2	○	○
VCUT	Vref connect bit (Note 2)	1 : Vref connected	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A-D conversion.

Figure 1.16.8. A-D conversion control registers in repeat sweep mode 1

(a) Sample and Hold

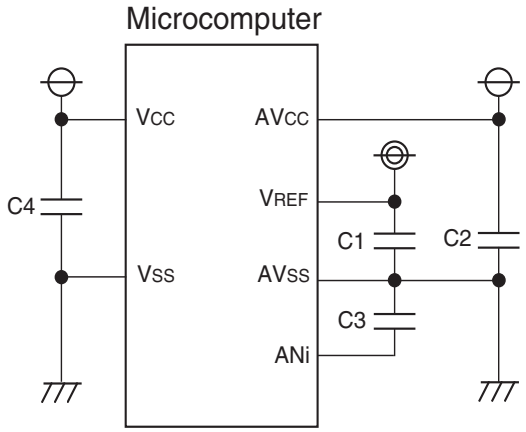
Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D4$_{16}$) to "1". When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 ϕ_{AD} cycle is achieved in 8-bit resolution and 33 ϕ_{AD} in 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify whether to use sample and hold before starting A-D conversion.

(b) Caution of using A-D converter

(1) Set the port direction bits corresponding to the following pins to input: those P10 pins to be used for analog input, plus external trigger input pin (P1$_5$).

(2) In using a key-input interrupt, none of 4 pins (AN$_4$ through AN$_7$) can be used as an A-D conversion port (if the A-D input voltage goes to "L" level, a key-input interrupt occurs).

(3) Insert the capacitor between AV_{CC} and AV_{SS}, between V_{REF} and AVss, and between the analog input pin (ANi) and AV_{SS}, to prevent a malfunction or program runaway, and to reduce conversion error, due to noise. Figure 1.16.9 is an example connection of each pin.



Note 1: C1 ≥ 0.47uF, C2 ≥ 0.47uF, C3 ≥ 100pF, C4 ≥ 0.1uF (reference)
Note 2: Ensure these connections are as thick and short as possible.

Figure 1.16.9. Example connection of V_{CC}, V_{SS}, AV_{CC}, AV_{SS}, V_{REF} and ANi

Programmable I/O Ports

There are 38 programmable I/O ports: $P1_5$ - $P1_7$, P6, P7, P8 (except $P8_4$), $P9_0$ - $P9_3$, and P10. Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set.

Figures 1.17.1 to 1.17.3 show the programmable I/O ports. Figure 1.17.4 shows the I/O pins.
Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.
To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices, they function as outputs regardless of the contents of the direction registers. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

(1) Direction registers

Figure 1.17.5 shows the port direction registers.
These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

Note: The Port 9 direction register incorporates a write protection function. Before writing to the port 9 direction register the write protection must be disabled by setting PRC2 of the Protect register (bit 2 at $000A_{16}$) to "1". Note that PRC2 is automatically reset to "0" after the next write to an SFR address.

(2) Port registers

Figure 1.17.6 shows the port registers.
These registers are used to read and write data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in the port registers corresponds one for one to each I/O pin.

(3) Pull-up control registers

Figure 1.17.7 shows the pull-up control registers.
The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set to input.

(4) Port control register

Figure 1.17.8 shows the port control register.
Bit 0 of the port control register is used to control the read of port P1 as follows:

0 : When port P1 is set as an input port, the port input level is read.
When port P1 is set as an output port , the contents of the port P1 register are read.

1 : The contents of the port P1 register are always read.

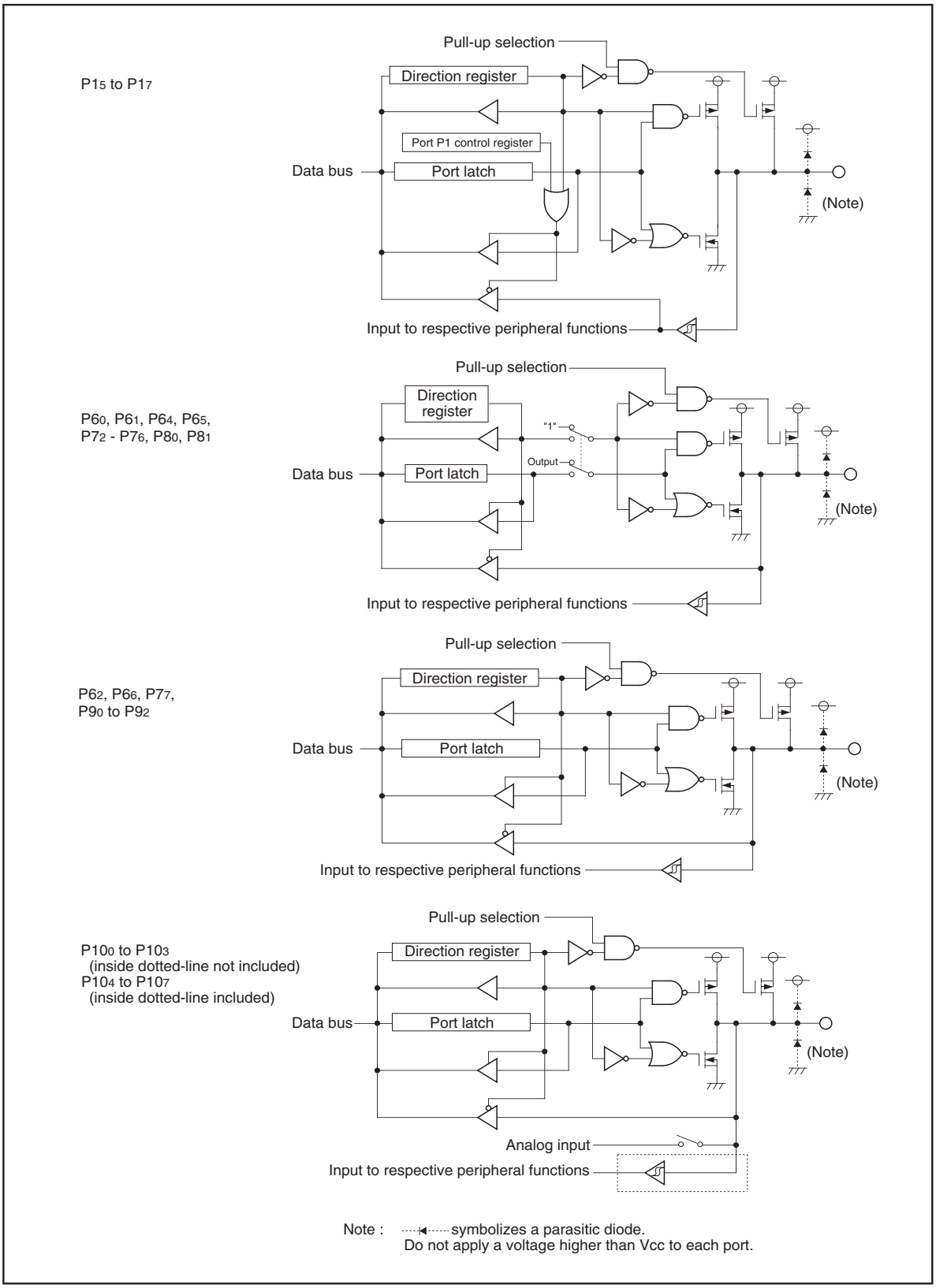


Figure 1.17.1. Programmable I/O ports (1)

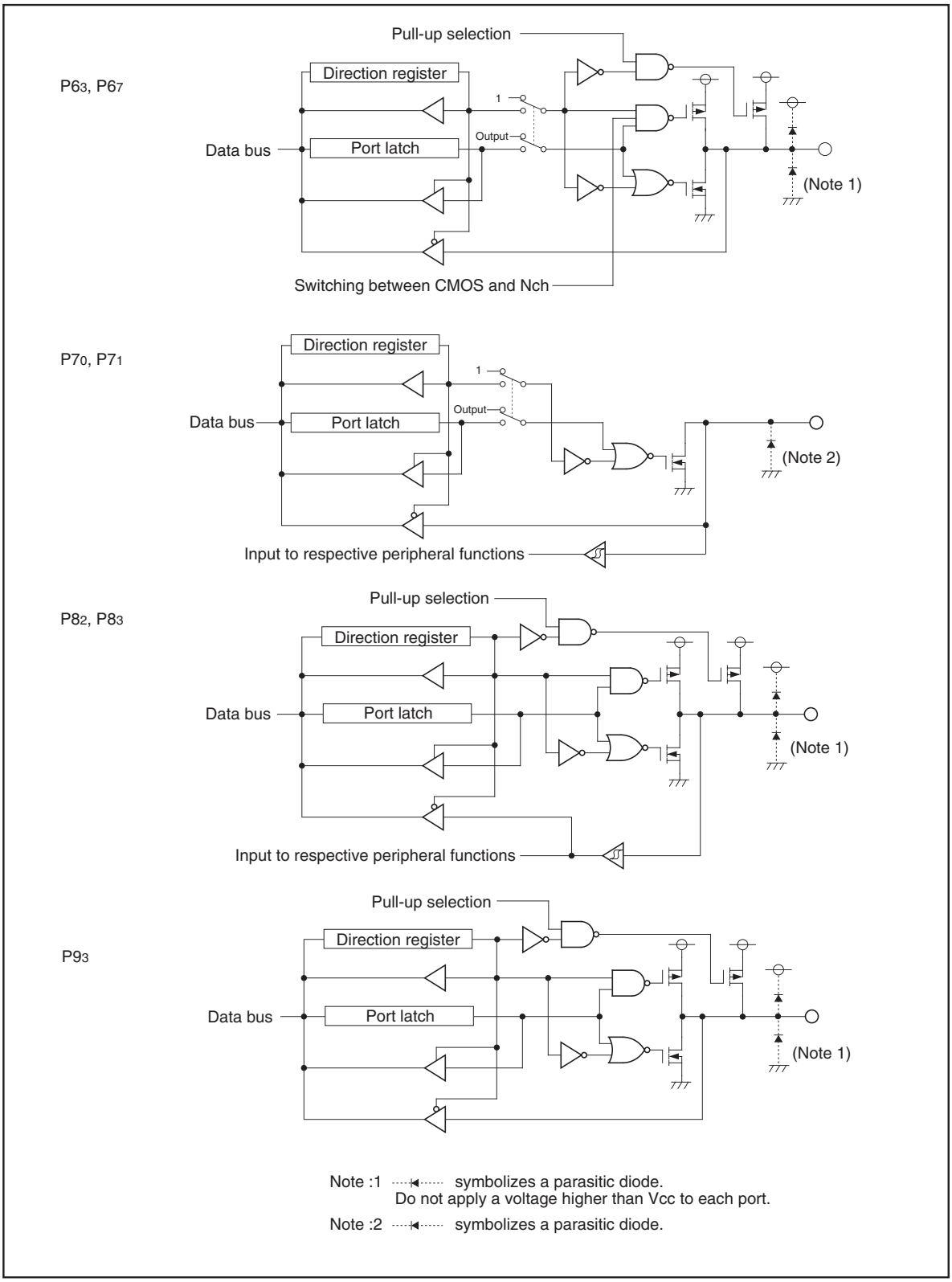


Figure 1.17.2. Programmable I/O ports (2)

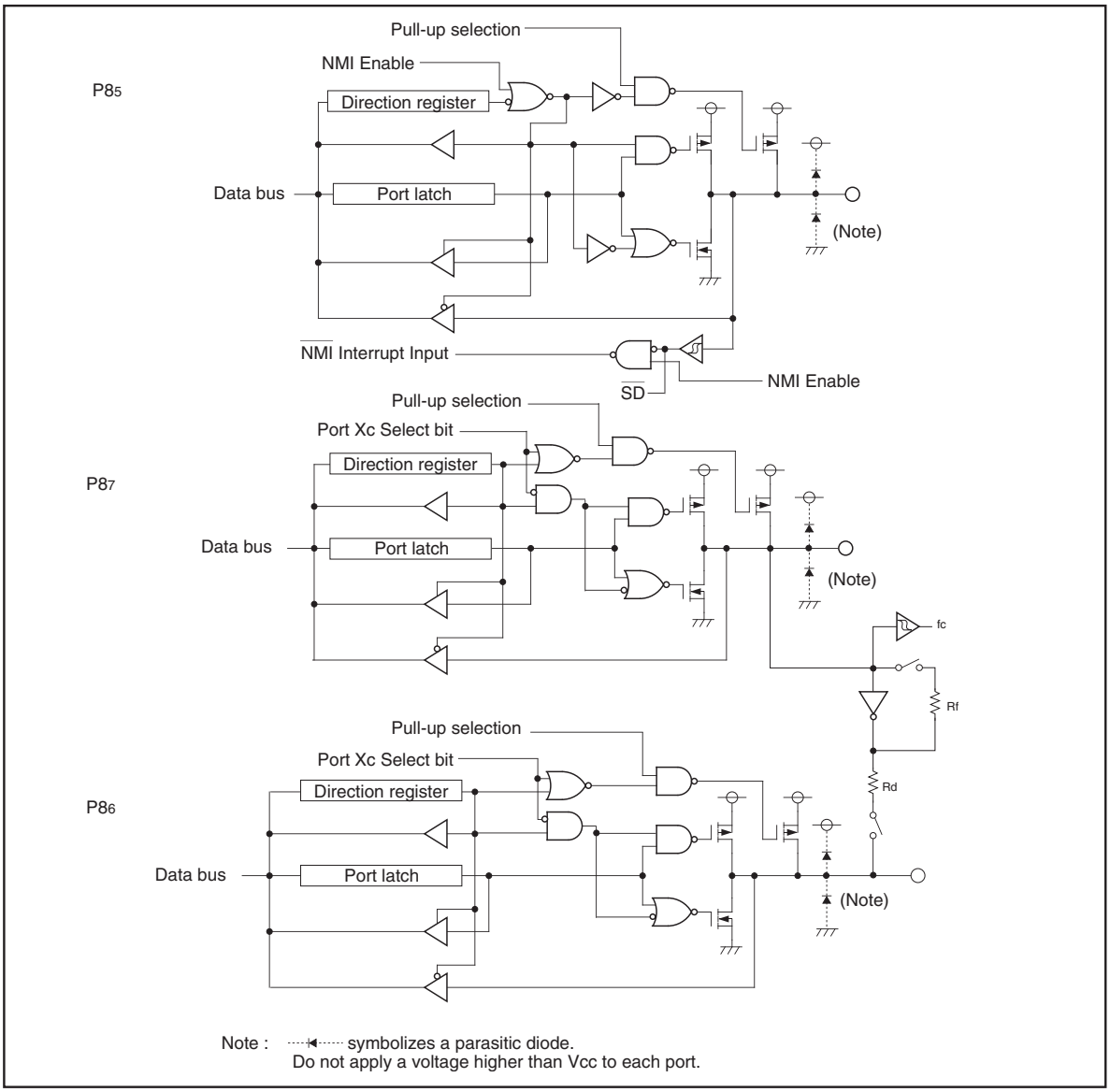


Figure 1.17.3. Programmable I/O ports (3)

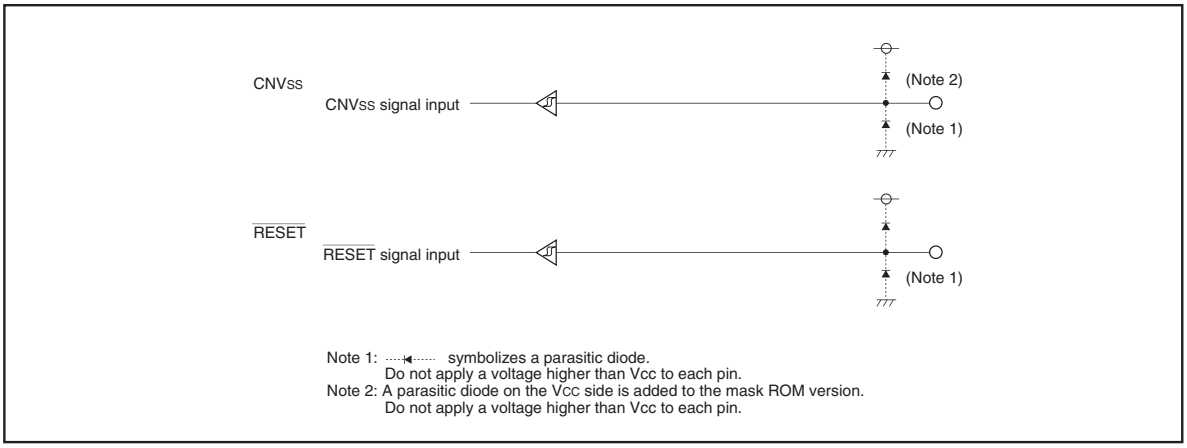
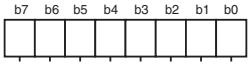


Figure 1.17.4. I/O pins

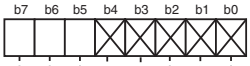
Port Pi direction register



Symbol	Address	When reset
PDi (i = 6, 7, 10)	03EE16, 03EF16, 03F616	0016

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi0 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 6, 7, 10)	O	O
PDi_1	Port Pi1 direction register		O	O
PDi_2	Port Pi2 direction register		O	O
PDi_3	Port Pi3 direction register		O	O
PDi_4	Port Pi4 direction register		O	O
PDi_5	Port Pi5 direction register		O	O
PDi_6	Port Pi6 direction register		O	O
PDi_7	Port Pi7 direction register		O	O

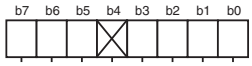
Port P1 direction register



Symbol	Address	When reset
PD1	03E316	000XXXXX2

Bit symbol	Bit name	Function	R	W
Nothing is assigned. Write "0" when writing to this bit. The value is "0" when read.			—	—
PD1_5	Port P15 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
PD1_6	Port P16 direction register		O	O
PD1_7	Port P17 direction register		O	O

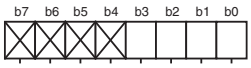
Port P8 direction register



Symbol	Address	When reset
PD8	03F216	000X00002

Bit symbol	Bit name	Function	R	W
PD8_0	Port P80 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
PD8_1	Port P81 direction register		O	O
PD8_2	Port P82 direction register		O	O
PD8_3	Port P83 direction register		O	O
Nothing is assigned. Write "0" when writing to this bit. The value is "0" when read.			—	—
PD8_5	Port P85 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
PD8_6	Port P86 direction register		O	O
PD8_7	Port P87 direction register		O	O

Port P9 direction register (Note)



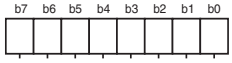
Symbol	Address	When reset
PD9	03F316	XXXX00002

Bit symbol	Bit name	Function	R	W
PD9_1	Port P90 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	O	O
PD9_2	Port P91 direction register		O	O
PD9_3	Port P92 direction register		O	O
PD9_4	Port P93 direction register		O	O
Nothing is assigned. Write "0" when writing to this bit. The value is "0" when read.			—	—

Note : Set bit 2 of the protect register (address 000A16) to "1" when writing new values to this register.

Figure 1.17.5. Direction registers

Port Pi register

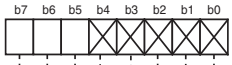


Symbol: Pi (i = 6, 7, 10) | Address: 03EC16, 03ED16, 03F416 | When reset: XX16

Bit symbol	Bit name	Function	R	W
Pi_0	Port Pi0 register	(Note) 0 : "L" level data 1 : "H" level data (i = 6, 7, 10)	O	O
Pi_1	Port Pi1 register		O	O
Pi_2	Port Pi2 register		O	O
Pi_3	Port Pi3 register		O	O
Pi_4	Port Pi4 register		O	O
Pi_5	Port Pi5 register		O	O
Pi_6	Port Pi6 register		O	O
Pi_7	Port Pi7 register		O	O

Note : Since P70 and P71 are N-channel open drain ports, the data is high impedance.

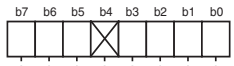
Port P1 register



Symbol: P1 | Address: 03E116 | When reset: XX16

Bit symbol	Bit name	Function	R	W
Nothing is assigned. Write "0" when writing to this bit. The value is "0" when read.			—	—
P1_5	Port P15 register	(Note) 0 : "L" level data 1 : "H" level data	O	O
P1_6	Port P16 register		O	O
P1_7	Port P17 register		O	O

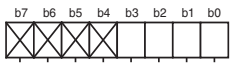
Port P8 register



Symbol: P8 | Address: 03F016 | When reset: XX16

Bit symbol	Bit name	Function	R	W
P8_0	Port P80 register	(Note) 0 : "L" level data 1 : "H" level data	O	O
P8_1	Port P81 register		O	O
P8_2	Port P82 register		O	O
P8_3	Port P83 register		O	O
Nothing is assigned. Write "0" when writing to this bit. The value is "0" when read.			—	—
P8_5	Port P85 register	(Note) 0 : "L" level data 1 : "H" level data	O	O
P8_6	Port P86 register		O	O
P8_7	Port P87 register		O	O

Port P9 register



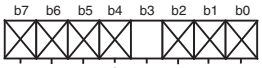
Symbol: P9 | Address: 03F116 | When reset: XX16

Bit symbol	Bit name	Function	R	W
P9_0	Port P90 register	(Note) 0 : "L" level data 1 : "H" level data	O	O
P9_1	Port P91 register		O	O
P9_2	Port P92 register		O	O
P9_3	Port P93 register		O	O
Nothing is assigned. Write "0" when writing to this bit. The value is "0" when read.			—	—

Note: Data is input and output to and from each pin by reading and writing to and from each corresponding bit.

Figure 1.17.6. Port registers

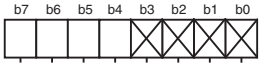
Pull-up control register 0



Symbol: PUR0 | Address: 03FC16 | When reset: 00000000₂

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			—	—
PU03	P15 to P17 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			—	—

Pull-up control register 1

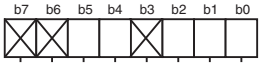


Symbol: PUR1 | Address: 03FD16 | When reset: 00000000₂

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			—	—
PU14	P60 to P63 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU15	P64 to P67 pull-up		○	○
PU16	P72 to P73 pull-up (Note 1)		○	○
PU17	P74 to P77 pull-up		○	○

Note 1: Since P70 and P71 are N-channel open drain ports, pull-up is not available for them.

Pull-up control register 2



Symbol: PUR2 | Address: 03FE16 | When reset: 00000000₂

Bit symbol	Bit name	Function	R	W
PU20	P80 to P83 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU21	P85 to P87 pull-up		○	○
PU22	P90 to P93 pull-up		○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			—	—
PU24	P100 to P103 pull-up	0 : Not pulled high 1 : Pulled high	○	○
PU25	P104 to P107 pull-up		○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			—	—

Figure 1.17.7. Pull-up Control registers

Port control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbpl	Address	When reset
PCR	03FF16	00000000₂

Bit symbol	Bit name	Function	R	W
PCR0	Port P1 control register	0 : When input port, read port input level. When output port, read the contents of port P1 register. 1 : Read the contents of port P1 register when input and output port.	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be 0.			—	—

Figure 1.17.8. Port control register

Table 1.17.1. Example connection of unused pins in single-chip mode

Pin name	Connection
Ports P15-P17, P6, P7, P80-P83, P86-P87, P90-P93, P10	After setting to input mode, connect every pin to Vss via a resistor (pull-down); or after setting to output mode, leave these pins open.
P85(NMI/SD)	After setting to input mode, connect to Vcc via a resistor (pull-up).
XOUT (Note 1)	Open
AVcc	Connect to Vcc
AVSS, VREF	Connect to Vss
VDC	Connect via a 0.1uF capacitor to Vss

Note 1: With external clock input to X_{IN} pin.

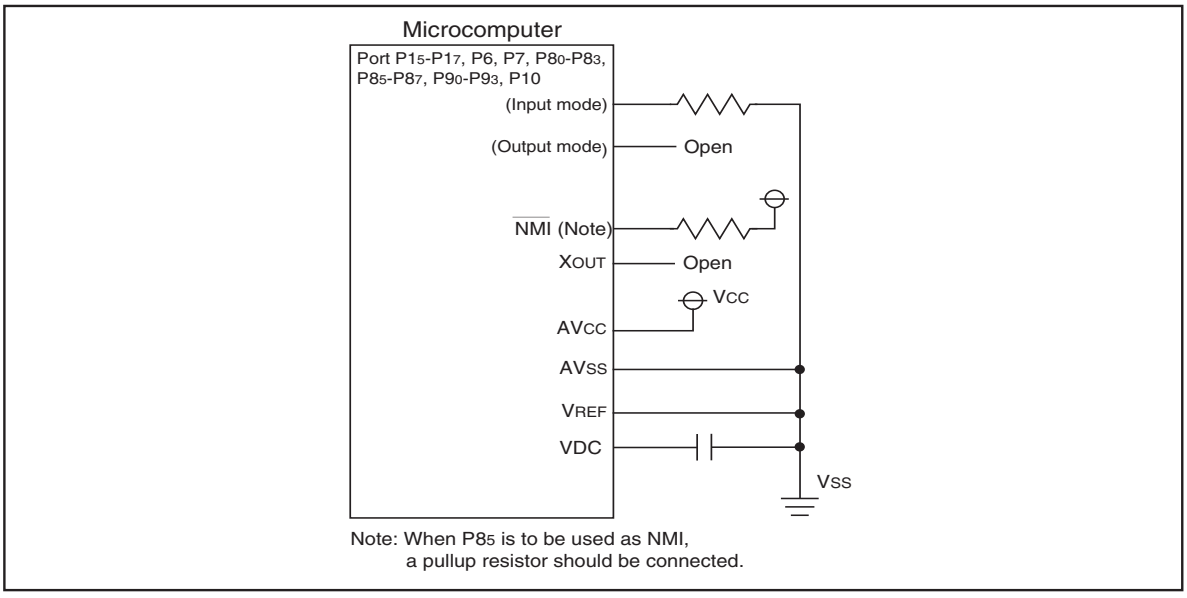


Figure 1.17.9. Example connection of unused pins

Electrical Characteristics

Table 1.18.1. Absolute maximum ratings

Symbol	Parameter		Condition	Rated 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	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$,$P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, V_{REF}, RESET, CNV$_{SS}$		-0.3 to V_{CC}+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.5	V
V_O	Output voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{OUT}		-0.3 to V_{CC}+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.5	V
P_d	Power dissipation		Topr=25°C	300	mW
T_{opr}	Operating ambient temperature			-20 to 85 / -40 to 85 (Note)	°C
T_{stg}	Storage temperature			-65 to 150	°C

Note : Specify a product of -40°C to 85°C to use it.

Tables 1.18.2a. & 1.18.2b. Electrical Characteristics for Flash ROM E/W Cycles

Table 1.18.2a. Characteristics (Note 1) for 100 E/W cycle products (D3, D5, U3, U5)

Symbol	Parameter		Standard value			Unit
			Min.	Typ. (Note 2)	Max.	
–	Erase/Write cycle (Note 3)		100 (Note 4)			cycle
–	Word write time			75	600	µs
–	Block erase time	2Kbyte block		0.2	9	s
		8Kbyte block		0.4	9	s
		16Kbyte block		0.7	9	s
		32Kbyte block		1.2	9	s
td(SR-ES)	Time delay from Suspend Request until Erase Suspend				20	ms
–	Data retention time (Note 5)		10			year

Table 1.18.2b. Characteristics (Note 6) for 10000 E/W cycle products (D7, D9, U7, U9) [Block A and Block B (Note 7)]

Symbol	Parameter	Standard value			Unit
		Min.	Typ. (Note 2)	Max.	
–	Erase/Write cycle (Notes 3, 8, 9)	10000 (Notes 4, 10)			cycle
–	Word write time		100		µs
–	Block erase time (2Kbyte block)		0.3		s
td(SR-ES)	Time delay from Suspend Request until Erase Suspend			20	ms

Note 1: When not otherwise specified, Vcc = 2.7-5.5V; Topr = 0-60°C.
Note 2: Vcc=5.0V; Topr = 25°C.
Note 3: Definition of E/W cycle: Each block may be written to a variable number of times - up to a maximum of the total number of distinct word addresses - for every block erase. Performing multiple writes to the same address before an erase operation is prohibited.
Note 4: Maximum number of E/W cycles for which operation is guaranteed.
Note 5: Topr = -40-85°C (D3, D7, U3, U7) / -20-85°C (D5, D9, U5, U9).
Note 6: When not otherwise specified, Vcc = 2.7-5.5V; Topr = -20-85°C (D9, U9) / -40-85°C (D7, U7).
Note 7: Table 1.18.2b applies for Block A or B E/W cycles > 1000. Otherwise, use Table 1.18.2a.
Note 8: To reduce the number of E/W cycles, a block erase should ideally be performed after writing as many different word addresses (only one time each) as possible. It is important to track the total number of block erases.
Note 9: Should erase error occur during block erase, attempt to execute clear status register command, then block erase command at least three times until erase error disappears.
Note 10: When Block A or B E/W cycles exceed 100 (D7, D9, U7, U9), select one wait state per block access. 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 setting of FMR17.
Note 11: Customers desiring E/W failure rate information should contact their Renesas technical support representative.

Table 1.18.3. Recommended operating conditions (referenced to V_{CC} = 2.7V to 5.5V at Topr = – 20°C to 85°C / – 40°C to 85°C (Note 3) unless otherwise specified)

Symbol	Parameter			Standard Min.	Standard Typ.	Standard Max.	Unit
V_{CC}	Supply voltage(V_{CC})			2.7		5.5	V
AVcc	Analog supply voltage				Vcc		V
Vss	Supply voltage				0		V
AVss	Analog supply voltage				0		V
V_{IH}	HIGH input voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, RESET, CNVss		0.8Vcc		Vcc	V
		$P7_0$, $P7_1$		0.8Vcc		6.5	V
V_{IL}	LOW input voltage	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$, X_{IN}, RESET, CNVss		0		0.2Vcc	V
$I_{OH\ (peak)}$	HIGH peak output current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$				-10.0	mA
$I_{OH\ (avg)}$	HIGH average output current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$				-5.0	mA
$I_{OL\ (peak)}$	LOW peak output current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$				10.0	mA
$I_{OL\ (avg)}$	LOW average output current	$P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$				5.0	mA
f (X_{IN})	Main clock input oscillation frequency (Note4)	No wait	Vcc=3.0V to 5.5V	0		20	MHz
			Vcc=2.7V to 3.0V	0		33.33 X Vcc -80	MHz
f (X_{CIN})	Subclock oscillation frequency				32.768	50	kHz
f (Ring)	Ring oscillation frequency				1		MHz
f (BCLK)	CPU operation clock			0		20	MHz

Note 1: The mean output current is the mean value within 100ms.

Note 2: The total I_{OL} (peak) output current for all ports must be 80mA max. The total I_{OH} (peak) output current for all ports must be -80mA max.

Note 3: Specify a product of -40°C to 85°C to use it.

Note 4: Relationship between main clock oscillation frequency and supply voltage.

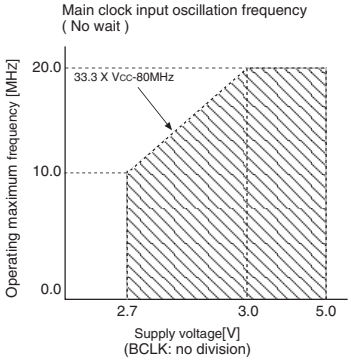


Table 1.18.4. A-D conversion characteristics (Notes 1–3)

Symbol	Parameter		Measuring condition		Standard Min.	Typ.	Max.	Unit
–	Resolution		V_{REF} =V_{CC}				10	Bits
INL	Integral non-linearity error	10 bit	V_{REF}= V_{CC} = 5V	AN_0 to AN_7 input			±3	LSB
			V_{REF}= V_{CC} = 3.3V	AN_0 to AN_7 input			±5	LSB
		8 bit	V_{REF} =V_{CC} =3.3V				±2	LSB
–	Absolute accuracy	10 bit	V_{REF}= V_{CC} = 5V	AN_0 to AN_7 input			±3	LSB
			V_{REF}= V_{CC} = 3.3V	AN_0 to AN_7 input			±5	LSB
		8 bit	V_{REF} =V_{CC} =3.3V				±2	LSB
DNL	Differential non-linearity error						±1	LSB
–	Offset error						±3	LSB
–	Gain error						±3	LSB
R_{LADDER}	Ladder resistance		V_{REF} =V_{CC}		10		40	kΩ
t_{CONV}	Conversion time(10bit), Sample & hold function available		V_{REF} =V_{CC} =5V, $ø_{AD}$=10MHz		3.3			µs
t_{CONV}	Conversion time(8bit), Sample & hold function available		V_{REF} =V_{CC}= 5V, $ø_{AD}$=10MHz		2.8			µs
t_{SAMP}	Sampling time				0.3			µs
V_{REF}	Reference voltage				2.0		V_{CC}	V
V_{IA}	Analog input voltage				0		V_{REF}	V

Note 1: Referenced to V_{CC} = AV_{CC}=V_{REF}=3.3 to 5.5V, V_{SS}=AV_{SS}=0V at Topr = -20 to 85 ˚C / -40 to 85 ˚C unless otherwise specified. Specify a product of -40 to 85 ˚C to use it.
Note 2: AD operation clock frequency ($ø_{AD}$ frequency) must be 10 MHz or less. And divide the f_{AD} if V_{CC} is less than 4.2V, and make $ø_{AD}$ frequency equal to or lower than f_{AD}/2.
Note 3: When not using sample & hold function, $ø_{AD}$ frequency must be ≥ 250 kHz, but less than the upper limit set by Note 2. When using sample & hold function, $ø_{AD}$ frequency must be ≥ 1MHz, but less than the upper limit set by Note 2.

Table 1.18.5. Flash memory version electrical characteristics
(referenced to V_{CC} = 5.0V, at Topr = 25°C, unless otherwise specified)

Parameter		Standard Min.	Typ. ~1K E/W cycles	Typ. ~10K E/W cycles	Max.	Unit
Word program time		–	75	100	–	us
Block erase time:	2Kbyte block	–	0.2	0.3	–	s
	8Kbyte block	–	0.4	–	–	
	16Kbyte block	–	0.7	–	–	
	32Kbyte block	–	1.2	–	–	

Table 1.18.6. Low Voltage Detection Circuit Electrical Characteristics (Note 1)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
Vdet4	Power supply down detection voltage (Note 1)	Vcc = 0.8 to 5.5V	3.3	3.8	4.4	V
Vdet3	Reset level detection voltage (Note 1)		2.2	2.8	3.6	V
Vdet3s	Low voltage reset retention voltage (Note 2)		0.8			V
Vdet3r	Low voltage reset release voltage		2.2	2.9	4.0	V

Note 1: VDET4 > VDET3
Note 2: VDET3s is the min voltage at which "hardware reset 2" is maintained. Below this voltage, "hardware reset 1" (using reset pin) must be applied in order to resume operation.

Table 1.18.7. Power Supply Circuit Timing Characteristics

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
td(R-S)	STOP release time (Note 2)	Vcc = 2.7 to 5.5V			150	µs
td(W-S)	Low power dissipation mode wait mode release time (Note 2)				150	µs
td(M-L)	Time for internal power supply stabilization when main clock oscillation starts				50	µs
td(S-R)	Hardware reset 2 release wait time	Vcc = VDET3r to 5.5V		6 (Note 1)	20	ms
td(E-A)	Low voltage detection circuit operation start time (Note 3)	Vcc = 2.7 to 5.5V			20	µs

Note 1: When Vcc = 5V
Note 2: This is the time between interrupt for (STOP/WAIT) mode release and resumption of CPU clock operation.
Note 3: After enabling low voltage detection, this time is required before proper detection can occur.

Table 1.18.8. Hardware Reset 2 Release Wait Time

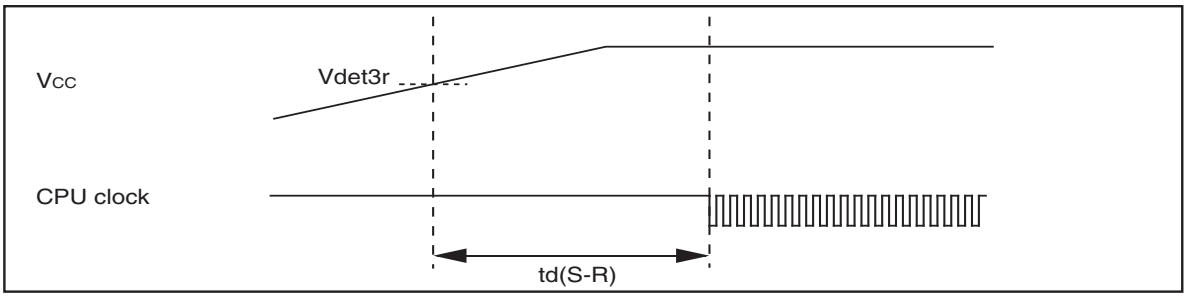
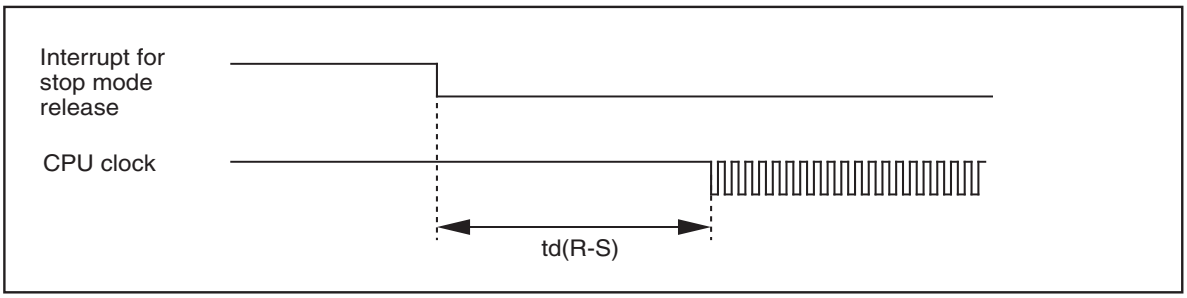


Table 1.18.9. STOP Release Time



Vcc = 5V

Table 1.18.10. Electrical characteristics (referenced to Vcc = 4.2V to 5.5V, Vss = 0V at Topr = – 20°C to 85°C / – 40°C to 85°C (Note 2), f(XIN) = 20MHz unless otherwise specified)

Symbol	Parameter		Measuring condition		Standard Min.	Standard Typ.	Standard Max.	Unit
V_{OH}	HIGH output voltage $P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OH}=-5mA		3.0		Vcc	V
V_{OH}	HIGH output voltage $P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OH}=-200µA		4.7		Vcc	V
V_{OH}	HIGH output voltage X_{OUT}	HIGHPOWER	I_{OL}=-1mA		3.0		Vcc	V
		LOWPOWER	I_{OH}=-0.5mA		3.0		Vcc	
	HIGH output voltage X_{COUT}	HIGHPOWER	With no load applied			2.5		V
		LOWPOWER	With no load applied			1.6		
V_{OL}	LOW output voltage $P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OL}=5mA				2.0	V
V_{OL}	LOW output voltage $P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		I_{OL}=200µA				0.45	V
V_{OL}	LOW output voltage X_{OUT}	HIGHPOWER	I_{OL}=1mA				2.0	V
		LOWPOWER	I_{OL}=0.5mA				2.0	
	LOW output voltage X_{COUT}	HIGHPOWER	With no load applied			0		V
		LOWPOWER	With no load applied			0		
V_{T+}-V_{T-}	Hysteresis $TA0_{IN}$ to $TA4_{IN}$, $TB0_{IN}$ to $TB2_{IN}$, $\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_3}$ to $\overline{INT_5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS_0}$ to $\overline{CTS_2}$, SCL, SDA, CLK_0 to CLK_2, $TA2_{OUT}$ to $TA4_{OUT}$, $\overline{KI_0}$ to $\overline{KI_3}$, RxD_0 to RxD_2				0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis $\overline{RESET}$				0.2		2.5	V
I_{IH}	HIGH input current $P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=5V				5.0	µA
I_{IL}	LOW input current $P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=0V				-5.0	µA
R_{PULLUP}	Pull-up resistance $P1_5$ to $P1_7$, $P6_0$ to $P6_7$, $P7_2$ to $P7_7$, $P8_0$ to $P8_3$, $P8_5$ to $P8_7$, $P9_0$ to $P9_3$, $P10_0$ to $P10_7$		V_I=0V		30.0	50.0	170.0	kΩ
R_{fXIN}	Feedback resistance X_{IN}					1.5		MΩ
R_{fXCIN}	Feedback resistance X_{CIN}					15		MΩ
V_{RAM}	RAM retention voltage		When clock is stopped		2.0			V
Icc	Power supply current (Vcc = 3.0 to 5.5V)	The output pins are open and other pins are Vss	Flash memory version	f(X_{IN}) = 20MHz Square wave, no division		16	19	mA
			Flash memory version	f(X_{CIN}) = 32kHz Square wave, in RAM		25		µA
			Flash memory version	f(X_{CIN}) = 32kHz Square wave, in Flash		420		µA
			Flash memory version	f(X_{CIN}) = 32kHz When a WAIT instruction is executed. (Note 1) Oscillation capacity High		7.5		µA
				f(X_{CIN}) = 32kHz When a WAIT instruction is executed. (Note 1) Oscillation capacity Low		2.0		µA
				Topr = 25°C when clock is stopped		0.8	3.0	µA
IDET4	VDET4 detection dissipation current (Note 3)					0.7	4	µA
IDET3	Reset level detection dissipation current (Note 3)					1.2	8	µA

Note 1: With one timer operated using f_{C32}.
Note 2: Specify a product of -40°C to 85°C to use it.
Note 3: IDET is dissipation current when the following bit is set to "1" (detection circuit enabled)

VCC = 5V

Timing requirements

(referenced to VCC = 5V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C (*) unless otherwise specified)

* : Specify a product of -40 to 85°C to use it.

Table 1.18.11. External Clock Input (XIN 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	22		ns
$t_{w(L)}$	External clock input LOW pulse width	22		ns
t_r	External clock rise time		5	ns
t_f	External clock fall time		5	ns

VCC = 5V

Timing requirements

(referenced to VCC = 5V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C (*) unless otherwise specified)

* : Specify a product of -40 to 85°C to use it.

Table 1.18.12. 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 1.18.13. 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 1.18.14. 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 1.18.15. 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 1.18.16. Timer A input (up/down 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 1.18.17. 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

VCC = 5V

Timing requirements

(referenced to VCC = 5V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C (*) unless otherwise specified)

* : Specify a product of -40 to 85°C to use it.

Table 1.18.18. 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 1.18.19. 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 1.18.20. 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 1.18.21. A-D trigger input

Symbol	Paramete	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 1.18.22. 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	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 1.18.23. External interrupt INTi inputs

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

VCC = 3V

Table 1.18.24. Electrical characteristics (referenced to VCC = 2.7 to 3.3V, VSS = 0V at Topr = – 20ºC to 85ºC / – 40ºC to 85ºC (Note 1), f(XIN) = 10MHz with wait unless otherwise specified)

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
VOH	HIGH output voltage P15 to P17, P60 to P67, P72 to P77, P80 to P83, P85 to P87, P90 to P93, P100 to P107		IOH=-1mA		2.5			V
VOH	HIGH output voltage XOUT	HIGHPOWER	IOH=-0.1mA		2.5			V
		LOWPOWER	IOH=-50µA		2.5			
	HIGH output voltage XCOUT	HIGHPOWER	With no load applied			2.5		V
		LOWPOWER	With no load applied			1.6		
VOL	LOW output voltage P15 to P17, P60 to P67, P70 to P77, P80 to P83, P85 to P87, P90 to P93, P100 to P107		IOL=1mA				0.5	V
VOL	LOW output voltage XOUT	HIGHPOWER	IOL=0.1mA				0.5	V
		LOWPOWER	IOL=50µA				0.5	
	LOW output voltage XCOUT	HIGHPOWER	With no load applied			0		V
		LOWPOWER	With no load applied			0		
VT+-VT-	Hysteresis TA0IN to TA4IN, TB0IN to TB2IN, $\overline{INT0}$, $\overline{INT1}$, $\overline{INT3}$ to $\overline{INT5}$, NMI, $\overline{ADTRG}$, $\overline{CTS0}$ to $\overline{CTS2}$, SCL, SDA, CLK0 to CLK2, TA2OUT to TA4OUT, $\overline{KI0}$ to $\overline{KI3}$, RxD0 to RxD2				0.2		0.8	V
VT+-VT-	Hysteresis $\overline{RESET}$				0.2		1.8	V
IIH	HIGH input current P15 to P17, P60 to P67, P70 to P77, P80 to P83, P85 to P87, P90 to P93, P100 to P107, XIN, $\overline{RESET}$, CNVss		VI=3V				4.0	µA
IIL	LOW input current P15 to P17, P60 to P67, P70 to P77, P80 to P83, P85 to P87, P90 to P93, P100 to P107, XIN, $\overline{RESET}$, CNVss		VI=0V				-4.0	µA
RPULLUP	Pull-up resistance P15 to P17, P60 to P67, P72 to P77, P80 to P83, P85 to P87, P90 to P93, P100 to P107		VI=0V		66	160	500	kΩ
RfXIN	Feedback resistance XIN					3.0		MΩ
RfXCIN	Feedback resistance XCIN					25.0		MΩ
VRAM	RAM retention voltage		When clock is stopped		2.0			V
Icc	Power supply current (Vcc=2.7 to 3.0V)	The output pins are open and other pins are Vss	Flash memory version	f(XIN) = 10MHz Square wave, no division		8	13	mA
			Flash memory version	f(XCIN) = 32kHz Square wave, in RAM		25		µA
			Flash memory version	f(XCIN) = 32kHz Square wave, in Flash		420		µA
			Flash memory version	f(XCIN) = 32kHz When a WAIT instruction is executed.(Note 2) Oscillation capacity High		6.0		µA
				f(XCIN) = 32kHz When a WAIT instruction is executed.(Note 2) Oscillation capacity Low		1.8		µA
				Topr = 25°C when clock is stopped		0.7	3.0	µA
IDET4	VDET4 detection dissipation current (Note 3)					0.6	4	µA
IDET3	Reset level detection dissipation current (Note 3)					0.4	2	µA

Note 1: Specify a product of -40°C to 85°C to use it.
Note 2: With one timer operated using fc32.
Note 3: IDET is dissipation current when the following bit is set to "1" (detection circuit enabled).
IDET4: VC27 bit of VCR2 register
IDET3: VC26 bit of VCR2 register

VCC = 3V

Timing requirements

(referenced to VCC = 3V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C (*) unless otherwise specified)

* : Specify a product of -40 to 85°C to use it.

Table 1.18.25. External Clock Input (XIN 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

VCC = 3V

Timing requirements

(referenced to VCC = 3V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C (*) unless otherwise specified)

* : Specify a product of -40 to 85°C to use it.

Table 1.18.26. 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 1.18.27. 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 1.18.28. 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 1.18.29. 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 1.18.30. Timer A input (up/down 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 1.18.31. 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

(referenced to VCC = 3V, VSS = 0V, at Topr = – 20 to 85°C / – 40 to 85°C (*) unless otherwise specified)

* : Specify a product of -40 to 85°C to use it.

Table 1.18.32. 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)	160		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	160		ns

Table 1.18.33. Timer B input (pulse period measurement mode)

Symbo	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 1.18.34. Timer B input (pulse width measurement mode)

Symbo	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 1.18.35. A-D trigger input

Symbol	Parameter U	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{\text{ADTRG}}$ input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ input LOW pulse width	200		ns

Table 1.18.36. 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	50		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 1.18.37. External interrupt INTi inputs

Symbo	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input LOW pulse width	380		ns

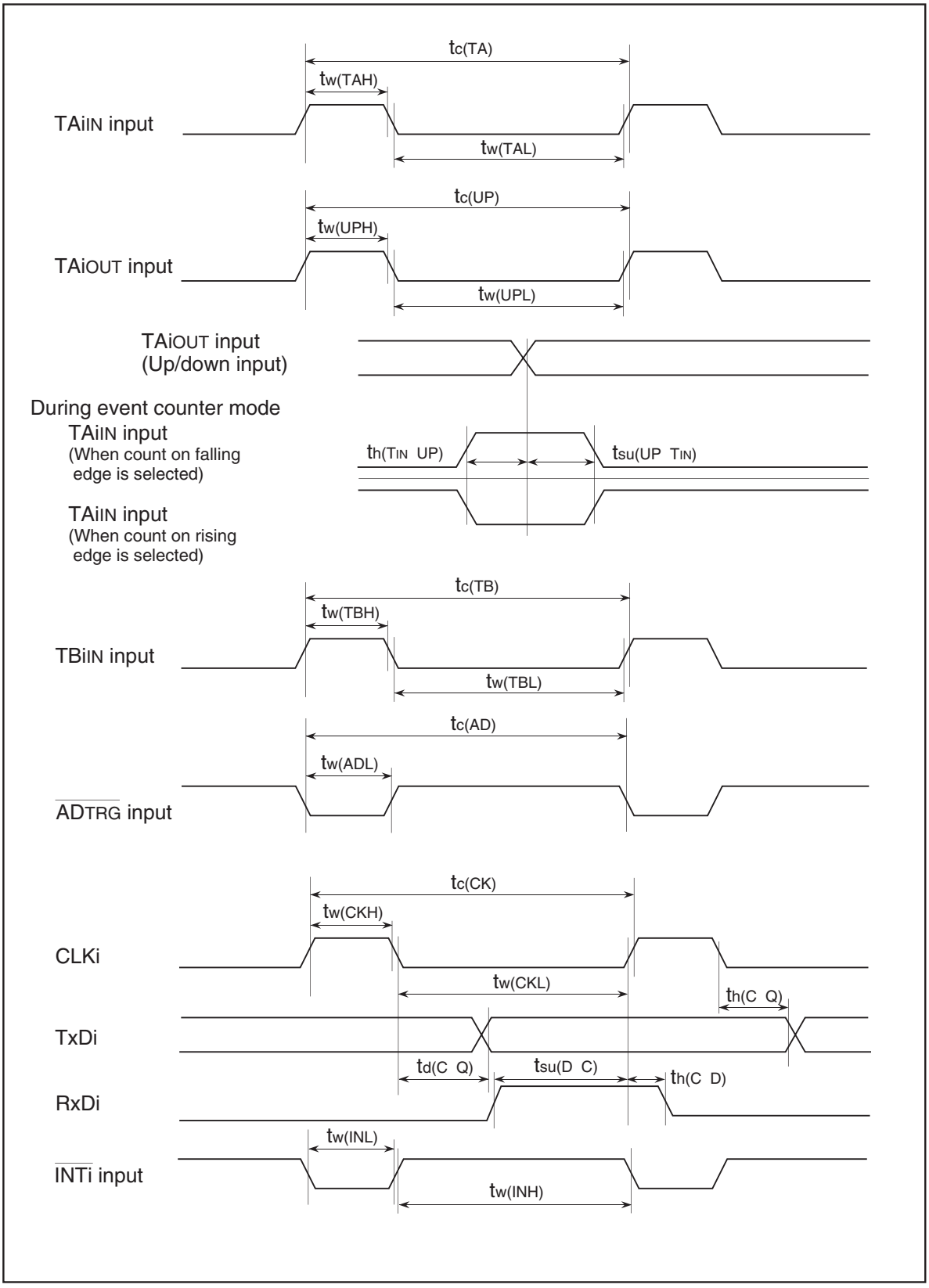


Figure 1.18.1. Timing diagram (1)

Outline Performance

Table 1.19.1 shows the outline performance of the M16C/26 (flash memory version).

Table 1.19.1. Outline performance of the M16C/26 (flash memory version)

Item		Performance
Flash memory operation mode		Three user modes (standard serial I/O, CPU rewrite, parallel I/O)
Erase block division	User ROM area	See Figure 1.19.1
Write method		In units of word.
Erase method		Block erase
Erase/Write (E/W) control method		E/W control by software command
Protect method		Two 8Kbyte user by register lock bit (FMR02)
Number of commands		5 commands
Erase/Write count (Notes 1, 2)		Depends on device code (see Table 1.2b)
Data Retention		10 years
ROM code protect		Parallel I/O and standard serial I/O modes are supported.

Note 1: Block A and Block B are 10,000 times E/W. All other blocks are 1000 times E/W. (Under development; mass production scheduled to start in the 3rd quarter of 2003.)

Note 2: Definition of E/W times:
The E/W times are defined to be per-block erasure times. For example, assume a case whereby a 4-Kbyte Block A is programmed one word at a time and erased once all 2048 write operations have completed. In this case, the block is considered to have been written and erased once.
If a product is 100 times E/W, each block in it can be erased up to 100 times. When 10,000 times E/W, Block A and Block B can each be erased up to 10,000 times. All other blocks can each be erased up to 1000 times.

Flash Memory

The M16C/26 (flash memory version) contains the flash memory that can be rewritten with a single voltage. For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the following sections.

The flash memory is divided into several blocks as shown in Figure 1.19.1, so that memory can be erased one block at a time.

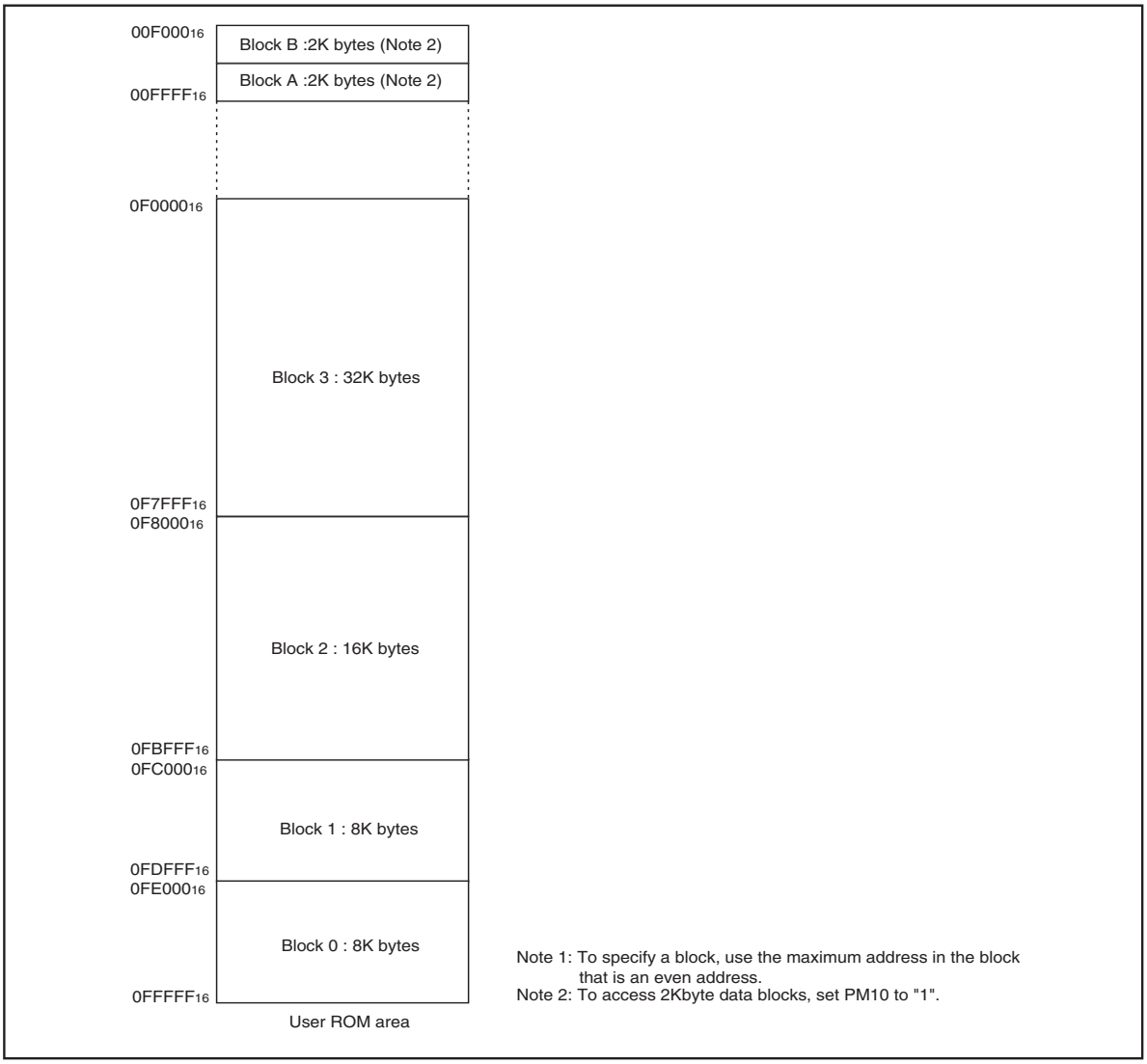


Figure 1.19.1. Block diagram of flash memory (64-Kbyte) version (Note 1)

CPU Rewrite Mode

The CPU rewrite mode is used to perform a read, program, or erase operation on the internal flash memory under control by the central processing unit (CPU). There are two variations of this mode: erase write 0 mode (EW0) in which said operation is performed using a rewrite program residing in memory (i.e. RAM) other than the internal flash memory and erase write 1 mode (EW1) in which said operation is performed using the program residing in the internal flash memory.

In CPU rewrite mode, only the user ROM area shown in Figure 1.19.1 can be rewritten. The Program and Block Erase commands can be executed only for the user ROM area in block intervals.

The EW0 mode control program must be stored in the user ROM area. In EW0 mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to memory locations other than the internal flash memory before it can be executed.

The EW1 mode control program must be stored in the user ROM area. In this mode, the rewrite control program does not have to be transferred to other memory locations before executing it. However, in this mode, the block where the rewrite control program is stored cannot be operated on by the Program or Erase commands.

User Mode and Boot Mode

The control program for CPU rewrite mode must be written into the user ROM area, in parallel I/O mode, beforehand.

Normal user mode is entered when the microcomputer is reset with pulling CNV_{SS} pin low. In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling both the CNV_{SS} and the $P8_6$ [$\overline{CE}$] pins high, the CPU starts operating using the standard serial I/O control program. This mode is called the “boot” mode. When reset is deasserted in boot mode, be sure the $P6_5$ pin is not at high or the $P6_7$ pin is not at low.

Block Address

Block addresses refer to an even address of each block. These addresses are used in the block erase command.

Outline Performance (CPU Rewrite Mode)

In the CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. There are two variations of this mode: erase write 0 mode (EW0) where operation is executed in other than the internal flash memory such as the internal RAM and erase write 1 mode (EW1) where operation is executed in the internal flash memory. One-wait must be set in both modes.

EW0 mode (CPU rewrite mode)

In this mode, the program must be executing out of RAM. The microcomputer is placed in EW0 mode by setting the CPU rewrite mode select bit (address $01B7_{16}$, bit 1) to 1, becoming ready to accept software

commands.

In CPU rewrite mode (whether EW0 or EW1), make sure all software commands and data are written to and read from even addresses (byte address Ao = 0) in 16 bit units. Make sure the write data is written in 16 bit units beginning with an even address. Writing in 16 bit units beginning with an odd address and writing in 8 bit units are inhibited. When using 8-bit software commands, always be sure to write to even addresses. Writing to odd addresses has no effect.

Use software commands to control program and erase operations. Whether program and erase operations have terminated normally or in error can be verified by reading the status register. Even when reading the status register, set to even addresses in the user ROM area also.

EW1 mode (CPU rewrite mode)

In EW1 mode, operation is executed using the control program residing in the internal flash memory. Unlike in EW0 mode, there is no need to transfer the control program to other than the internal flash memory (e.g., internal RAM). However, make sure the control program is located in the 64-Kbyte user block or 4-Kbyte data block.

For EW1 mode, set the EW0 mode select bit (address 01B716, bit 1) to 1 (by writing 0 and then 1 in succession) and set the EW1 mode select bit (address 01B516, bit 6) to 1 (by writing 0 and then 1 in succession). This places the microcomputer in EW1 mode, ready to accept software commands. Although software commands operate the same way as in EW0 mode, there are following differences.

(1) Do not perform Block Erase or Program operations on control program execution blocks (blocks in which the control program is located).

(2) Do not execute the Read Status Register command. (In EW1 mode, this command has no effect.) After erase and program operations are completed during EW1 mode, the microcomputer is in read array mode, and not in read status mode. (During EW0 mode, the microcomputer is in read status mode after operations are completed.)

Therefore, to verify whether program or erase operations have terminated normally or in error, read the flash memory control register 0. Be aware that during EW1 mode, the status register cannot be read.

During EW1 mode, the CPU remains in a hold state while executing erase and program operations. The ports retain the status in which they were before commands were issued. (They do not go Hi-Z even while executing erase or program operations.)

After erase or program operations are completed, the CPU restarts execution of the rest of the control program.

While erase or program operations are underway in EW1 mode, make sure that no interrupts except NMI, watchdog timer, and reset will be generated, and that no DMA transfers will be committed.

Erase-Suspend Feature

The M16C/26 Flash ROM has been designed to be more compact and require a smaller layout footprint. This, as a result, causes longer erase times. The M16C/26 Flash ROM is however not available/accessible during an erase operation. This may sometimes cause time critical interrupt driven operations requiring data/program in the flash to not be satisfied during the erase operation.

To circumvent this issue, the M16C/26 Flash ROM offers a new 'erase-suspend' feature which allows the erase operation to be suspended, and access made available to the flash. The erase operation may subsequently be resumed via software.

There are CPU erase/write (CPUEW) modes available EW0 (execution out of RAM) and EW1 (execution out of FLASH). The erase-suspend feature is different in each of these modes. Please note that 1-wait needs to be set in CPUEW operations.

EW0:

In EW0, program code is executed out of the RAM. After the erase command has been executed, program execution continues in the RAM. As stated earlier the FLASH is not accessible during an erase operation. If there is a request for data/code from the FLASH (via a maskable peripheral/external interrupt), the interrupt must first request an erase-suspend. This is achieved by setting bits FMR40 (SUSPEND_ENABLE) and FMR41 (SUSPEND_REQUEST). The routine then polls FMR46 (SUSPENDL) until it is set. At this point the erase has been suspended and the flash is accessible. Once the required accesses are complete FMR41 (SUSPEND_REQUEST) is cleared and the routine is completed. The erase operation resumes and continues to completion or until another erase-suspend request occurs.

User actions:

1.0 Execute erase command out of RAM.
2.0 Maskable Interrupt request.
  2.1 Set FMR40 & FMR41.
  2.2 Poll FMR46 until '1'.
  2.3 Access flash data/code.
  2.4 Clear FMR41.
  2.5 Return
3.0 Continue execution out of RAM

FMR40 may also be set before the erase command is executed, instead of in the interrupt routine.

EW1:

In EW1, program code is executed out of the FLASH. First FMR40 (SUSPEND_ENABLE) must be set to allow the erase-suspend feature. After ther erase command is executed the CPU goes into HOLD. A maskable interrupt will set FMR41 (SUSPEND_REQUEST) if in an erase operation. Once the erase is suspended the HOLD is deasserted and execution from the FLASH continues. The interrupt routine can now be servised, after which control is returned to the main program. If the SUSPEND_REQUEST has been set, it should be cleared, and when the erase resumes the CPU goes back into HOLD until the erase operation is complete or another interrupt occurs.

User actions:

1.0 Set FMR40
2.0 Execute erase command out of FLASH.
3.0 HOLD_POINT (in HOLD)
4.0 Maskable Interrupt request. (FMR41 set by h/w)
  4.1 Access flash data/code.
  4.2 Return
5.0 if FMR41 is set, clear FMR41, jump to HOLD_POINT.
6.0 Continue execution out of FLASH.

Register Description

Figure 1.20.1 shows the flash identification register, flash memory control register 0 and flash memory control register 1.

Flash memory control register 0 (FMR0):

Bit 0 of the flash memory control register 0 is the RY/BY status flag used exclusively to read the operating status of the flash memory. During programming, erase, and erase-suspend operations, it is “0”. Otherwise, it is “1”.

Bit 1 of the flash memory control register 0 is the CPU rewrite mode select bit. The CPU rewrite mode is entered by setting this bit to “1”, so that software commands become acceptable. To set this bit to “1”, it is necessary to write “0” and then write “1” in succession. To set this bit to “0” by only writing a “0”.

Bit 2 of the flash memory control register 0 allows program and erase operations to occur on the two 8K byte user blocks. When this bit is set to "0", no program or erase operations can occur on these blocks. To permit program and erase operations to occur on these blocks, set this bit to a "1". To set this bit to “1”, it is necessary to write “0” and then write “1” in succession. This bit can be manipulated only when the CPU rewrite mode select bit = “1” (Bit 1 of this register).

Bit 3 of the flash memory control register is the flash memory reset bit used to reset the control circuit of the internal flash memory. This bit is used when exiting CPU rewrite mode and when flash memory access has failed. When the CPU rewrite mode select bit is “1”, writing “1” for this bit resets the control circuit. To release the reset, it is necessary to set this bit to “0” when RY/BY status flag is “1”. Also when this bit is set to “1”, power is not supplied to the internal flash memory, thus power consumption can be reduced. However, in this state, the internal flash memory cannot be accessed. To set this bit to “1”, it is necessary to write “0” and then write “1” in succession when the CPU rewrite mode select bit is “1”. Use this bit mainly in the low speed mode (when XCIN is the count source of BCLK).

It is not particularly necessary to set bit 3 of the flash memory control register 0 on return from STOP/WAIT.

Figure 1.20.2c shows a flowchart for shifting to the low power dissipation mode. Always perform operation as indicated in these flowcharts.

Bit 6 of the flash memory control register 0 is the program status flag used exclusively to read the operating status of the auto program operation. If a program error occurs, it is set to “1”. Otherwise, it is “0”.

Bit 7 of the flash memory control register 0 is the erase status flag used exclusively to read the operating status of the auto erase operation. If an erase error occurs, it is set to “1”. Otherwise, it is “0”.

Figure 1.20.2a shows a EW0 mode set/reset flowchart, figure 1.20.2b shows a EW1 mode set/reset flowchart. Always perform operation as indicated in these flowcharts.

Flash memory control register 1 (FMR1):

Bit 1 allows the user to enter EW1 mode. This bit is relevant only if bit FMR01 is set.

Bit 7, when set to "0", inserts one wait state 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 setting of FMR17. In cases where E/W cycles to Block A or B exceed 100 times (D7, D9, U7, U9), please set FMR17 to "1" (with wait).

Flash memory control register 4 (FMR4):

Bit 0 must be set to enable the erase-suspend feature.

Bit 1 is to be used to request a suspend of an erase operation. This bit is set automatically in EW1 by a maskable interrupt, and by software in EW0. This bit is to be always cleared by software at the end of the erase suspend.

Bit 6 indicates suspend status. Poll this bit in EW0 after requesting a suspend, before accessing the flash.

Flash memory control register 0

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

Symbol: FMR0 | Address: 01B7$_{16}$ | When reset: 00000001$_2$

Bit symbol	Bit name	Function	R	W
FMR00	RY/$\overline{BY}$ status flag	0: Busy (being written or erased) 1: Ready	○	×
FMR01	EW entry bit (Note 1)	0: Normal mode (Software commands invalid) 1: EW mode (Software commands acceptable)	○	○
FMR02	8Kbyte EW mode enable bit (Note 2)	0: EW mode disabled on 8Kbyte blocks 1: EW mode enabled on 8Kbyte blocks	○	○
FMSTP	Flash memory reset bit (Note 3, Note 5)	0: Normal operation 1: Reset	○	○
Reserved bits		Must always be set to 0	○	○
FMR06	Program status flag (Note 6)	0: Pass 1: Error	○	×
FMR07	Erase status flag (Note 6)	0: Pass 1: Error	○	×

Flash memory control register 1

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

Symbol: FMR1 | Address: 01B5$_{16}$ | When reset: 0XX00101$_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to "1"	○	×
FMR11	EW mode select bit (Note 1)	0: EW0 mode 1: EW1 mode	○	○
Reserved bit		Must always be set to "1"	○	×
Reserved bits		Must always be set to "0"	○	○
Nothing is assigned. In an attempt to write to these bits, write 0. The value, if read, turns out to be indeterminate.			–	–
FMR17	Blocks A and B access wait bit (Note 7)	0: PM17 controls wait state insertion 1: Wait state inserted (1 wait)	○	○

Flash memory control register 4

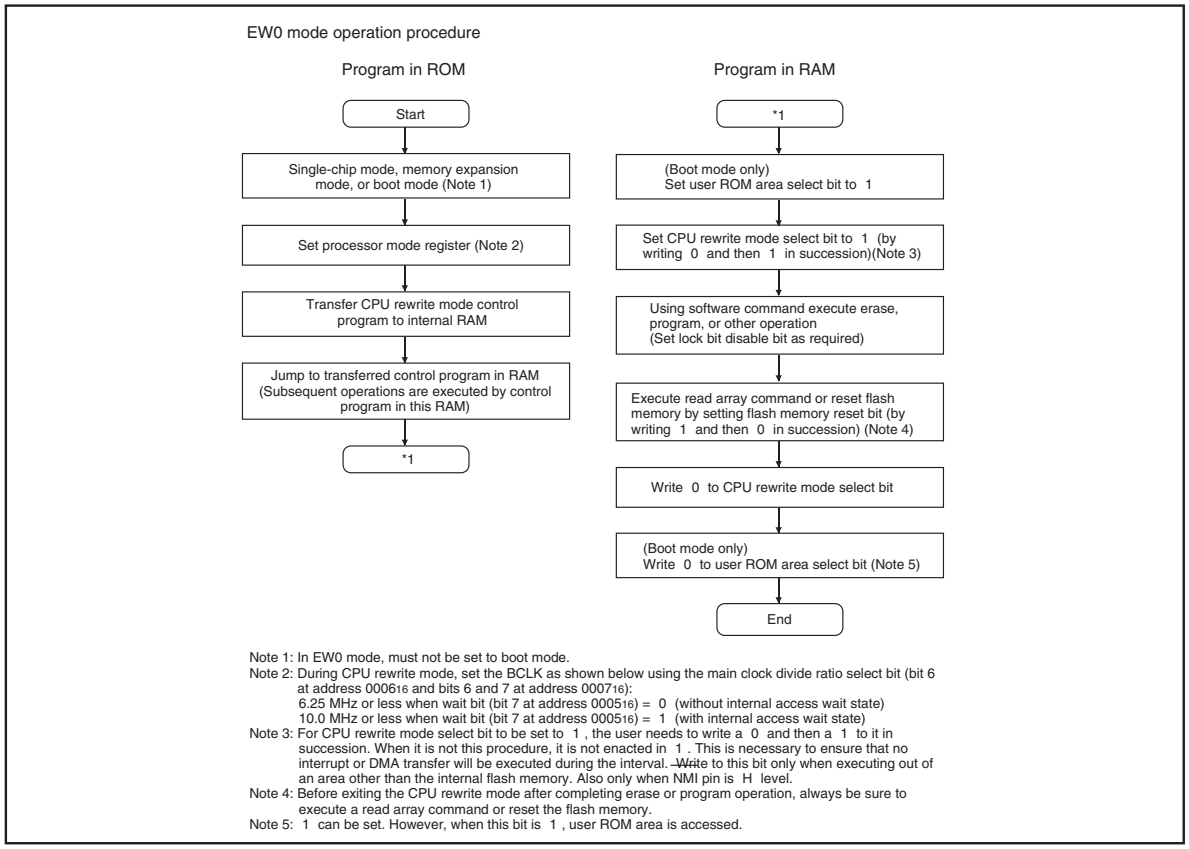
b7	b6	b5	b4	b3	b2	b1	b0
0		0	0	0	0		

Symbol: FMR4 | Address: 01B3$_{16}$ | When reset: 01000000$_2$

Bit symbol	Bit name	Function	R	W
FMR40	Suspend enable (Note 1)	0: Invalid 1: Valid	○	○
FMR41	Suspend request (Note 4)	0: Erase restart 1: Suspend request	○	○
Reserved bits		Must always be set to 0	○	×
FMR46	Suspend status	0: Erase Active 1: Erase Inactive	○	×
Reserved bits		Must always be set to 0	○	○

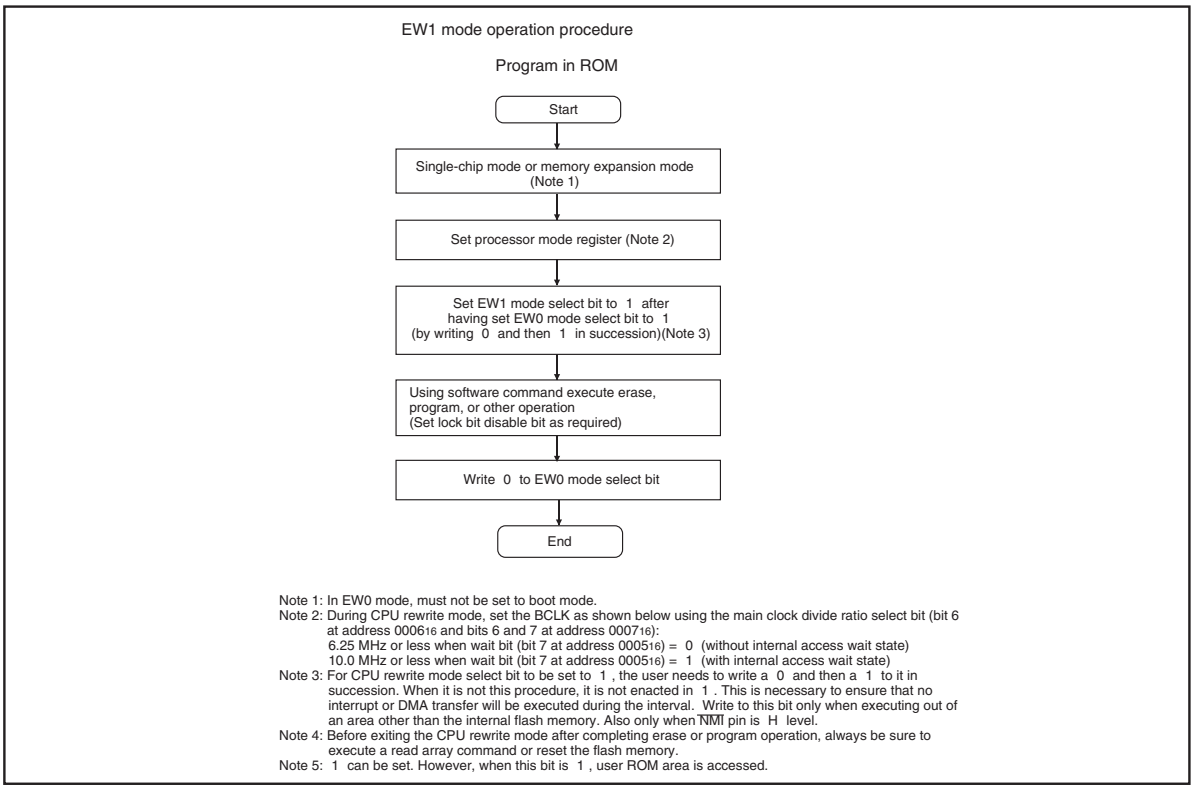
Note 1: To set this bit to "1", write a "0" and then a "1" to it in succession. Make sure no interrupts or DMA transfers occur before completion of these two write operations. While in EW0 mode, write to this bit from a program located in other than flash memory.
Note 2: To set this bit to "1", first ensure that the CPU rewrite mode select bit is set = "1"; then write a "0" followed by a "1" to FMR02 in succession. Make sure no interrupts or DMA transfers occur before completion of these last two successive write operations. Additionally, bit FMR01 must also be set to "1" prior to setting this bit to a "1".
Note 3: Effective only when the CPU rewrite mode select bit = "1". After writing "1", write "0" when $\overline{RY}$/BY status flag is "1".
Note 4: This bit becomes valid only when FMR40 = "1" and when in an erase operation.
• In EW0 mode, this bit can be set to "0" or "1" by program.
• In EW1 mode, this bit is automatically set to "1" when a maskable interrupt occurs. It can NOT be set to "1" by program. (Writing "0" is available.)
Note 5: Write to this bit from a program in other than the flash memory.
Note 6: This flag is cleared to "0" by executing the Clear Status command.
Note 7: In cases where E/W cycles to Block A or B exceed 100 times (D7, D9, U7, U9), please set this bit to "1" (with wait). 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 setting of FMR17.

Figure 1.20.1. Flash memory control registers 0, 1



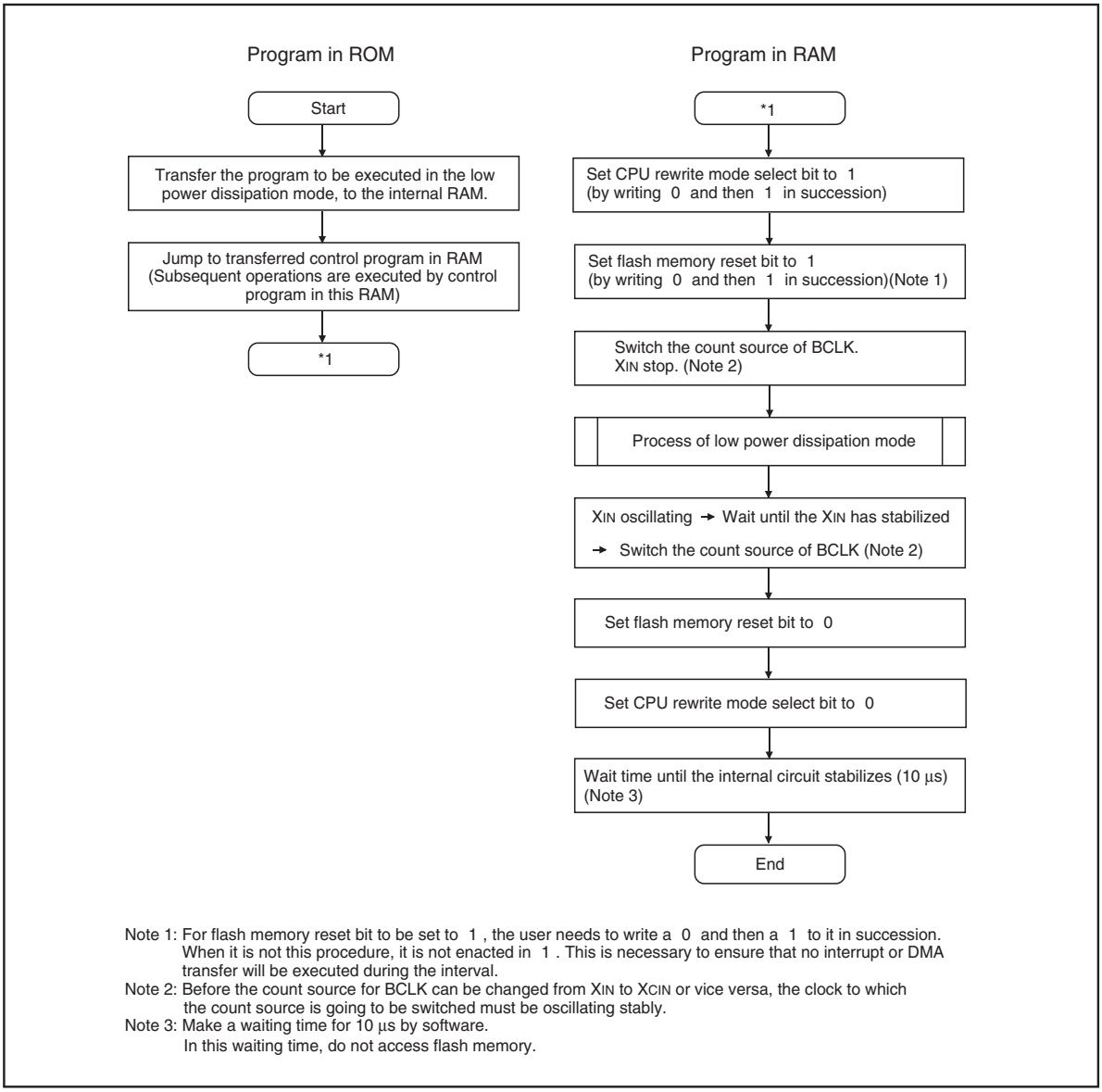
Note 1: In EW0 mode, must not be set to boot mode.
Note 2: During CPU rewrite mode, set the BCLK as shown below using the main clock divide ratio select bit (bit 6 at address 0006_{16} and bits 6 and 7 at address 0007_{16}):
6.25 MHz or less when wait bit (bit 7 at address 0005_{16}) = 0 (without internal access wait state)
10.0 MHz or less when wait bit (bit 7 at address 0005_{16}) = 1 (with internal access wait state)
Note 3: For CPU rewrite mode select bit to be set to 1 , the user needs to write a 0 and then a 1 to it in succession. When it is not this procedure, it is not enacted in 1 . This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. ~~Write~~ to this bit only when executing out of an area other than the internal flash memory. Also only when NMI pin is H level.
Note 4: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute a read array command or reset the flash memory.
Note 5: 1 can be set. However, when this bit is 1 , user ROM area is accessed.

Figure 1.20.2a. EW0 mode set/reset flowchart



Note 1: In EW0 mode, must not be set to boot mode.
Note 2: During CPU rewrite mode, set the BCLK as shown below using the main clock divide ratio select bit (bit 6 at address 0006_{16} and bits 6 and 7 at address 0007_{16}):
6.25 MHz or less when wait bit (bit 7 at address 0005_{16}) = 0 (without internal access wait state)
10.0 MHz or less when wait bit (bit 7 at address 0005_{16}) = 1 (with internal access wait state)
Note 3: For CPU rewrite mode select bit to be set to 1 , the user needs to write a 0 and then a 1 to it in succession. When it is not this procedure, it is not enacted in 1 . This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. Write to this bit only when executing out of an area other than the internal flash memory. Also only when $\overline{\text{NMI}}$ pin is H level.
Note 4: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute a read array command or reset the flash memory.
Note 5: 1 can be set. However, when this bit is 1 , user ROM area is accessed.

Figure 1.20.2b. EW1 mode set/reset flowchart



Note 1: For flash memory reset bit to be set to 1 , the user needs to write a 0 and then a 1 to it in succession. When it is not this procedure, it is not enacted in 1 . This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.
Note 2: Before the count source for BCLK can be changed from XIN to XCIN or vice versa, the clock to which the count source is going to be switched must be oscillating stably.
Note 3: Make a waiting time for 10 μs by software.
In this waiting time, do not access flash memory.

Figure 1.20.2c. Shifting to the low power dissipation mode flowchart

Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

(1) Operation speed

During CPU rewrite mode (EW0/EW1 mpde), set the BCLK as shown below using the main clock divide ratio select bit (bit 6 at address 0006_{16} and bits 6 and 7 at address 0007_{16}):

10.0 MHz or less when wait bit (bit 7 at address 0005_{16}) = 1 (with internal access wait state)

Note : Always perform it with a condition mentioned above.

(2) Instructions inhibited against use

The instructions listed below cannot be used during EW0 mode because they refer to the internal data of the flash memory:

UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

(3) Interrupts inhibited against use

The address match interrupt cannot be used during EW0 mode because they refer to the internal data of the flash memory. If interrupts have their vector in the variable vector table, they can be used by transferring the vector into the RAM area. The NMI and watchdog timer interrupts can be used to automatically initialize the flash identification register and flash memory control register 0 to “0”, then return to normal operation. However, these two interrupts' jump addresses are located in the fixed vector table and there must exsist a routine to be executed. Since the rewrite operation is halted when an NMI or watchdog timer interrupts occurs, you must reset the CPU rewite mode select bit to “1” and the perform the erase/program operation again.

(4) How to access

For EW0 mode select bit and lock bit disable select bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. When it is not this procedure, it is not enacted in “1”. This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. Also only when NMI pin is “H” level.

(5) Writing in the user ROM area

If power is lost while rewriting blocks that contain the flash rewrite program with the CPU rewrite mode, those blocks may not be correctly rewritten and it is possible that the flash memory can no longer be rewritten after that. Therefore, it is recommended to use the standard serial I/O mode or parallel I/O mode to rewrite these blocks.

(6) STOP/WAIT

Both instructions disrupt erase/program operation, and the state of the blocks operated upon is not guaranteed. Inhibit these instructions when in CPU rewrite mode.

Software Commands

Table 1.20.1 lists the software commands available with the M16C/26 (flash memory version).

After setting the CPU rewrite mode select bit to 1, write a software command to specify an erase or program operation. Note that when entering a software command, the upper byte (D_8 to D_{15}) is ignored.

The content of each software command is explained below.

The first bus cycle commands must be written to an even address in the user ROM area.

Table 1.20.1. List of software commands (CPU rewrite mode)

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	Write	X	FF_{16}			
Read status register	Write	X	70_{16}	Read	X	SRD (Note 2)
Clear status register	Write	X	50_{16}			
Program (Note 3)	Write	WA	40_{16}	Write	WA (Note 3)	WD (Note 3)
Block erase	Write	X	20_{16}	Write	BA (Note 4)	$D0_{16}$

Note 1: When a software command is input, the high-order byte of data (D_8 to D_{15}) is ignored.
Note 2: SRD = Status Register Data (Set an address to even address in the user ROM area)
Note 3: WA = Write Address (even address), WD = Write Data (16-bit data)
Note 4: BA = Block Address (Enter the maximum address of each block that is an even address.)
Note 5: X denotes a given address in the user ROM area (that is an even address).

Read Array Command (FF_{16})

The read array mode is entered by writing the command code "FF_{16}" in the first bus cycle. When an even address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D_0–D_{15}), 16 bits at a time.

The read array mode is retained intact until another command is written.

However, please begin to read data in the following procedures when a user uses read array command after program command.

(1) Set FF_{16}, FF_{16}, FF_{16}, FF_{16} to arbitrary continuing four address beforehand
(2) Input the top address which FF_{16} was set at (in read array mode)
(3) Input the top address till $FFFF_{16}$ agrees with the value that begins to have been read
(4) Input top address +2
(5) Input top address +2 till $FFFF_{16}$ agrees with the value that begins to have been read
(6) Input an arbitrary address

Read Status Register Command (70_{16})

When the command code "70_{16}" is written in the first bus cycle, the content of the status register is read out at the data bus (D_0–D_7) by a read in the second bus cycle (Set an address to even address in the user ROM area).

The status register is explained in the next section.

In EW1 mode, cannot use read status register command.

Clear Status Register Command (50_{16})

This command is used to clear the bits SR4 and SR5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "50_{16}" in the first bus cycle.

Program Command (40$_{16}$)

Program operation starts when the command code "40$_{16}$" is written in the first bus cycle. Then, if the address and data to program are written in the 2nd bus cycle, program operation (data programming and verification) will start. Make an address in the first bus cycle same as an address to program by the second bus cycle.

Whether the write operation is completed can be confirmed by reading the status register or the RY/BY status flag. When the program starts, the read status register mode is accessed automatically and the content of the status register is read into the data bus (D0 - D7). The status register bit 7 (SR7) is set to 0 at the same time the write operation starts and is returned to 1 upon completion of the write operation. In this case, the read status register mode remains active until the Read Array command (FF$_{16}$) is written.

The RY/BY status flag is 0 during write operation and 1 when the write operation is completed as is the status register bit 7.

At program end, program results can be checked by reading the status register.

Figure 1.20.3 shows an example of a program flowchart.

Each block of the flash memory can be write protected by using a lock bit. For details, refer to the section where the data protect function is detailed.

Additional writes to the already programmed pages are prohibited.

Do a command to use in right after of program command as follows

Make an address in the first bus cycle same as an address to program by the second bus cycle of program command.

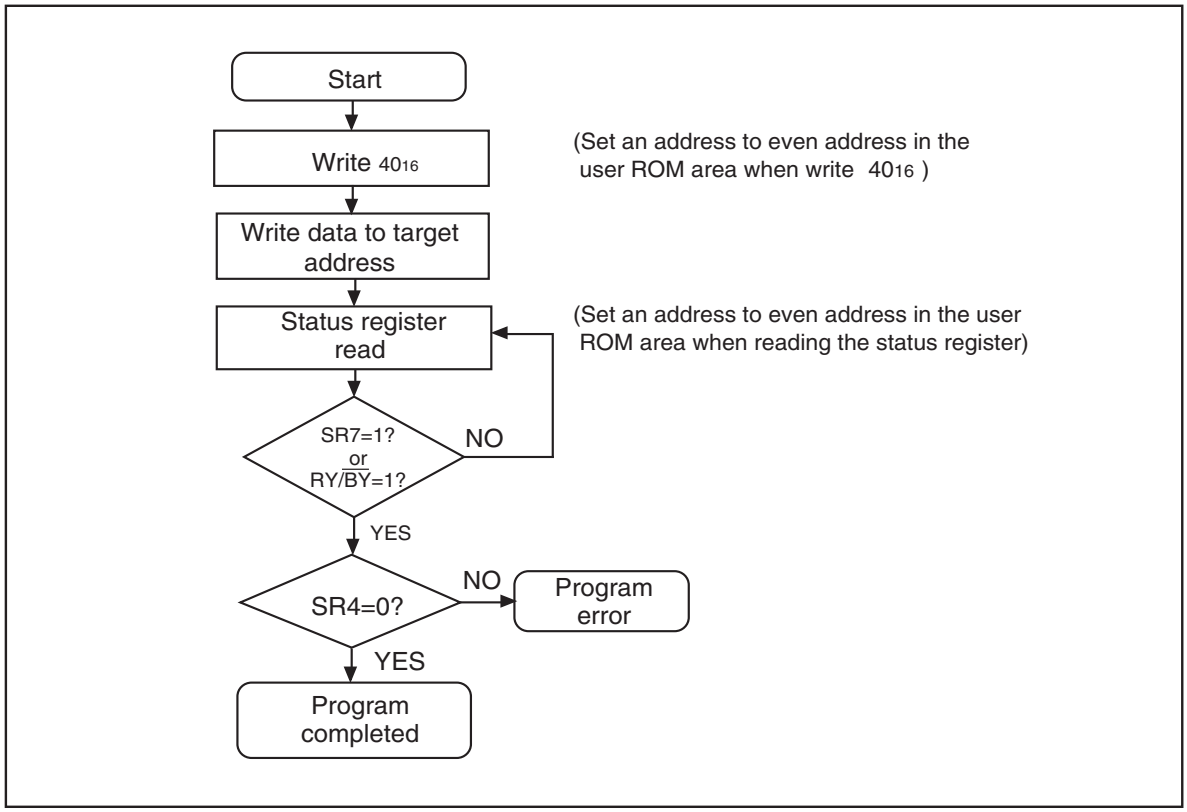


Figure 1.20.3. Program flowchart

Block Erase Command (20₁₆/D0₁₆)

By writing the command code "20₁₆" in the first bus cycle and the confirmation command code "D0₁₆" in the second bus cycle that follows to the block address of a flash memory block, the system initiates an auto erase (erase and erase verify) operation.

Whether the auto erase operation is completed can be confirmed by reading the status register or the flash memory control register 0. At the same time the auto erase operation starts, the read status register mode is automatically entered, so the content of the status register can be read out. The status register bit 7 (SR7) is set to 0 at the same time the auto erase operation starts and is returned to 1 upon completion of the auto erase operation. In this case, the read status register mode remains active until the Read Array command (FF₁₆) or Read Lock Bit Status command (71₁₆) is written or the flash memory is reset using its reset bit.

The RY/BY status flag of the flash memory control register 0 is 0 during auto erase operation and 1 when the auto erase operation is completed as is the status register bit 7.

After the auto erase operation is completed, the status register can be read out to know the result of the auto erase operation. For details, refer to the section where the status register is detailed.

Figure 1.20.4 shows an example of a block erase flowchart.

Each block of the flash memory can be protected against erasure by using a lock bit. For details, refer to the section where the data protect function is detailed.

During EW1 mode, do not execute this command on blocks in which the control program is stored.

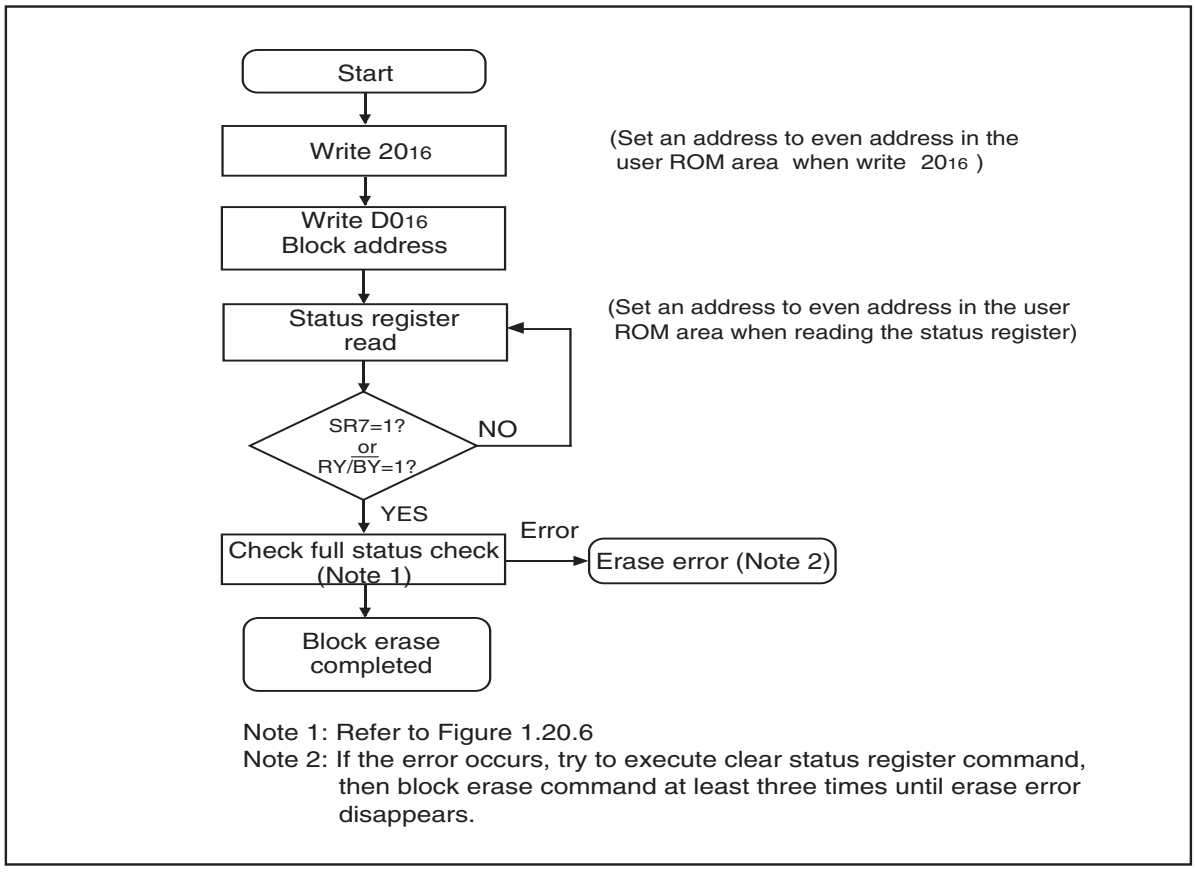
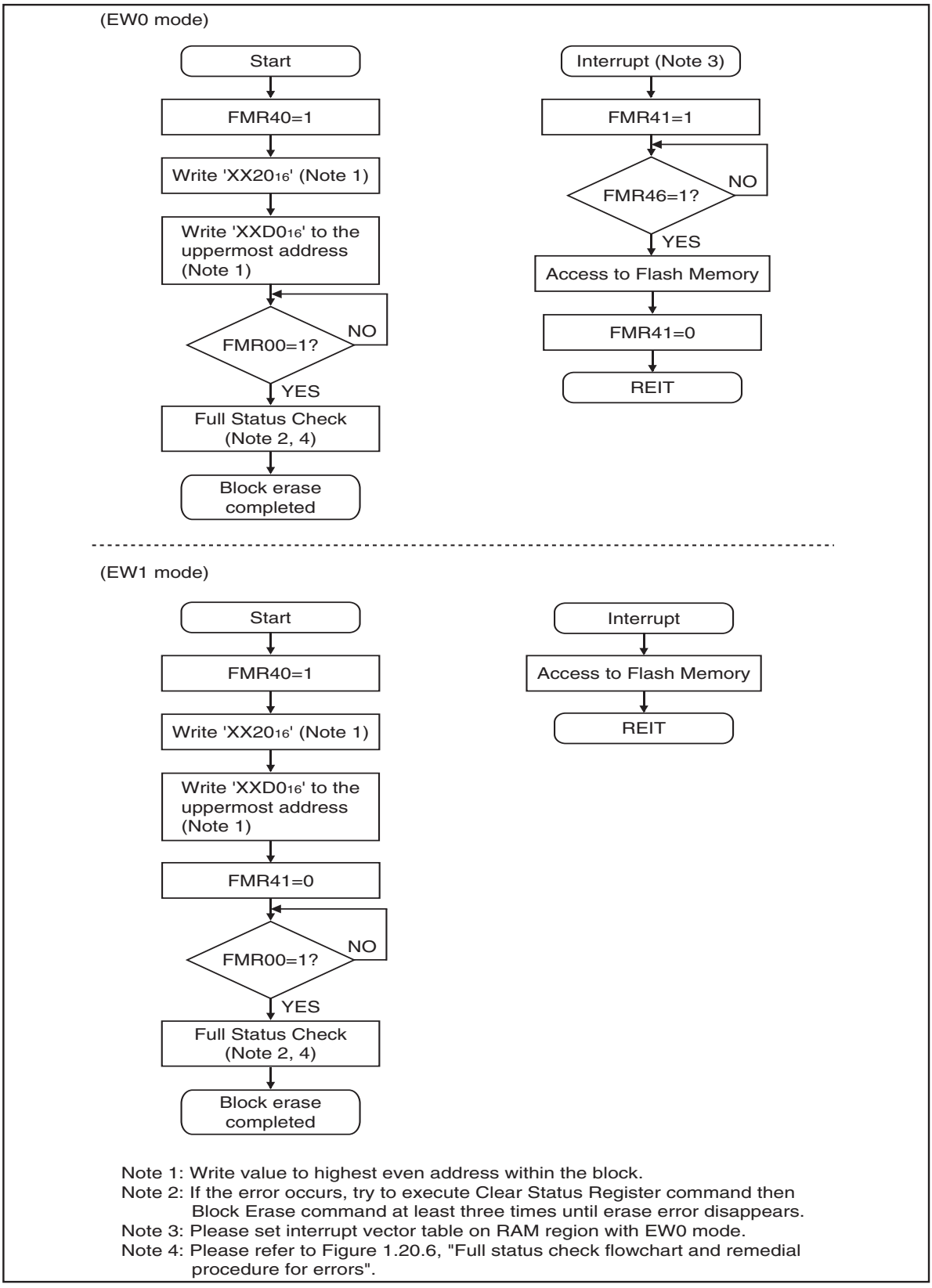


Figure 1.20.4. Block erase flowchart



Note 1: Write value to highest even address within the block.
Note 2: If the error occurs, try to execute Clear Status Register command then Block Erase command at least three times until erase error disappears.
Note 3: Please set interrupt vector table on RAM region with EW0 mode.
Note 4: Please refer to Figure 1.20.6, "Full status check flowchart and remedial procedure for errors".

Figure 1.20.5. Block erase flowchart with erase suspend function

Status Register

The status register shows the operating state of the flash memory and whether erase operations and programs ended successfully or in error. It can be read only in the EW0 mode, in the following ways. The status cannot be read out in the EW1 mode and during suspend.

(1) By reading an arbitrary even address from the user ROM area after writing the read status register command (70_{16})

(2) By reading an arbitrary even address from the user ROM area in the period from when the program starts or erase operation starts to when the read array command (FF_{16}) is input

Table 1.20.2 shows the status register.
In the EW1 mode, the status corresponding to the below is stored in the flash memory control register 0. Read this register for status check.
Also, the status register can be cleared in the following way.
(1) By writing the clear status register command (50_{16})
After a reset, the status register is set to "80_{16}".
Each bit in this register is explained below.

Sequencer status (SR7) / FMR00

After power-on, the sequencer status is set to 1(ready).
The sequencer status indicates the operating status of the device. This status bit is set to "0" (busy) during write or erase operation and is set to "1" upon completion of these operations.

Erase status (SR5) / FMR07

The erase status informs the operating status of erase operation to the CPU. When an erase error occurs, it is set to "1".
The erase status is reset to "0" when cleared.

Program status (SR4)

The program status informs the operating status of write operation to the CPU. When a write error occurs, it is set to "1".
The program status is reset to "0" when cleared.
When an erase command is in error (which occurs if the command entered after the block erase command (20_{16}) is not the confirmation command ($D0_{16}$), both the program status and erase status (SR5) are set to "1".
When the program status or erase status ="1", only the following flash commands will be accepted: Read Array, Read Status Register, and Clear Status Register.

Also, in one of the following cases, both SR4 and SR5 are set to 1 (command sequence error):

(1) When the valid command is not entered correctly

(2) When the data entered in the second bus cycle of lock bit program (77_{16}/$D0_{16}$), block erase (20_{16}/$D0_{16}$), or erase all unlock blocks ($A7_{16}$/$D0_{16}$) is not the $D0_{16}$ or FF_{16}. However, if FF_{16} is entered, read array is assumed and the command that has been set up in the first bus cycle is canceled.

Table 1.20.2. Definition of each bit in status register

Each SRD bit	Status name	Definition	
		"1"	"0"
SR7 (D_7)	Write state machine (WSM)	Ready	Busy
SR6 (D_6)	Reserved	–	–
SR5 (D_5)	Erase status	Terminated in error	Terminated normally
SR4 (D_4)	Program status	Terminated in error	Terminated normally
SR4 (D_3)	Program status after program	Terminated in error	Terminated normally
SR2 (D_2)	Reserved	–	–
SR1 (D_1)	Reserved	–	–
SR0 (D_0)	Reserved	–	–

Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 1.20.6 shows a full status check flowchart and the action to be taken when each error occurs.

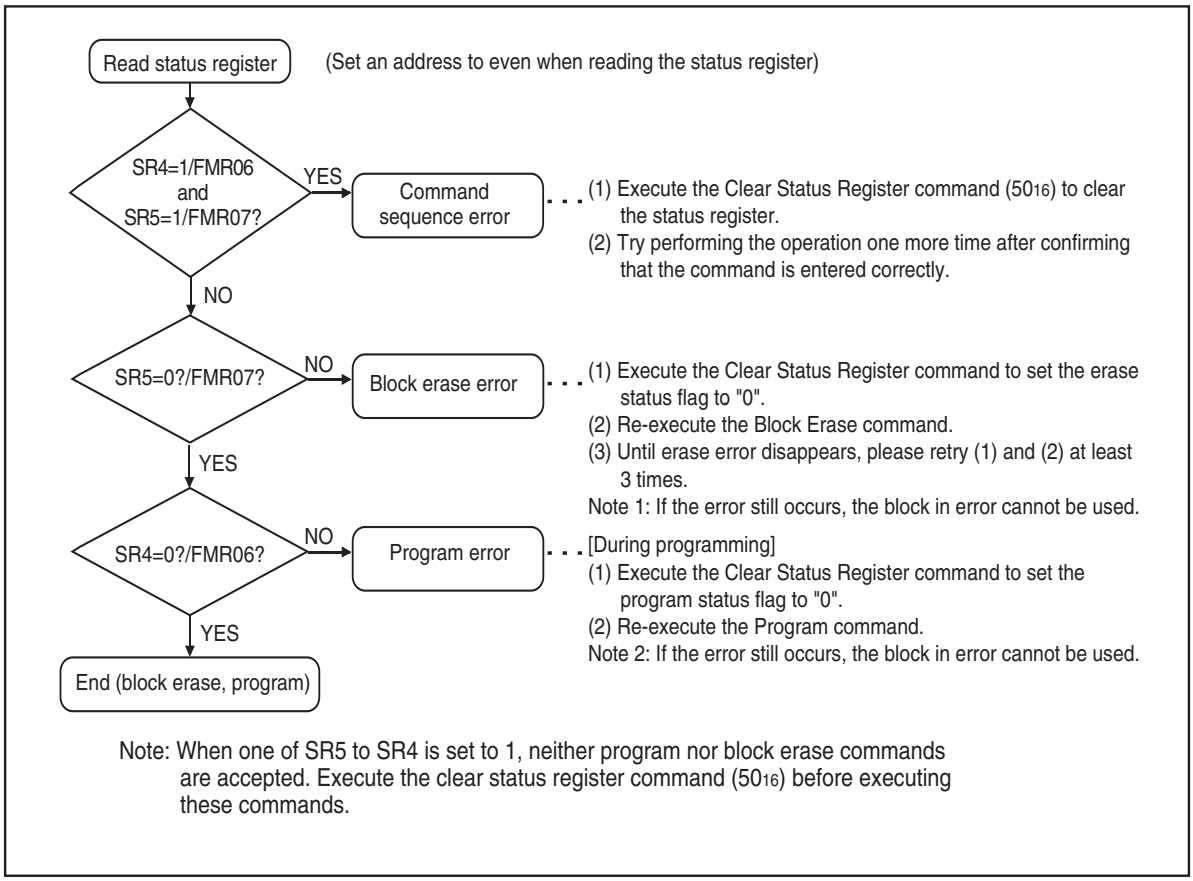


Figure 1.20.6. Full status check flowchart and remedial procedure for errors

Functions To Inhibit Rewriting Flash Memory Version

To prevent the contents of the flash memory version from being read out or rewritten easily, the device incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

ROM code protect function

The ROM code protect function is used to prohibit reading out or modifying the contents of the flash memory during parallel I/O mode and is set by using the ROM code protect control address register (0FFFFF16). Figure 1.21.1 shows the ROM code protect control address (0FFFFF16). (This address exists in the user ROM area.)

If either bit of ROMCP1 is set to '0', ROM code protect level 1 is turned on, so that the contents of the flash memory are protected against readout and modification.

If either bit of ROMCP2 is set to '0', ROM code protect level 2 is turned on, enabling additional protection against readout and modification (such as by a production LSI tester). If both level 1 and level 2 are enabled, level 2 is selected by default.

If both of the two ROM code protect reset bits are set to “00,” ROM code protect is turned off, so that the contents of the flash memory version can be read out or modified. Once ROM code protect is turned on, the contents of the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial I/O or some other mode to rewrite the contents of the ROM code protect reset bits.

ROM code protect control address

b7	b6	b5	b4	b3	b2	b1	b0
						1	1

Symbol: ROMCP
Address: 0FFFFF16
When reset: FF16

Bit symbol	Bit name	Function
Reserved bit		Always set this bit to 1.
ROMCP2	ROM code protect level 2 set bit (Note 1, 2)	b3 b2 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled
ROMCR	ROM code protect reset bit (Note 3)	b5 b4 00: Protect removed 01: Protect set bit effective 10: Protect set bit effective 11: Protect set bit effective
ROMCP1	ROM code protect level 1 set bit (Note 1)	b7 b6 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled

Note 1: When ROM code protect is turned on, the on-chip flash memory is protected against readout or modification in parallel input/output mode.
Note 2: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester, etc. also is inhibited. Customers desiring to use ROM code protect level 2 should first contact their Renesas technical support representative.
Note 3: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and ROM code protect level 2. However, since these bits cannot be changed in parallel input/output mode, they need to be rewritten in serial input/output or some other mode

Figure 1.21.1. ROM code protect control address

ID Code Check Function

Use this function in standard serial I/O mode. When the contents of the flash memory are not blank, the ID code sent from the peripheral unit is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the peripheral unit are not accepted. The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are $0FFFDF_{16}$, $0FFFE3_{16}$, $0FFFEB_{16}$, $0FFFEF_{16}$, $0FFFF3_{16}$, $0FFFF7_{16}$, and $0FFFFB_{16}$. Write a program which has had the ID code preset at these addresses to the flash memory.

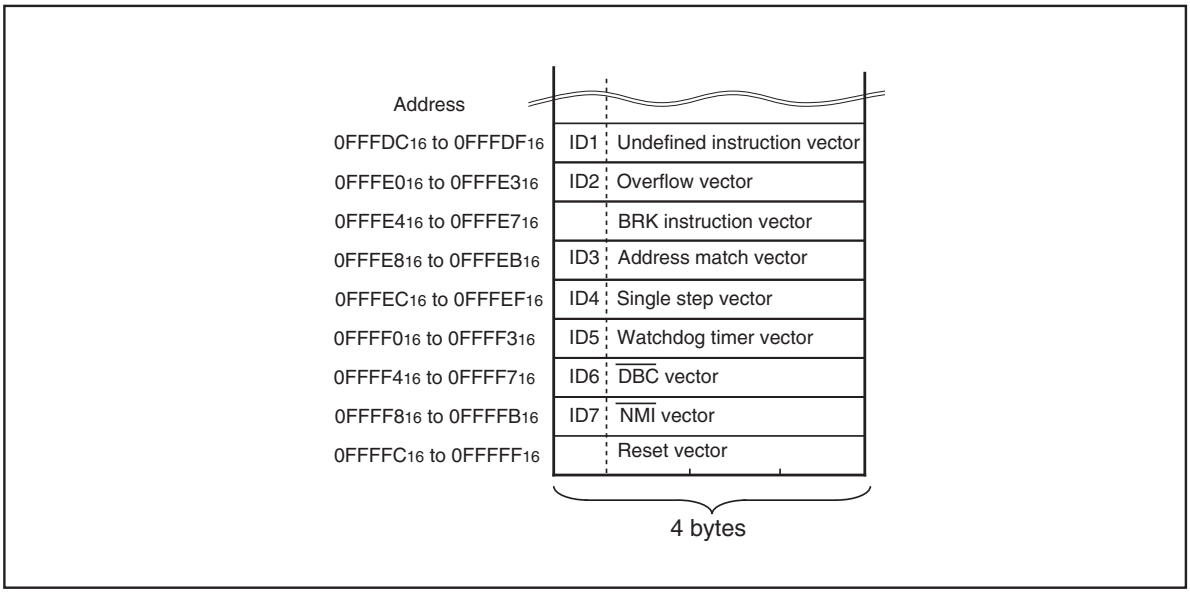


Figure 1.21.2. ID code store addresses

Appendix Standard Serial I/O Mode (Flash Memory Version)

Table 1.22.1. Pin functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply program/erase protection voltage to Vcc pin and 0 V to Vss pin.
CNVSS	CNVSS	I	Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
AVCC, AVSS	Analog power supply input	I	Connect AVss to Vss and AVcc to Vcc, respectively
IVCC	Power supply	—	This pin must be left unconnected.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P10 to P17	Input port P1	I	Input "H" or "L" level signal or open
P60 to P63	Input port P6	I	Input "H" or "L" level signal or open.
P64	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitors the boot program operation check signal output pin.
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
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P83, P86, P87	Input port P8	I	Input "H" or "L" level signal or open.
P86	$\overline{\text{CE}}$ input	I	Input "H" level signal.
P90 to P97	Input port P9	I	Input "H" or "L" level signal or open.
P100 to P107	Input port P10	I	Input "H" or "L" level signal or open.

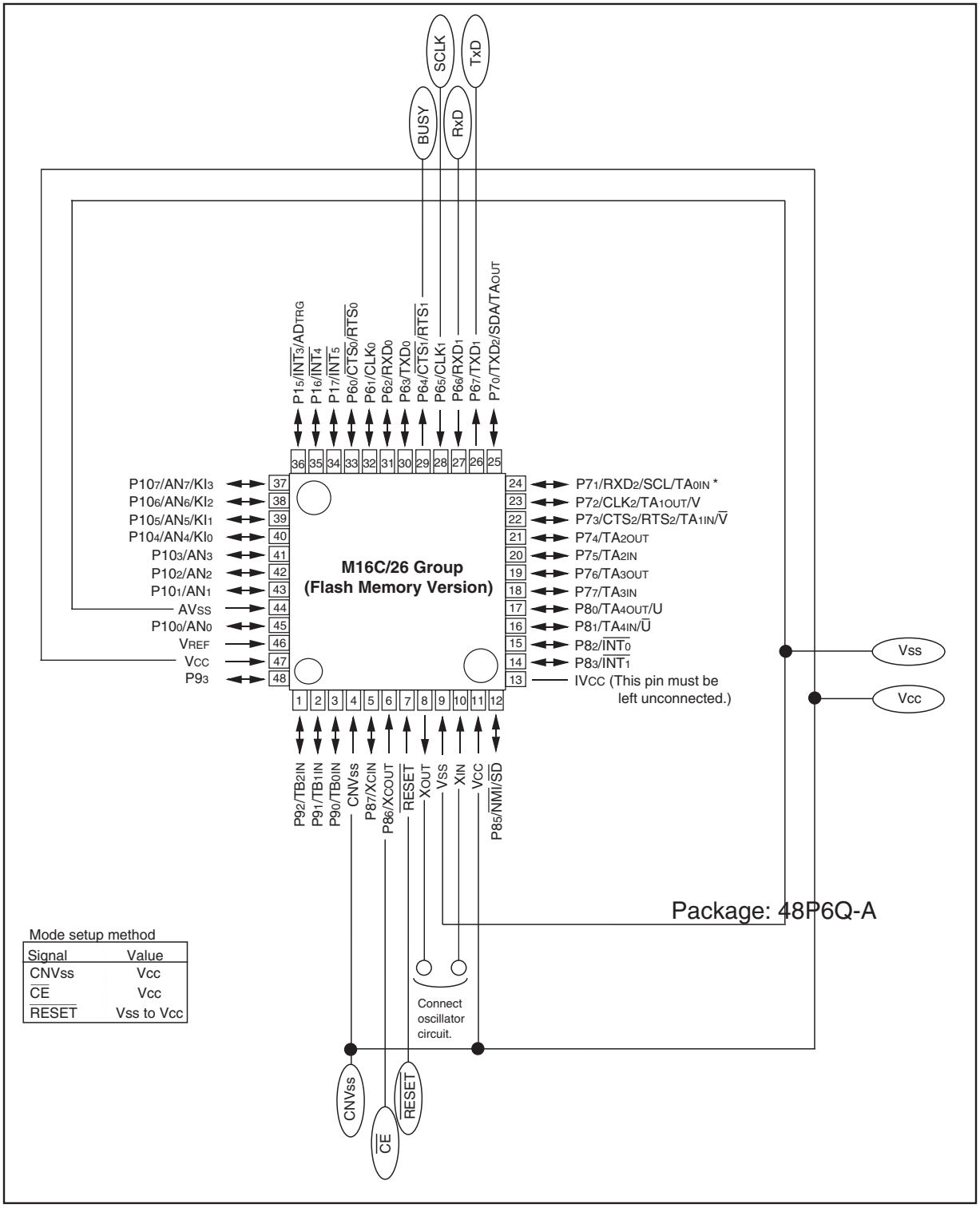


Figure 1.22.1. Pin connections for serial I/O mode (1)

Standard Serial I/O mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory using the serial I/O port UART1. The serial I/O mode transfers the data serially in 8-bit units.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU executes a control program for flash memory rewrite (using the CPU's rewrite mode), rewrite data input and so forth. It is started when both the $P8_6$(CE) pin and the CNVss pin are in “H” level after the reset is released. (In normal operation mode, set CNVss pin to “L” level.)

Figure 1.22.1 shows the pin connections for the standard serial I/O mode.

There are actually two standard serial I/O modes: mode 1, which is clock synchronized, and mode 2, which is asynchronized. Standard serial I/O switches between mode 1 (clock synchronous) and mode 2 (clock asynchronous) depending on the level of CLK_1 pin when the reset is released.

To use standard serial I/O mode 1 (clock synchronous), set the CLK_1 pin to “H” level and release the reset. The operation uses the four UART1 pins CLK_1, RxD_1, TxD_1 and RTS_1 (BUSY). The CLK_1 pin is the transfer clock input pin through which an external transfer clock is input. The TxD_1 pin is for CMOS output. The RTS_1 (BUSY) pin outputs an “L” level when ready for reception and an "H" level when reception starts. In mode 1, be sure the TxD_1 ($P6_7$) pin is at high before reset being deasserted.

To use standard serial I/O mode 2 (clock asynchronous), set the CLK_1 pin to “L” level and release the reset. The operation uses the two UART1 pins RxD_1 and TxD_1.

In the standard serial I/O mode, only the user ROM area indicated in Figure 1.19.1 can be rewritten. In addition, a 7-byte ID code exists to protect the device. When there is data in the flash memory, commands sent from the peripheral unit are not accepted unless the ID code matches.

Appendix Standard Serial I/O Mode 1 (Flash Memory Version)

Overview of standard serial I/O mode 1 (clock synchronous)

In standard serial I/O mode 1, software commands, addresses and data are input and output between the microcomputer and peripheral units (serial programer, etc.) using 4-wire clock-synchronized serial I/O (UART1). Standard serial I/O mode 1 is entered by releasing the reset with the $P6_5$ (CLK_1) pin "H" level (and $P8_6$ (CE) pin and the CNVss pin are in "H" level).

When receiving software commands, addresses and program data are synchronized with the rising edge of the transfer clock that is input to the CLK_1 pin, and are then input to the RXD_1 pin. When transmitting, read data and status are synchronized with the falling edge of the transfer clock, and output from the TxD_1 pin. The data transfer is in 8-bit units with LSB first.

The TxD_1 pin is a CMOS level output.

When the device is busy, such as during transmission, reception, erasing or program execution, the RTS_1 (BUSY) pin is "H" level. Accordingly, always start the next transfer after the RTS_1 (BUSY) pin is "L" level.

Also, data and status registers in memory can be read after inputting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. The following table shows the software commands, status registers, etc. in serial I/O mode 1.

Software Commands

Table 1.22.2 lists the software commands, and their descriptions, in standard serial I/O mode 1. Erase operations, programming, reading status, etc. are controlled by transferring software commands via the RxD_1 pin.

Table 1.22.2. Software commands (Standard serial I/O mode 1) (Note 1)

	Control Command	1st Byte Transfer	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	... nth Byte	When ID is not verified
1	Page read (Note 2)	FF_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41_{16}	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	20_{16}	Address (middle)	Address (high)	$D0_{16}$				Not acceptable
4	Erase all code blocks (Note 4)	$A7_{16}$	$D0_{16}$						Not acceptable
5	Read status register (Notes 2,3)	70_{16}	SRD output	SRD1 output					Acceptable
6	Clear status register	50_{16}							Not acceptable
7	ID check function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
8	Download function	FA_{16}	Size (low)	Size (high)	Checksum	Data input	To required no. of times		Not acceptable
9	Version data output function (Note 2)	FB_{16}	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
10	Read check data (Note 2)	FD_{16}	Check data (low)	Check data (high)					Not acceptable

Note 1: All commands can be accepted when the flash memory is totally blank.
Note 2: Shaded area indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.
Note 3: SRD refers to status register data. SRD1 refers to status register 1 data.
Note 4: The 'erase all' command does not erase the Flash data blocks.

Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

(1) Transfer the "FF_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first in sync with the fall of the clock.

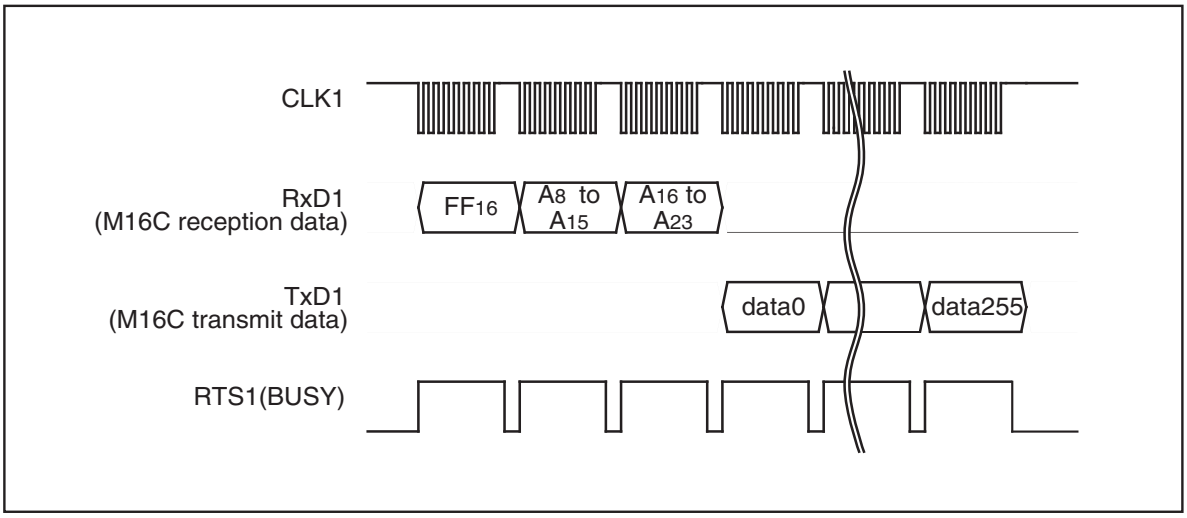


Figure 1.22.2. Timing for page read

Read Status Register Command

This command reads status information. When the "70_{16}" command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

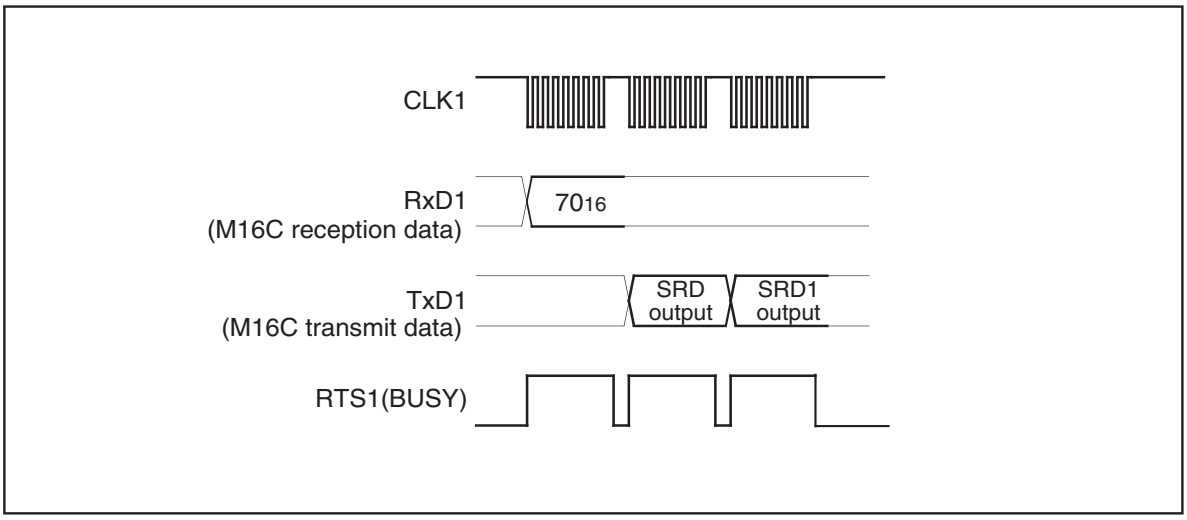


Figure 1.22.3. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR4, SR5) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS_1 (BUSY) signal changes from the "H" to the "L" level.

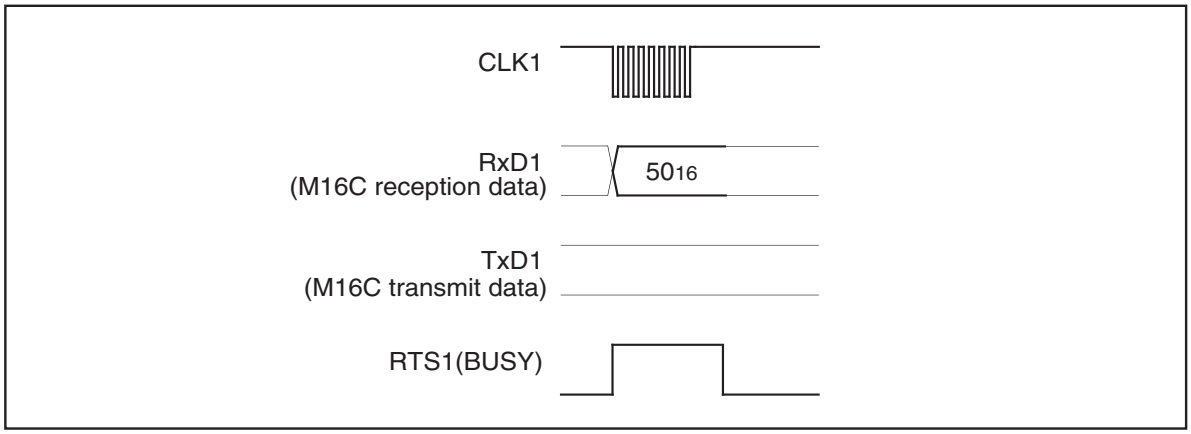


Figure 1.22.4. Timing for clearing the status register

Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

(1) Transfer the "41_{16}" command code with the 1st byte.
(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.
(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS_1 (BUSY) signal changes from the "H" to the "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

Each block can be write-protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.

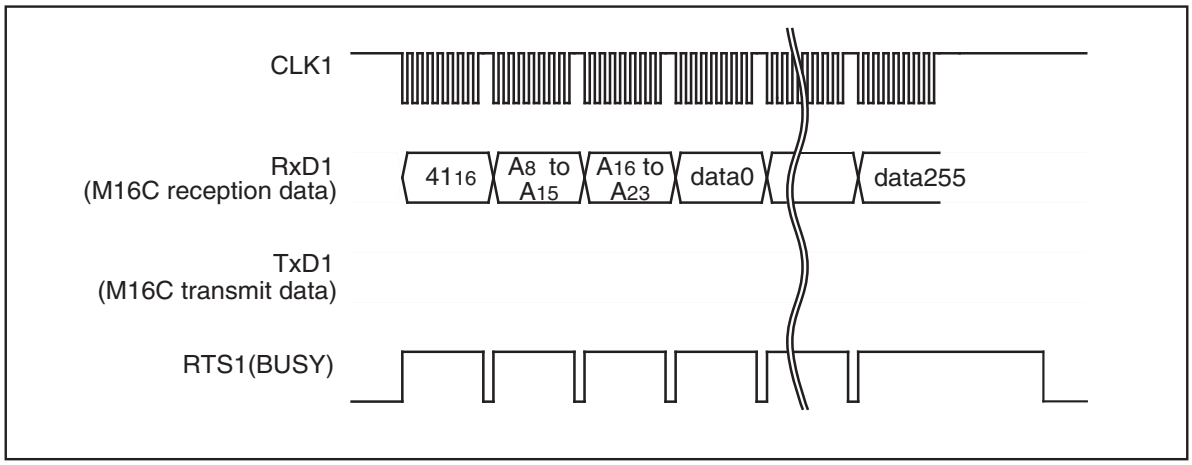


Figure 1.22.5. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

(1) Transfer the "20_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) Transfer the verifycommand code "$D0_{16}$" with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A_8 to A_{23}.

When block erasing ends, the RTS_1 (BUSY) signal changes from the "H" to the "L" level. After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.

Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

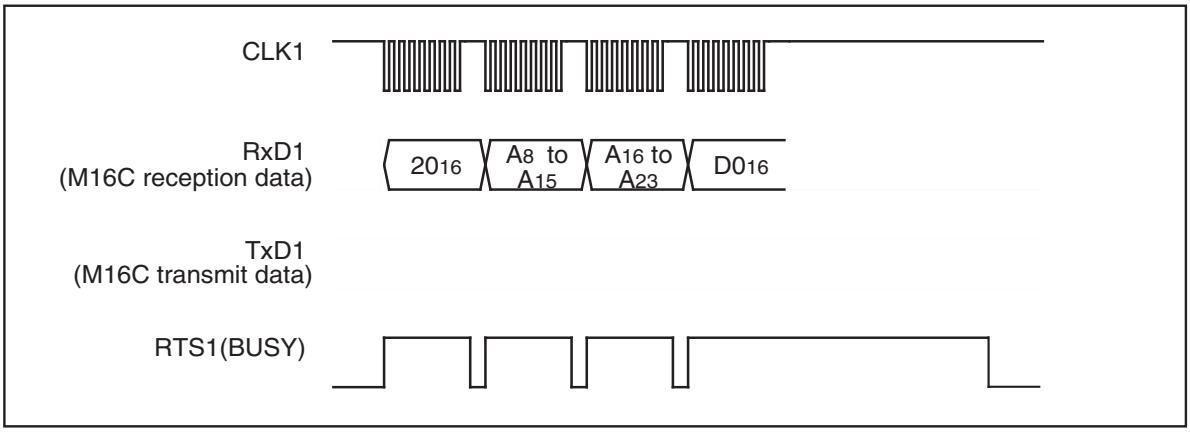


Figure 1.22.6. Timing for block erasing

Erase All Code Blocks Command

This command erases the content of all code blocks. Execute the erase all code blocks command as explained here following.

(1) Transfer the "A716" command code with the 1st byte.

(2) Transfer the verify command code "D016" with the 2nd byte. With the verify command code, the erase operation will start and continue for all code blocks in the flash memory.

When block erasing ends, the RTS1 (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register.

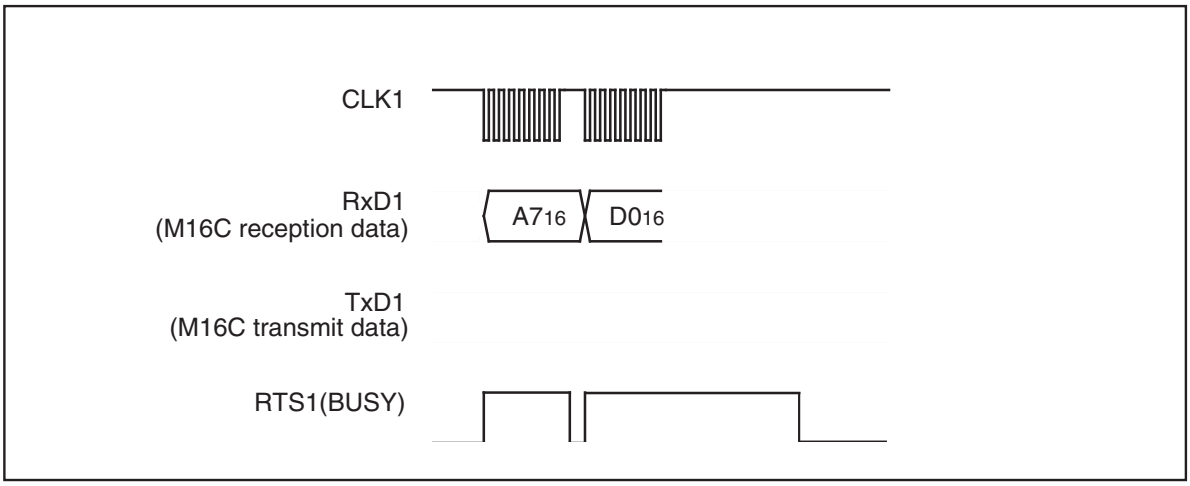


Figure 1.22.7. Timing for erasing all code blocks.

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Transfer the "FA16" command code with the 1st byte.
(2) Transfer the program size with the 2nd and 3rd bytes.
(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.
(4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

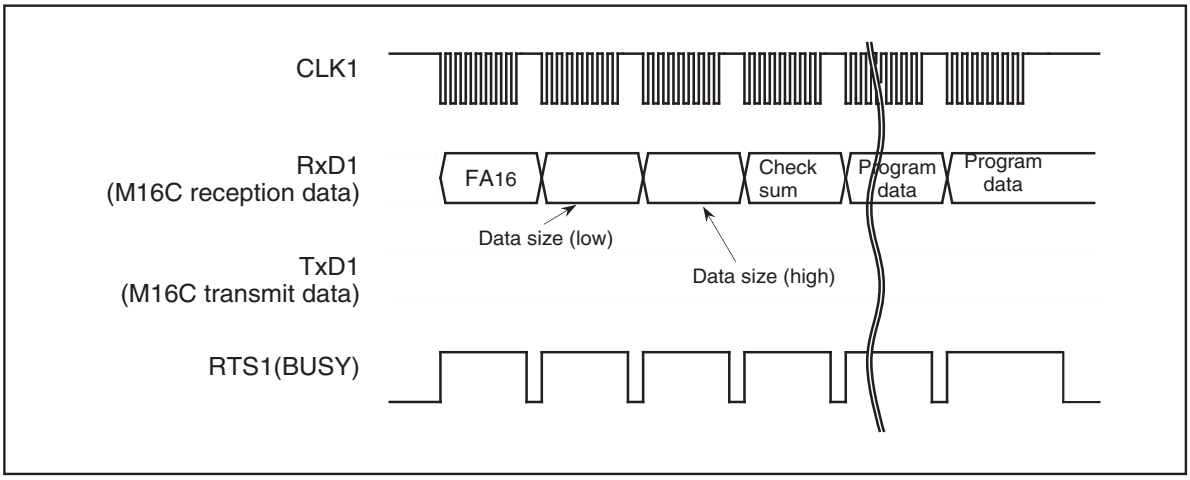


Figure 1.22.8. Timing for download

Version Information Output Command

This command outputs the version information of the control program. Execute the version information output command as explained here following.

(1) Transfer the "FB16" command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

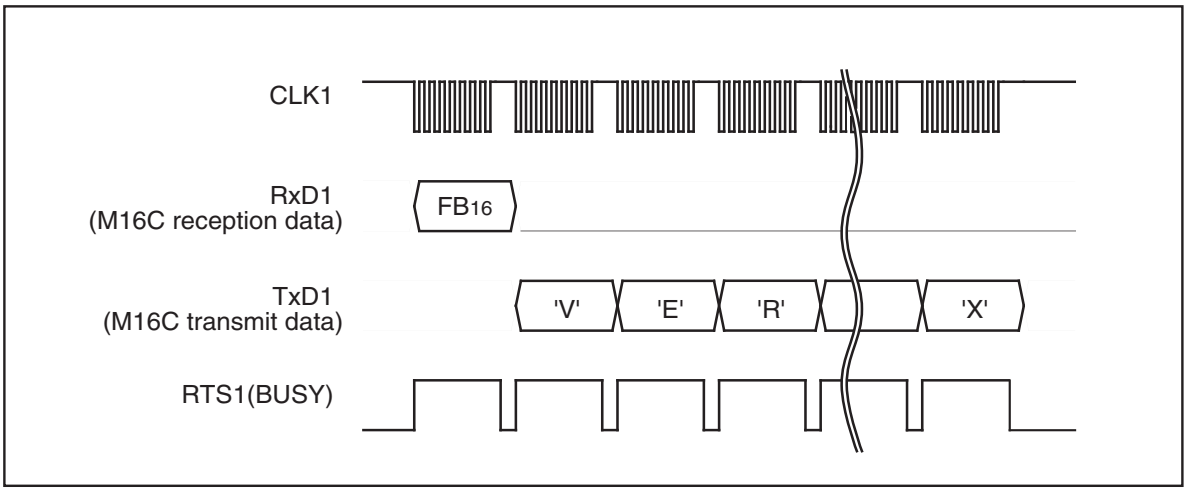


Figure 1.22.9. Timing for version information output

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Transfer the "F5$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

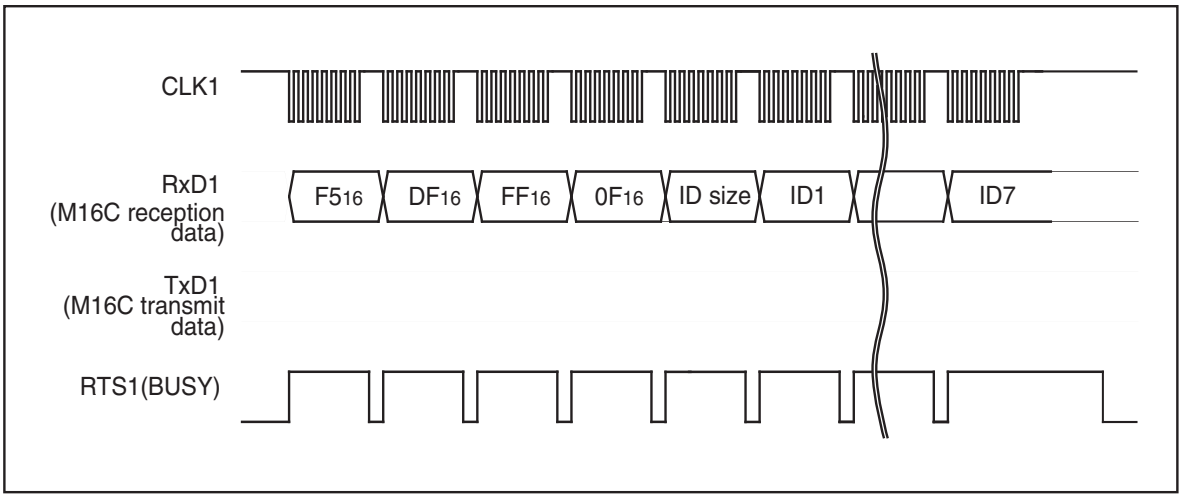


Figure 1.22.10. Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF$_{16}$, 0FFFE3$_{16}$, 0FFFEB$_{16}$, 0FFFEF$_{16}$, 0FFFF3$_{16}$, 0FFFF7$_{16}$ and 0FFFFB$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.

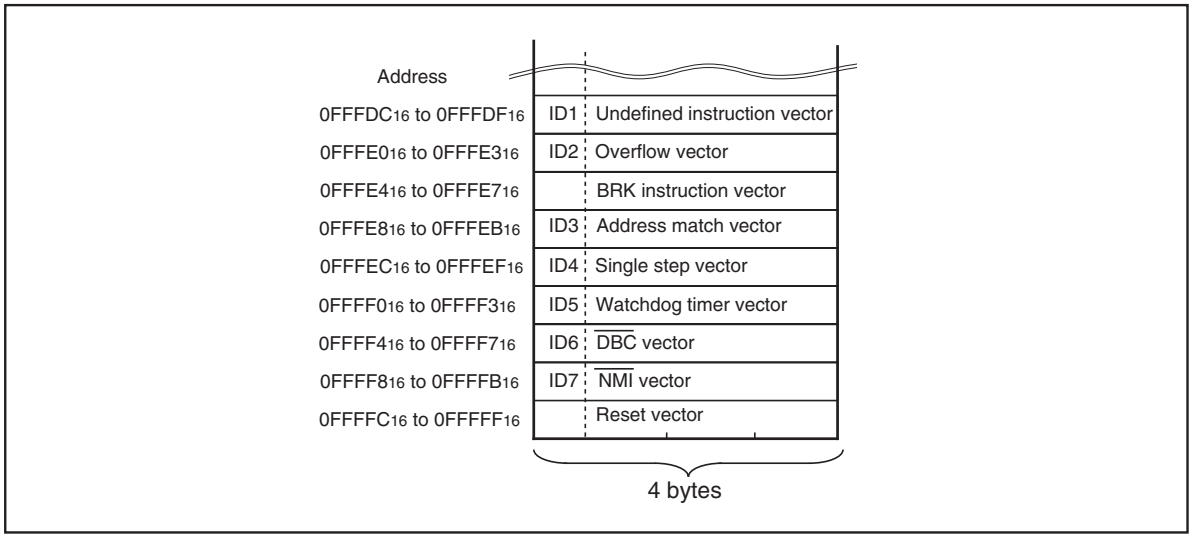


Figure 1.22.11. ID code storage addresses

Read Check Data

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

(1) Transfer the "FD16" command code with the 1st byte.

(2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

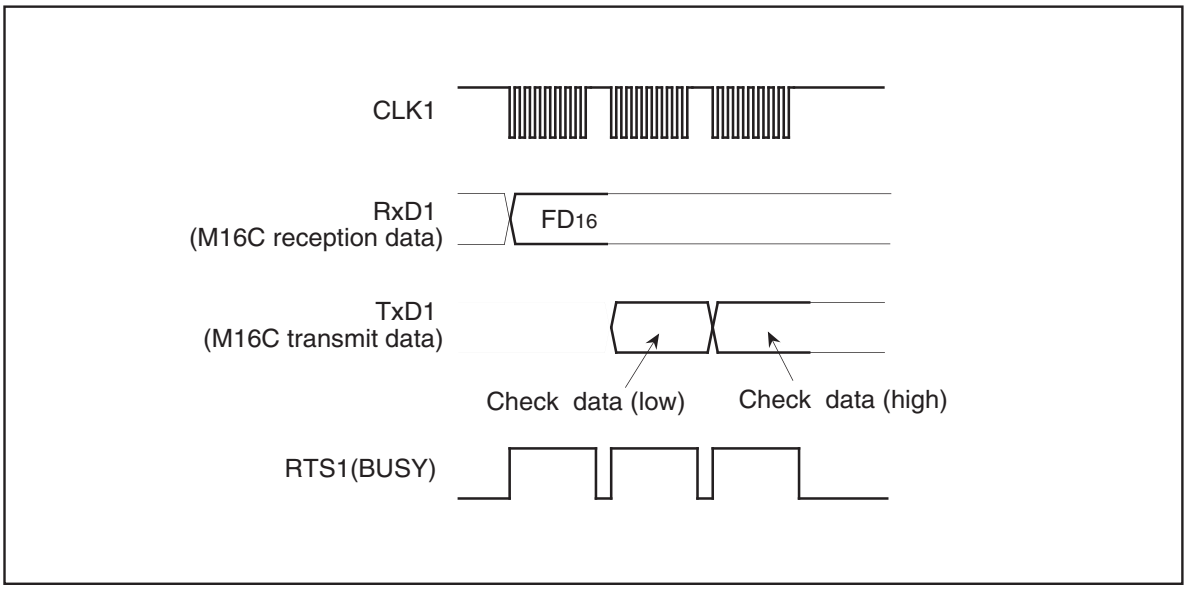


Figure 1.22.12. Timing for the read check data

Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70_{16}). Also, the status register is cleared by writing the clear status register command (50_{16}). Table 1.22.3 gives the definition of each status register bit. After clearing the reset, the status register outputs "80_{16}".

Table 1.22.3. Status register (SRD)

SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Sequencer status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Sequencer status (SR7)

After power-on, the sequencer status is set to 1(ready).

The sequencer status indicates the operating status of the device. This status bit is set to "0" (busy) during write or erase operation and is set to 1 upon completion of these operations.

Erase Status (SR5)

The erase status reports the operating status of the auto erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is set to "0".

Program Status (SR4)

The program status reports the operating status of the auto write operation. If a write error occurs, it is set to "1". When the program status is cleared, it is set to "0".

Status Register 1 (SRD1)

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (70_{16}). Also, status register 1 is cleared by writing the clear status register command (50_{16}).

Table 1.22.4 gives the definition of each status register 1 bit. "00_{16}" is output when power is turned ON and the flag status is maintained even after the reset.

Table 1.22.4. Status register 1 (SRD1)

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Flash identification value	HND	DINOR
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Check sum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 01 10 11	Not verified Verification mismatch Reserved Verified
SR9 (bit1)	Data receive time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-

Boot Update Completed Bit (SR15)

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

Flash Identification Value (SR14)

This flag indicates whether the flash memor type is HND or DINOR.

Check Sum Match Bit (SR12)

This flag indicates whether the check sum matches or not when a program, is downloaded for execution using the download function.

ID Check Completed Bits (SR11 and SR10)

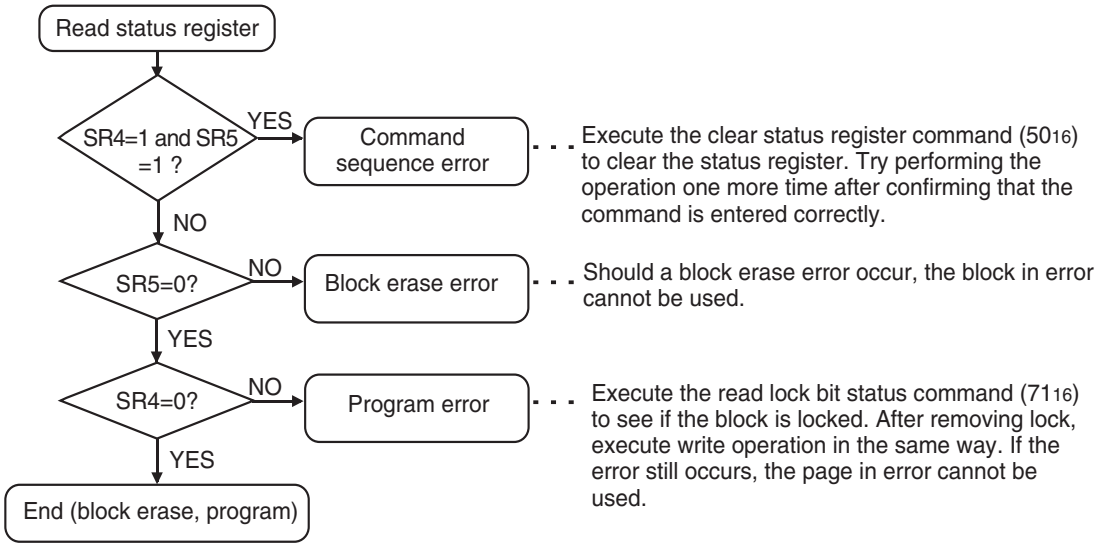
These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Receive Time Out (SR9)

This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the microcomputer returns to the command wait state.

Full Status Check

Results from executed erase and program operations can be known by running a full status check. Figure 1.22.13 shows a flowchart of the full status check and explains how to remedy errors which occur.

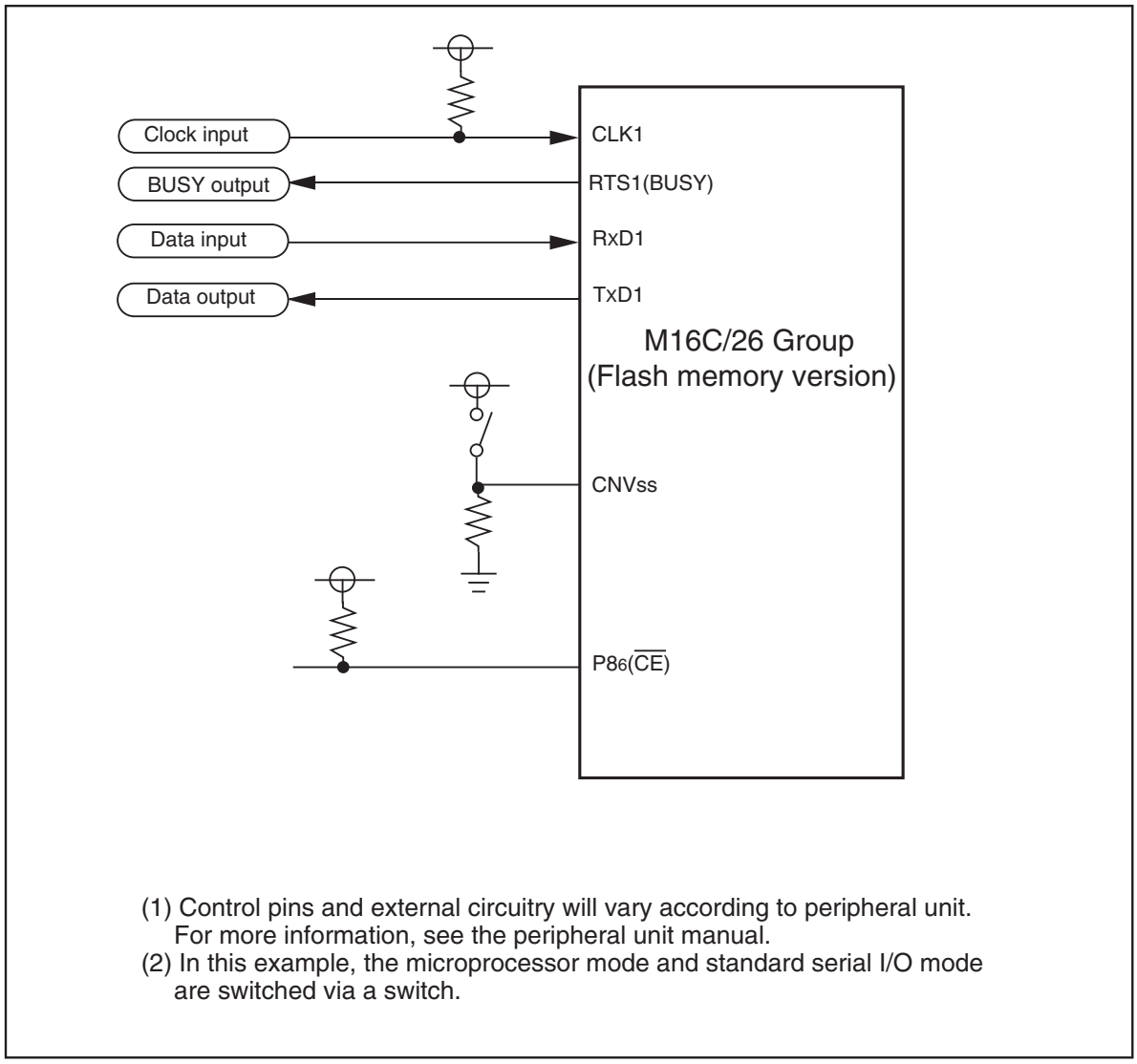


Note: When one of SR5 to SR4 is set to 1, none of the program, erase all blocks, and block erase commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Figure 1.22.13. Full status check flowchart and remedial procedure for errors

Example Circuit Application for The Standard Serial I/O Mode 1

The below figure shows a circuit application for the standard serial I/O mode 1. Control pins will vary according to programmer, therefore see the peripheral unit manual for more information.



(1) Control pins and external circuitry will vary according to peripheral unit. For more information, see the peripheral unit manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure 1.22.14. Example circuit application for the standard serial I/O mode 1

Appendix Standard Serial I/O Mode 2 (Flash Memory Version)

Overview of standard serial I/O mode 2 (clock asynchronized)

In standard serial I/O mode 2, software commands, addresses and data are input and output between the microcomputer and peripheral units (serial programer, etc.) using 2-wire clock-asynchronized serial I/O (UART1). Standard serial I/O mode 2 is engaged by releasing the reset with the $P6_5$ (CLK_1) pin "L" level and $P8_6$ (CE) pin and the CNVss pin are in "H" level).

The TxD_1 pin is for CMOS output. Data transfer is in 8-bit units with LSB first, 1 stop bit and parity OFF.

After the reset is released, connections can be established at 9,600 bps when initial communications (Figure 1.23.1) are made with a peripheral unit that requires a main clock with a minimum 2 MHz input oscillation frequency. Baud rate can also be changed from 9,600 bps to 19,200, 38,400 or 57,600 bps by executing software commands. However, if communication errors due to the oscillation frequency of the main clock, change the main clock's oscillation frequency and the baud rate.

After executing commands from a peripheral unit that requires time, i.e. erase, or write (program) data, allow sufficient time to pass or execute the read status command to check the device status, before executing the next command.

Data and status registers in memory can be read after transmitting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. The following describes the initial communications with peripheral units, how frequency is identified, and software commands.

Initial communications with peripheral units

After the reset is released, the bit rate generator is adjusted to 9,600 bps to match the oscillation frequency of the main clock, by sending the code as prescribed by the protocol for initial communications with peripheral units (Figure 1.23.1).

(1) Transmit "$B0_{16}$" from a peripheral unit. If the oscillation frequency input by the main clock is 10 or 16 MHz, the microcomputer with internal flash memory outputs the "$B0_{16}$" check code. If the oscillation frequency is anything other than 10 or 16 MHz, the microcomputer does not output anything.

(2) Transmit "00_{16}" from a peripheral unit 16 times. (The microcomputer with internal flash memory sets the bit rate generator so that "00_{16}" can be successfully received.)

(3) The microcomputer with internal flash memory outputs the "$B0_{16}$" check code and initial communications end successfully *1. Initial communications must be transmitted at a speed of 9,600 bps and a transfer interval of a minimum 15 ms. Also, the baud rate at the end of initial communications is 9,600 bps.

*1. If the peripheral unit cannot receive "$B0_{16}$" successfully, change the oscillation frequency of the main clock.

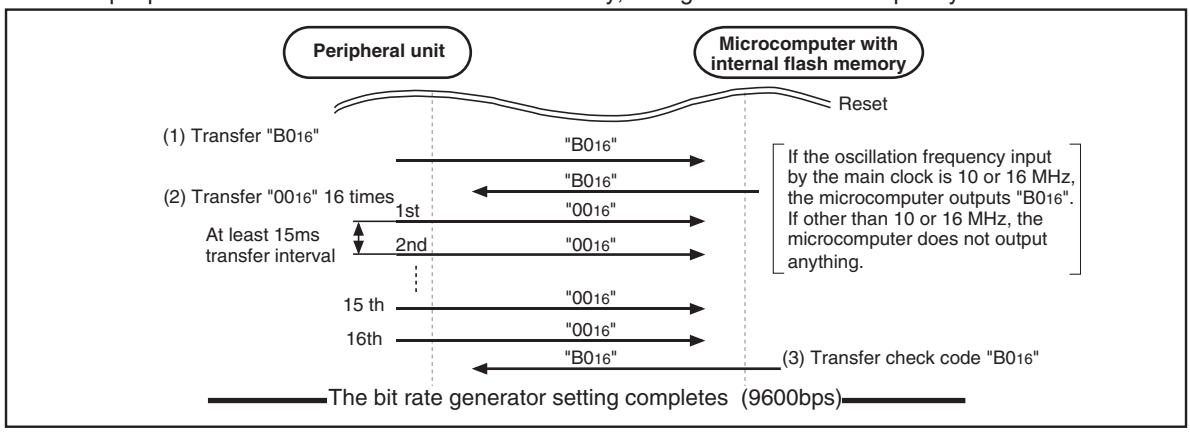


Figure 1.23.1. Peripheral unit and initial communication

How frequency is identified

When "00$_{16}$" data is received 16 times from a peripheral unit at a baud rate of 9,600 bps, the value of the bit rate generator is set to match the operating frequency (2 - 20 MHz). The highest speed is taken from the first 8 transmissions and the lowest from the last 8. These values are then used to calculate the bit rate generator value for a baud rate of 9,600 bps.

Baud rate cannot be attained with some operating frequencies. Table 1.23.1 gives the operation frequency and the baud rate that can be attained for.

Table 1.23.1 Operation frequency and the baud rate

Operation frequency (MHz)	Baud rate 9,600bps	Baud rate 19,200bps	Baud rate 38,400bps	Baud rate 57,600bps
20MHz	☑	☑	☑	☑
16MHz	☑	☑	☑	☑
12MHz	☑	☑	☑	–
11MHz	☑	☑	☑	–
10MHz	☑	☑	–	☑
8MHz	☑	☑	–	☑
7.3728MHz	☑	☑	☑	
6MHz	☑	☑	☑	–
5MHz	☑	☑	–	–
4.5MHz	☑	☑	–	☑
4.194304MHz	☑	☑	☑	–
4MHz	☑	☑	–	–
3.58MHz	☑	☑	☑	☑
3MHz	☑	☑	☑	–
2MHz	☑	–	–	–

☑: Communications possible
– : Communications not possible

Software Commands

Table 1.23.2 lists software commands. In the standard serial I/O mode 2, erase operations, programs and reading are controlled by transferring software commands via the RxD1 pin. Standard serial I/O mode 2 adds four transmission speed commands - 9,600, 19,200, 38,400 and 57,600 bps - to the software commands of standard serial I/O mode 1. Software commands are explained here below.

Table 1.23.2. Software commands (Standard serial I/O mode 2) (Note 1)

	Control Command	1st Byte Transfer	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	... nth Byte	When ID is not verified
1	Page read (Note 2)	FF16	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	4116	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	2016	Address (middle)	Address (high)	D016				Not acceptable
4	Erase all code blocks (Note 4)	A716	D016						Not acceptable
5	Read status register (Notes 2,3)	7016	SRD output	SRD1 output					Acceptable
6	Clear status register	5016							Not acceptable
7	ID check function	F516	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
8	Download function	FA16	Size (low)	Size (high)	Checksum	Data input	To required no. of times		Not acceptable
9	Version data output function (Note 2)	FB16	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
10	Read check data (Note 2)	FD16	Check data (low)	Check data (high)					Not acceptable
11	Baud rate 9600 (Note 2)	B016	B016						Acceptable
12	Baud rate 19200 (Note 2)	B116	B116						Acceptable
13	Baud rate 38400 (Note 2)	B216	B216						Acceptable
14	Baud rate 57600 (Note 2)	B316	B316						Acceptable

Note 1: All commands can be accepted when the flash memory is totally blank.
Note 2: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.
Note 3: SRD refers to status register data. SRD1 refers to status register 1 data.
Note 4: The 'erase all' command does not erase the Flash data blocks.

Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

(1) Transfer the "FF_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first.

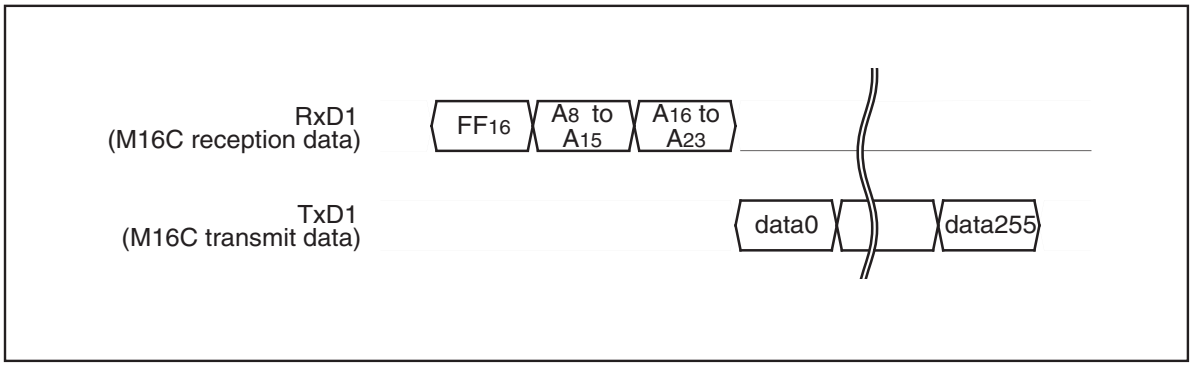


Figure 1.23.2. Timing for page read

Read Status Register Command

This command reads status information. When the "70_{16}" command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

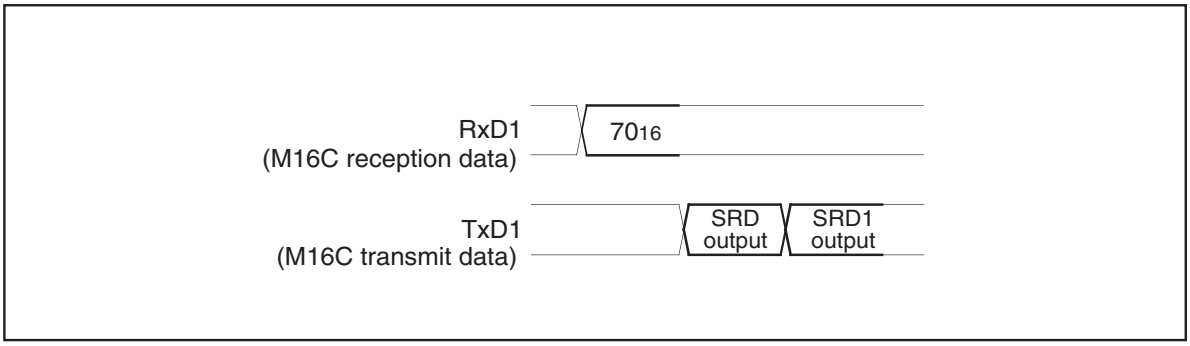


Figure 1.23.3. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR4, SR5) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared.

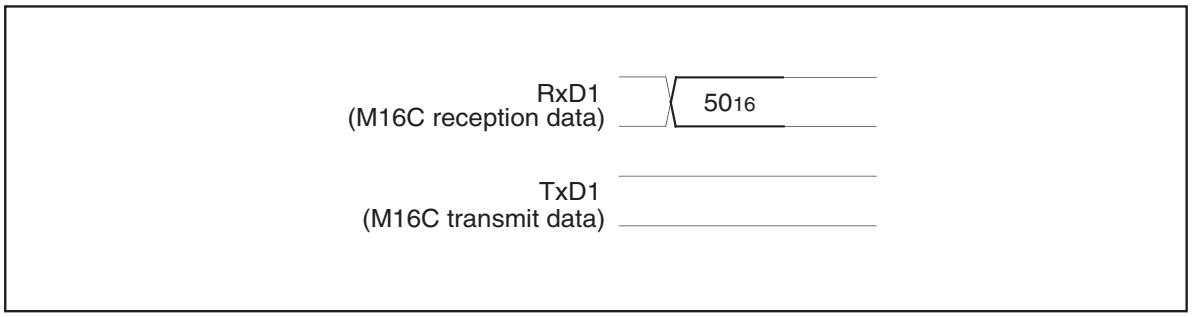


Figure 1.23.4. Timing for clearing the status register

Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

(1) Transfer the "41_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

The result of the page program can be known by reading the status register. For more information, see the section on the status register.

Each block can be write-protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.

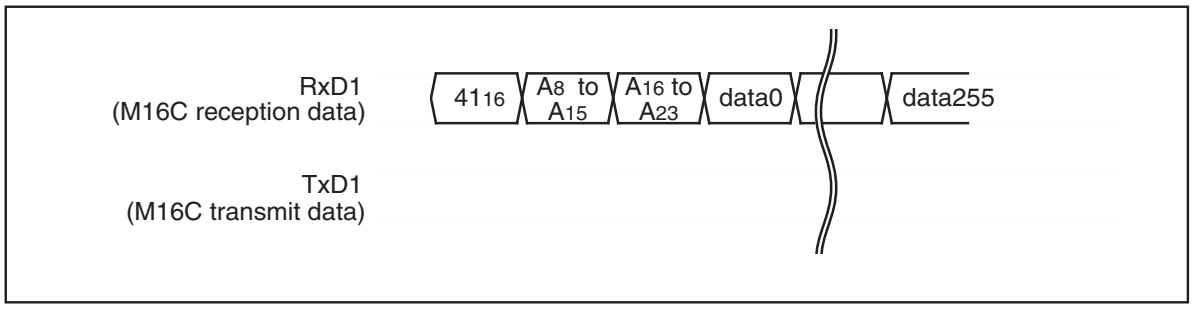


Figure 1.23.5. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

(1) Transfer the "20$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ with the 2nd and 3rd bytes respectively.

(3) Transfer the verify command code "D0$_{16}$" with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A$_8$ to A$_{23}$.

After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.

Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

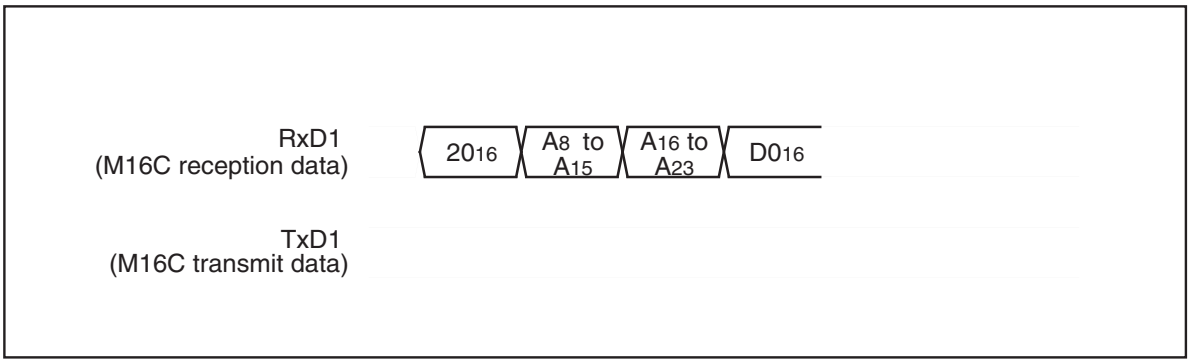


Figure 1.23.6. Timing for block erasing

Erase All Code Blocks Command

This command erases the content of all code blocks. Execute the erase all code blocks command as explained here following.

(1) Transfer the "A716" command code with the 1st byte.

(2) Transfer the verify command code "D016" with the 2nd byte. With the verify command code, the erase operation will start and continue for all code blocks in the flash memory.

The result of the erase operation can be known by reading the status register.

.

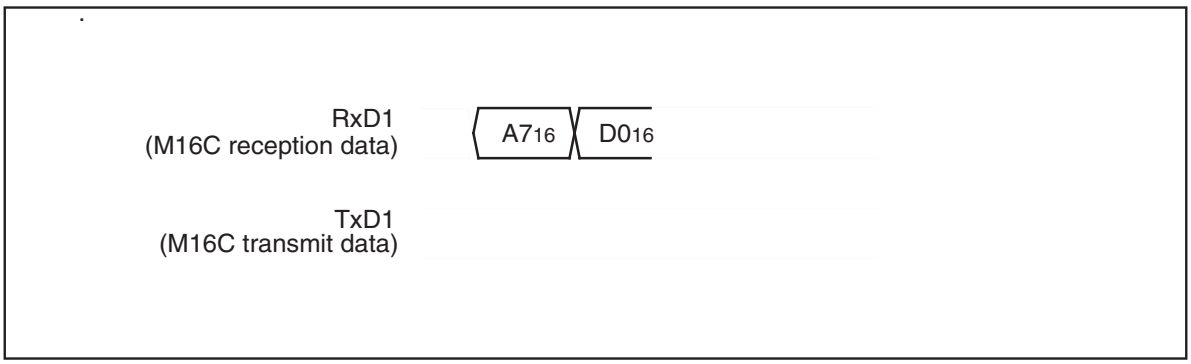


Figure 1.23.7. Timing for erasing all code blocks.

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Transfer the "FA16" command code with the 1st byte.
(2) Transfer the program size with the 2nd and 3rd bytes.
(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.
(4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

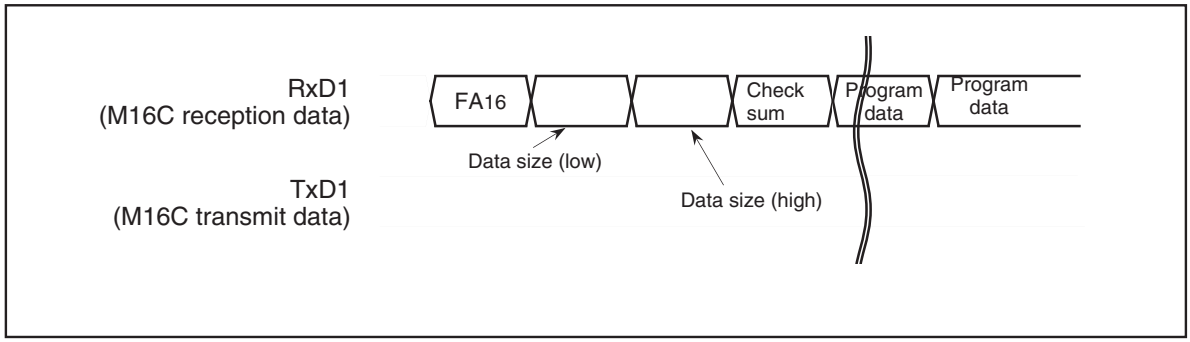


Figure 1.23.8. Timing for download

Version Information Output Command

This command outputs the version information of the control program. Execute the version information output command as explained here following.

(1) Transfer the "FB16" command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

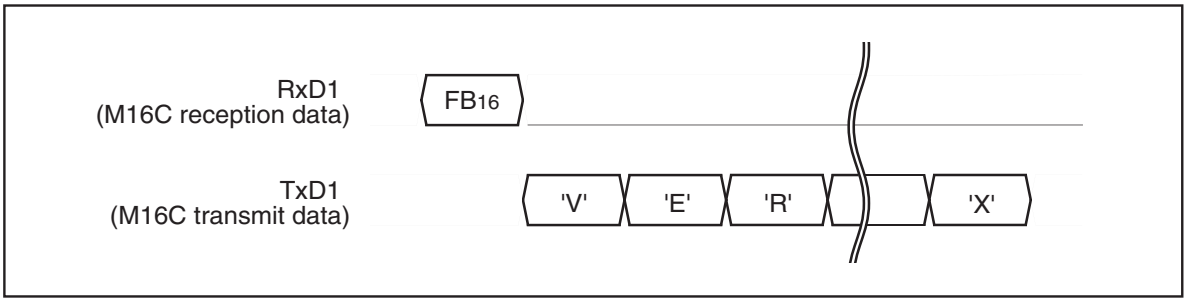


Figure 1.23.9. Timing for version information output

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Transfer the "F5$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

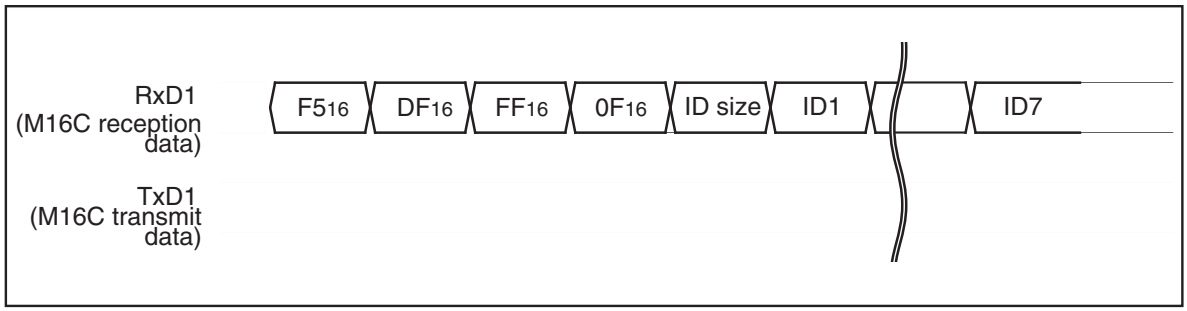


Figure 1.23.10. Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF$_{16}$, 0FFFE3$_{16}$, 0FFFEB$_{16}$, 0FFFEF$_{16}$, 0FFFF3$_{16}$, 0FFFF7$_{16}$ and 0FFFFB$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.

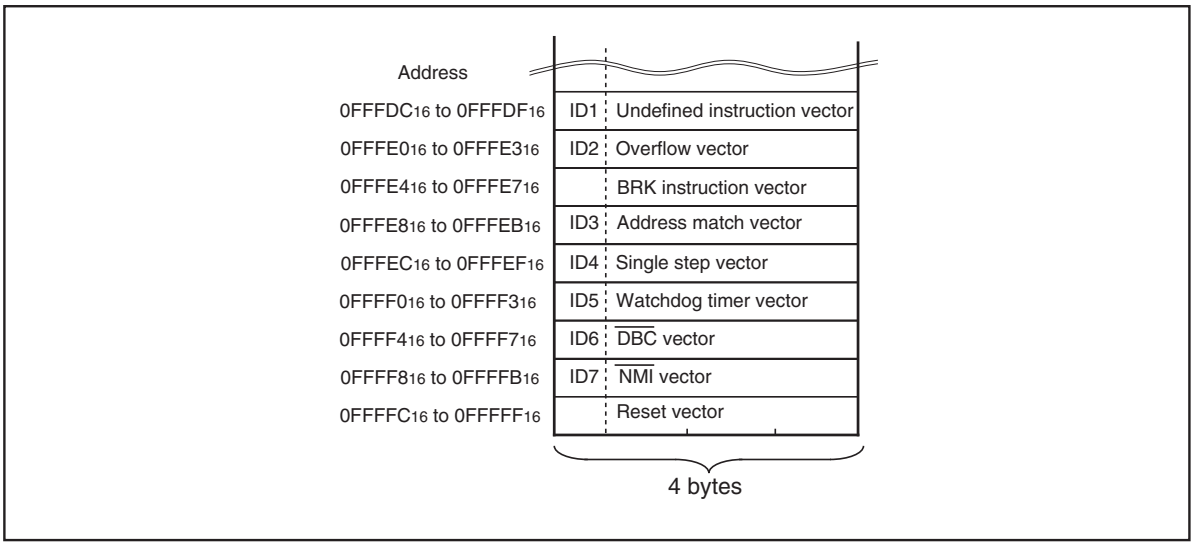


Figure 1.23.11. ID code storage addresses

Read Check Data

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

(1) Transfer the "FD16" command code with the 1st byte.

(2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

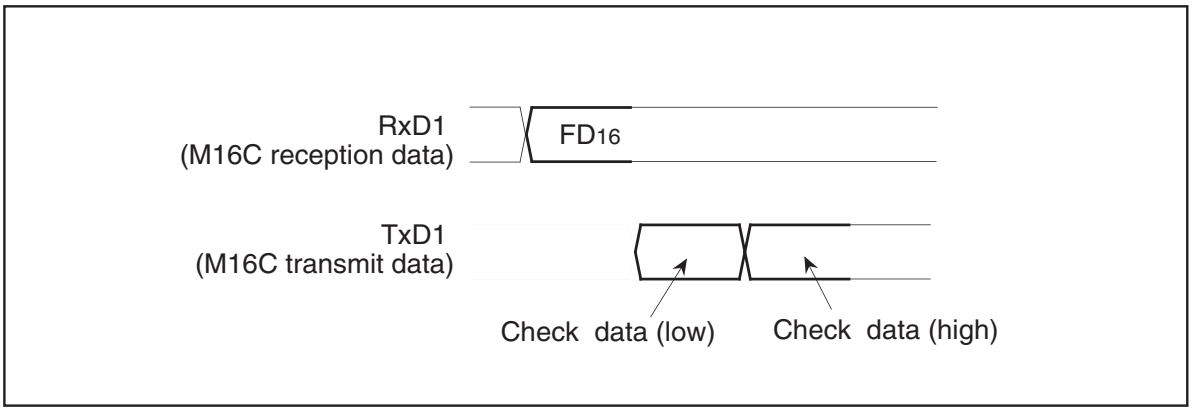


Figure 1.23.12. Timing for the read check data

Baud Rate 9600

This command changes baud rate to 9,600 bps. Execute it as follows.

(1) Transfer the "B016" command code with the 1st byte.

(2) After the "B016" check code is output with the 2nd byte, change the baud rate to 9,600 bps.

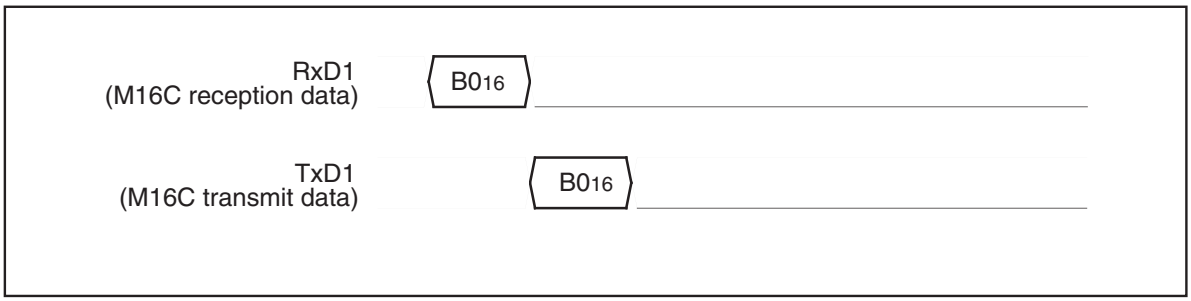


Figure 1.23.13. Timing of baud rate 9600

Baud Rate 19200

This command changes baud rate to 19,200 bps. Execute it as follows.

(1) Transfer the "B1$_{16}$" command code with the 1st byte.

(2) After the "B1$_{16}$" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

Figure 1.23.14. Timing of baud rate 19200

Baud Rate 38400

This command changes baud rate to 38,400 bps. Execute it as follows.

(1) Transfer the "B2$_{16}$" command code with the 1st byte.

(2) After the "B2$_{16}$" check code is output with the 2nd byte, change the baud rate to 38,400 bps.

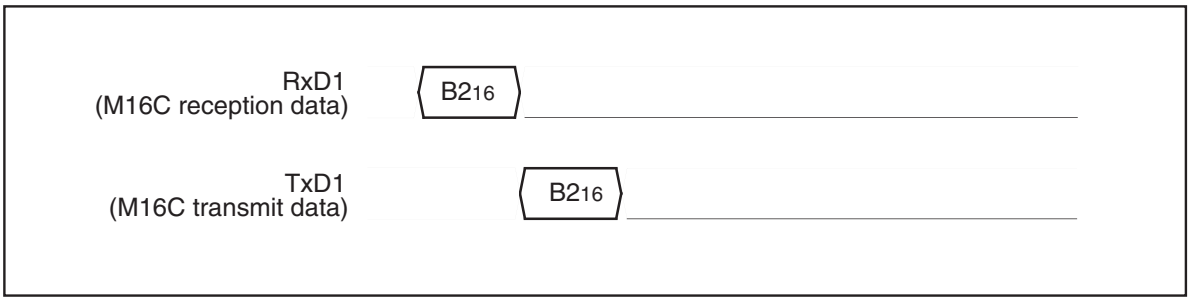


Figure 1.23.15. Timing of baud rate 38400

Baud Rate 57600

This command changes baud rate to 57,600 bps. Execute it as follows.

(1) Transfer the "B3$_{16}$" command code with the 1st byte.

(2) After the "B3$_{16}$" check code is output with the 2nd byte, change the baud rate to 57,600 bps.

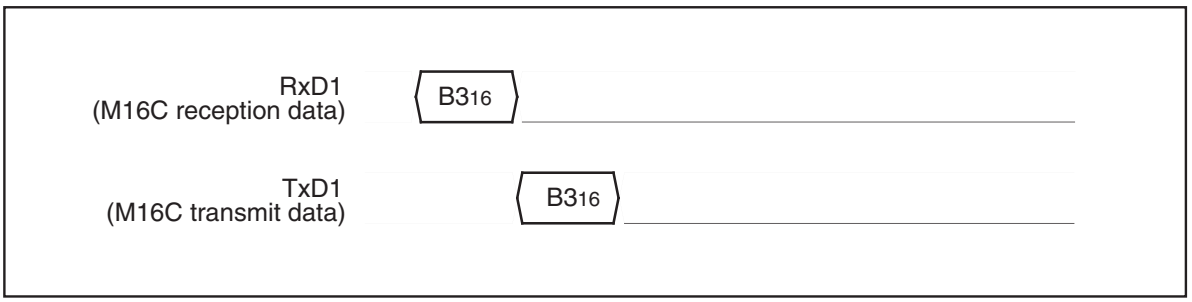
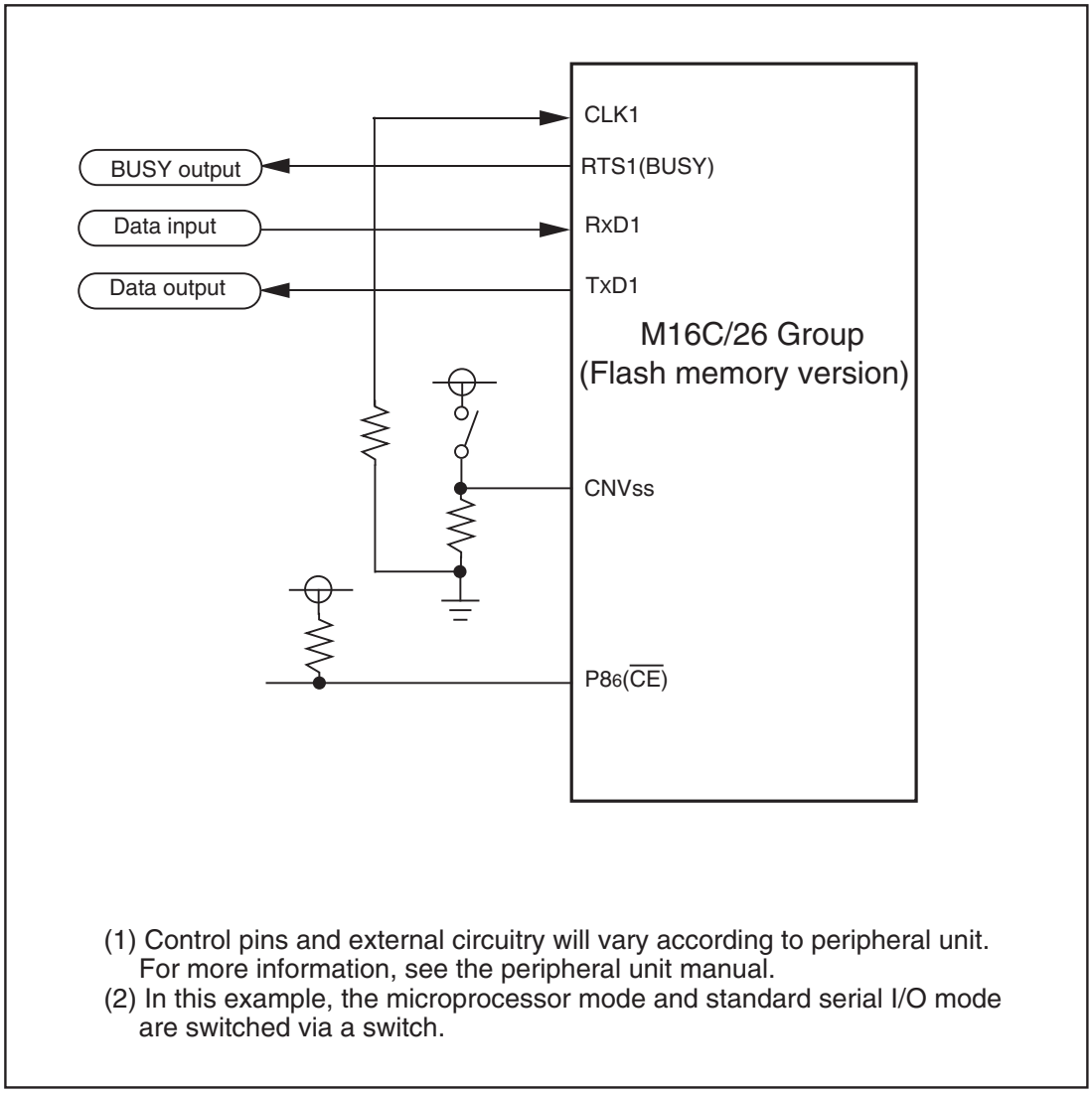


Figure 1.23.16. Timing of baud rate 57600

Example Circuit Application for The Standard Serial I/O Mode 2

The below figure shows a circuit application for the standard serial I/O mode 2.



(1) Control pins and external circuitry will vary according to peripheral unit. For more information, see the peripheral unit manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

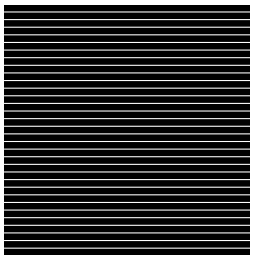
Figure 1.23.17. Example circuit application for the standard serial I/O mode 2

RENESAS 16-BIT CMOS SINGLE-CHIP MICROCOMPUTER
HARDWARE MANUAL
M16C/26 Group Rev.0.90

Editioned by
Committee of editing of RENESAS Semiconductor Hardware Manual

---

M16C/26 Group
Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan