1.0 Description

The M3024 group is a 16-bit microcomputer based on the M16C family core technology. They are single-chip USB peripheral microcontrollers based on the Universal Serial Bus (USB) Version 1.1 specification. They are packaged in an 80-pin, molded plastic QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency, making them capable of executing instructions at high speed. They also feature a built-in multiplier and DMAC, making them ideal for controlling office, communications, industrial equipment, and other high-speed processing applications.

1.1 Features

• CPU	16-bit (including a hardware multiplier)
• Number of instructions	91
• Shortest instruction execution time	83ns(f(Xin)=12MHz
• USB Features:	Five endpoint pairs (IN/OUT) FIFO Sizes (endpoints 0-4):32,128, 32, 32, 32 Conforms to USB V1.1 Specification
• USB Transceiver	Conforms to USB V1.1 Specification-Internal Vref
• Frequency Multiplier	PLL for 48MHz clock
• Memory capacity (mask device):	ROM (40K) / RAM (3.0 K)
• Memory capacity (OTP device):	EPROM (128K) / RAM (5K)
• Supply Voltage	4.1 to 5.5V (f(Xin)=12MHz)
• Interrupts	21 internal and 4 external interrupt sources, 4 software interrupt sources; 7 levels (including key input interrupt X 16)
• Multifunction timer	5 X 16-bit, w/integrated 20mA PWM outputs
• General purpose timer	3 X 16-bit, internal interrupt only
• UART	3 X 7/8/9 bits; Configurable for synchronous or asynchronous mode
• DMAC	2 channels (trigger: 16 sources)
• A-D Converter	10 bits X 8 channels
• CRC calculation circuit	Industry standard polynomial
• Watchdog timer	15-bit
• Programmable I/O	63 lines
• High current and LED Drivers	5 high current and 8 LED drivers
• Clock-generating circuit	1 built-in circuit including feedback resistor
• Package:	80P6N (0.8 mm pitch)

1.2 Applications

USB peripherals, such as telephones, audio systems, scanners, and digital cameras.

1.3 Pin Configuration

Figure 1 shows the pin configuration (top view).

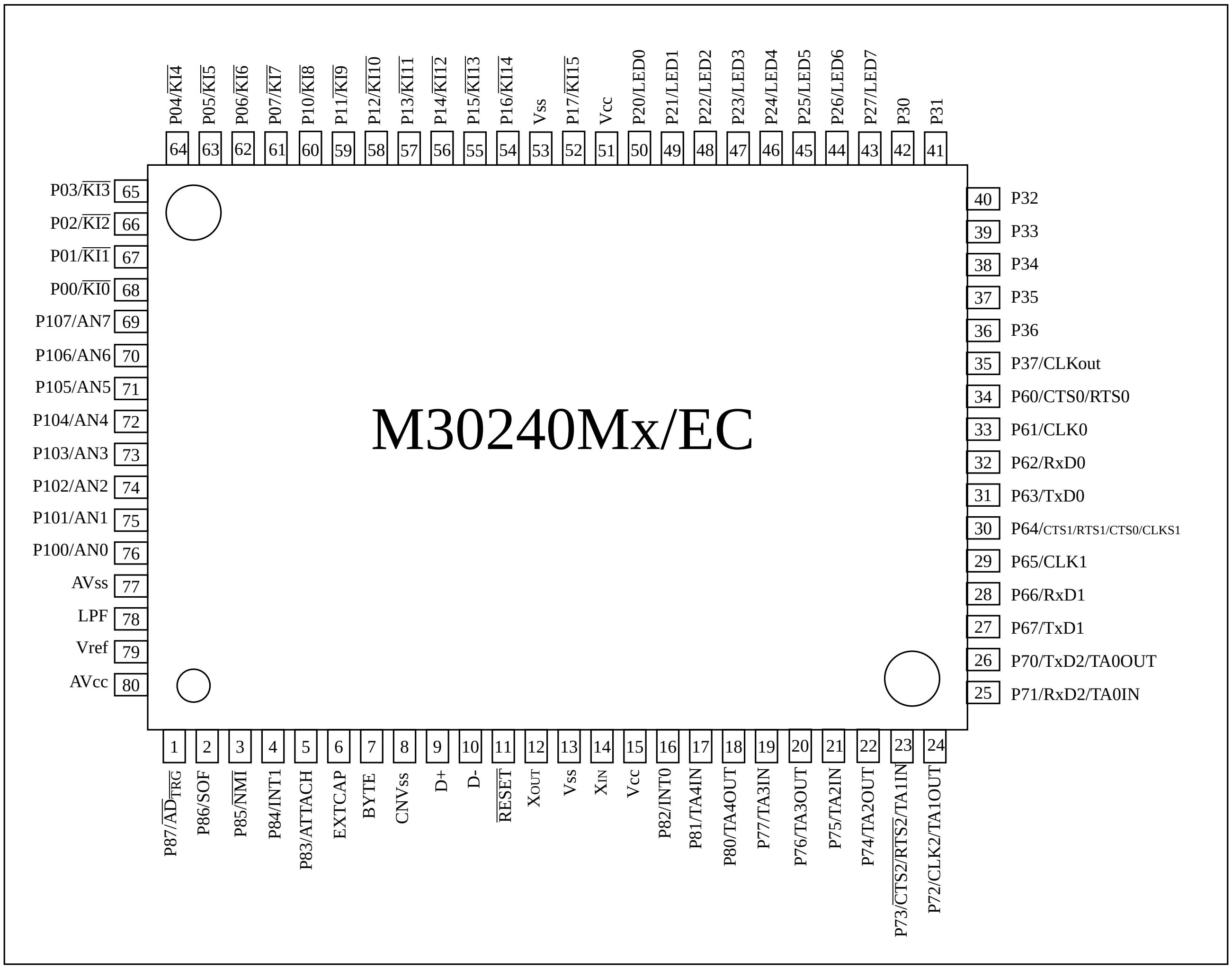
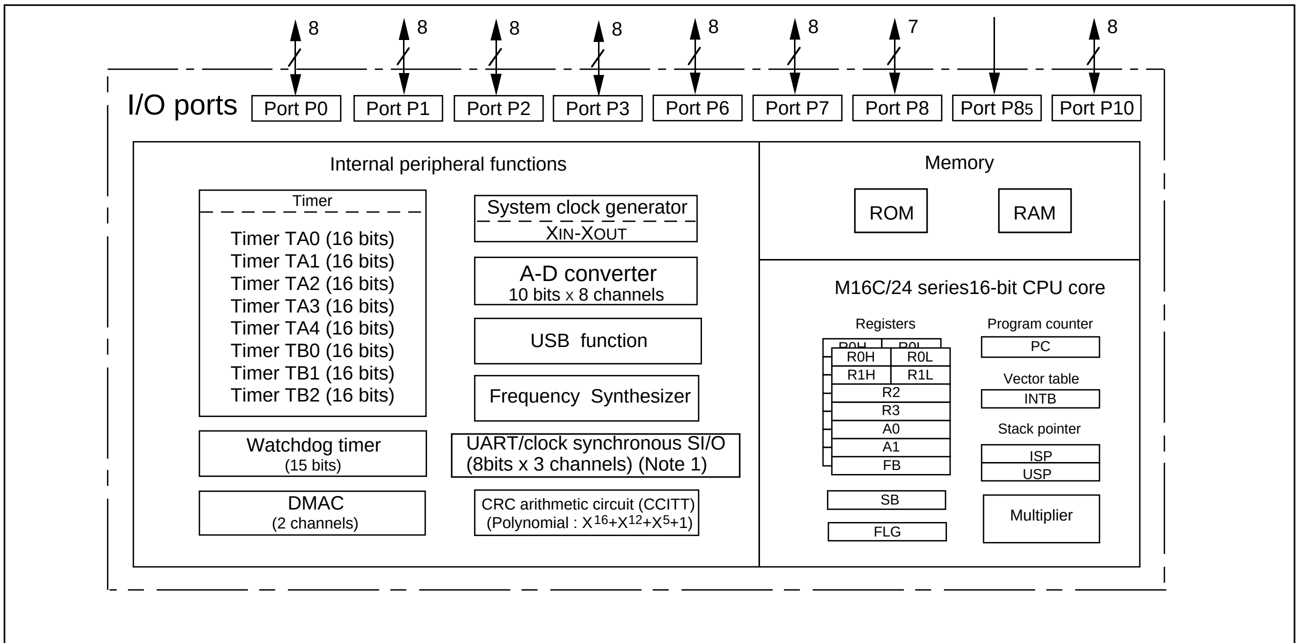


Figure 1: **Pin Configuration (top view)**

1.4 Block Diagram

Figure 2 is a block diagram of the M16C/24 group.



Note 1: One of serial I/O can be used for SIM interface.

Figure 2: Block diagram of M16C/24 group

1.5 Performance outline

Table 1 is a performance outline of the M16C/24 group.

Table 1: Performance outline of M16C/24 group

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		83ns (f(XIN)=12MHz)
Memory capacity	ROM	(See Figure 3: ROM capacity field)
	RAM	
I/O port	P0 to P3, P6,P7, P8 (except P85), P10	8 bits x 7, 7 bits x 1
Input port	P85	1 bit x 1
Multifunction Timer	TA0, TA1, TA2, TA3, TA4	16 bits x 5
General purpose Timer	TB0, TB1, TB2	16 bits x 3
Serial I/O	UART0, UART1, UART2	(UART or clock synchronous) x 3
A-D converter		10 bits x 8 channels
DMAC		2 channels (trigger: 24 sources)
CRC calculation circuit		CRC-CCITT
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		21 internal and 4 external sources, 4 software sources, 7 levels
Clock-generating circuit		Built-in clock generation circuit (built-in feedback resistor, and external ceramic or quartz oscillator)
Supply voltage		4.1 to 5.5V (f(XIN)=12MHz, without software wait)
Power consumption		83 mA @ 12MHz
I/O characteristics	I/O withstand voltage	5V
	Output current	20 mA available on ports P20 through P27; also ports P70, P72, P74, P76, and P80 are available.
Operating temperature		–40 to 85^{o}C
Device configuration		CMOS high performance silicon gate
Package		80-pin plastic molded QFP

Performance outline

Mitsubishi plans to release the following products in the M16C/24 group:

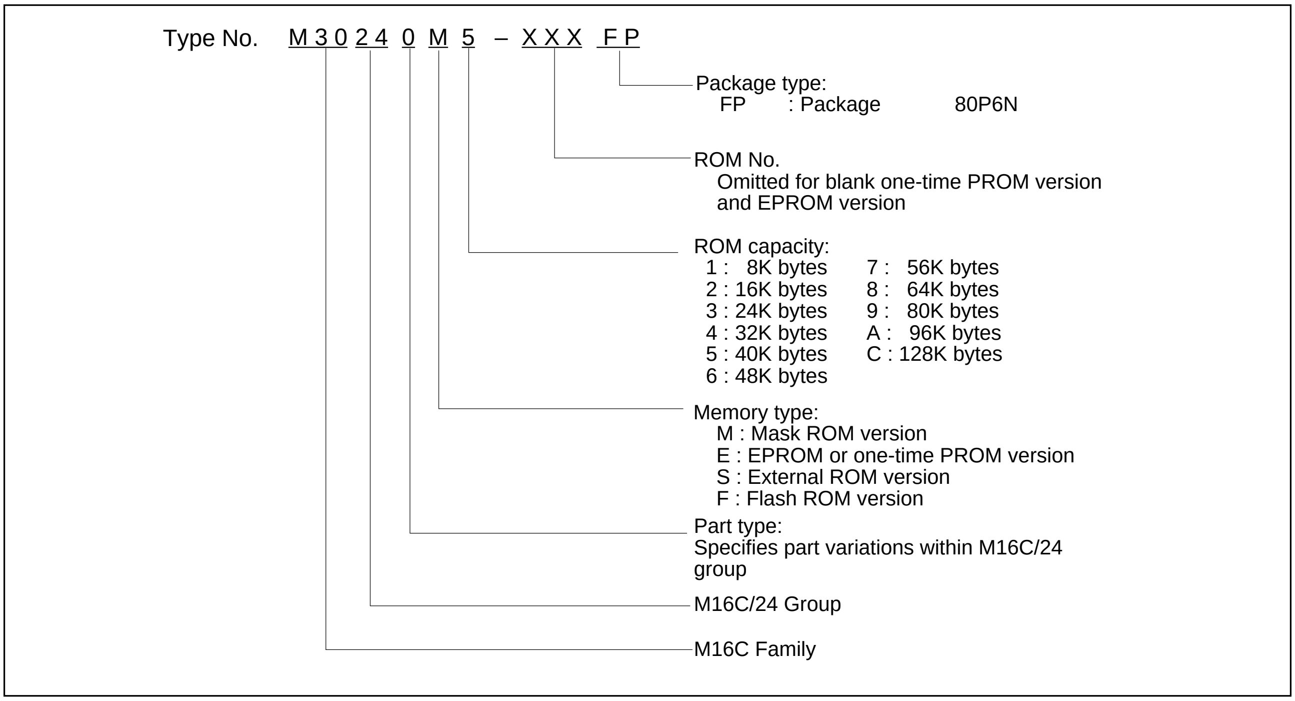
(1) Support for mask ROM version, one-time PROM version, and EPROM version

(2) ROM capacity

(3) Package

- 80P6N: Plastic molded QFP (mask ROM version and one-time PROM version)

Figure 3 shows the type number, memory size and package for the M16C/24 group.



Type No. M30240M5–XXXFP

Package type:
FP : Package 80P6N

ROM No.
Omitted for blank one-time PROM version and EPROM version

ROM capacity:

1 :	8K bytes	7 :	56K bytes
2 :	16K bytes	8 :	64K bytes
3 :	24K bytes	9 :	80K bytes
4 :	32K bytes	A :	96K bytes
5 :	40K bytes	C :	128K bytes
6 :	48K bytes		

Memory type:
M : Mask ROM version
E : EPROM or one-time PROM version
S : External ROM version
F : Flash ROM version

Part type:
Specifies part variations within M16C/24 group

M16C/24 Group

M16C Family

Figure 3: Type number, memory size, and package

1.6 Pin Description

Table 2: Figure Pin Description

Pin #	Name	I/O	Description
1	$P8_7$	I/O	CMOS I/O port. This pin also functions as an external trigger for A-D conversion.
2	$P8_6$	I/O	CMOS I/O port. This pin also functions as the start of frame (SOF) pulse for the USB module.
3	$P8_5$/($\overline{NMI}$)	I	CMOS input port. This pin also functions as a non-maskable external interrupt.
4,5	$P8_4$ ~ $P8_3$	I/O	CMOS I/O port. These pins also functions as external interrupt 1 and are used to enable the stealth detach function for the USB transceiver.
6	EXTCAP	I	An external capacitor (Ext. Cap) pin. If V_{dd} (AV_{dd}) =5V is used for the entire chip, a 2µf or larger capacitor connects between this pin and V_{ss} to ensure proper operation of the USB line driver. This option is enabled by setting bit 4 of the USB control register (0013_{16}) to a “1”.
7	BYTE	I	Connect this pin to Vss
8	CNV_{ss}	I	Connect this pin to Vss
9	USB D^+	I/O	USB D+ voltage line interface, a series resistor of 33 Ω is connected to this pin.
10	USB D^-	I/O	USB D- voltage line interface, a series resistor of 33 Ω is connected to this pin.
11	$\overline{RESET}$	I	A “L” on this input resets the microcomputer.
12	X_{out}	O	See Xin
13	V_{ss}	I	Ground: V_{ss} = 0V
14	X_{in}	I	Input and output signals to and from the internal clock generation circuit. Connect a ceramic resonator or quartz crystal between X_{in} and X_{out} pins to set the oscillation frequency. If an external clock is used, connect the clock source to the X_{in} pin and leave the X_{out} pin open.
15	V_{cc}	I	Power: V_{cc} = 4.1~ 5.5V
16	$P8_2$	I/O	CMOS I/O port. This pin also functions as external interrupt 0.
17-18	$P8_1$ ~ $P8_0$	I/O	CMOS I/O port . Pins in this port also function as TimerA4 input and output as selected by software.
19-22	$P7_7$ ~ $P7_4$	I/O	CMOS I/O port . Pins in this port also function as timer pins. $P7_7$ and $P7_6$ can function as TimerA3 input and output as selected by software. $P7_5$ and $P7_4$ can function as TimerA2 input and output as selected by software.
23-26	$P7_3$ ~ $P7_0$	I/O	CMOS I/O port . Pins in this port also function as UART2 CTS, RTS, CLK, RXD, and TXD as selected by software. $P7_3$ and $P7_2$ can function as TimerA1 input and output as selected by software. $P7_1$ and $P7_0$ can function as TimerA0 input and output as selected by software.
27-30	$P6_7$ ~ $P6_4$	I/O	CMOS I/O port . Pins in this port also function as UART1 CTS, RTS, CLK, Serial Clock, RXD, and TXD as selected by software. TXD(OE~) and RTS(SUSPEND) in addition to D+ and D- can be used to run the device in USB bypass mode.
31-34	$P6_3$ ~ $P6_0$	I/O	CMOS I/O port . Pins in this port also function as UART0 CTS, RTS, CLK, RXD, and TXD as selected by software.
35-42	$P3_7$ ~ $P3_0$	I/O	CMOS I/O port.
43-50	$P2_7$/LED7 ~ $P2_0$/LED0	I/O	CMOS I/O port. These pins are capable of driving up to 20mA for LEDs.

Table 2: Figure Pin Description

Pin #	Name	I/O	Description
51	V_{cc}	I	Power: V_{cc} = 4.1~ 5.5V
52	$P1_7/\overline{KI}_{\overline{15}}$	I/O	CMOS I/O port. This port can also function as the key-on wakeup interrupt $\overline{KI15}$.
53	V_{ss}	I	Ground: V_{ss} = 0V
54-60	$P1_6/\overline{KI}_{\overline{14}}$ ~ $P1_0/\overline{KI}_{\overline{8}}$	I/O	CMOS I/O port. This port can also function as the key-on wakeup interrupts ($\overline{KI8}$ ~ $\overline{KI14}$).
61-68	$P0_7/\overline{KI}_{\overline{7}}$ ~ $P0_0/\overline{KI}_{\overline{0}}$	I/O	CMOS I/O port. This port can also function as the key-on wakeup interrupts ($\overline{KI0}$ ~ $\overline{KI7}$).
69-76	$P10_7$ ~ $P10_0$	I/O	CMOS I/O port. These pins also function as Analog inputs 7-0 for A-D conversion
77	AV_{ss}	I	Analog ground: AV_{ss} = 0V
78	LPF	O	Loop filter for the frequency synthesizer.
79	V_{REF}	I	This pin is the reference voltage input for the A-D converter.
80	AV_{cc}	I	Analog power: AV_{cc} = 4.75~ 5.25V

1.7 Overview

The M30240 device is a single chip PC peripheral microcontroller based on the Universal Serial Bus (USB) Version 1.1 specification. This device provides interface between a USB-equipped host computer and PC peripherals such as telephones, audio systems, and digital cameras. The M30240 block diagram is shown in Figure 4.

The USB function control unit of the M30240 device can support all four data transfer types listed in the USB specification: Isochronous, Interrupt, Bulk, and Control. Each transfer type is used for controlling a different set of PC peripherals. Isochronous transfers provide guaranteed bus access, a constant data rate, and error tolerance for devices such as computer-telephone integration (CTI) and audio systems. Interrupt transfers are designed to support human input devices (HID) that communicate small amounts of data infrequently. Bulk transfers are necessary for devices such as digital cameras and scanners that communicate large amounts of data to the PC as bus bandwidth becomes free. Finally, control transfers are supported and are useful for bursty, host-initiated type communication where bus management is the primary concern.

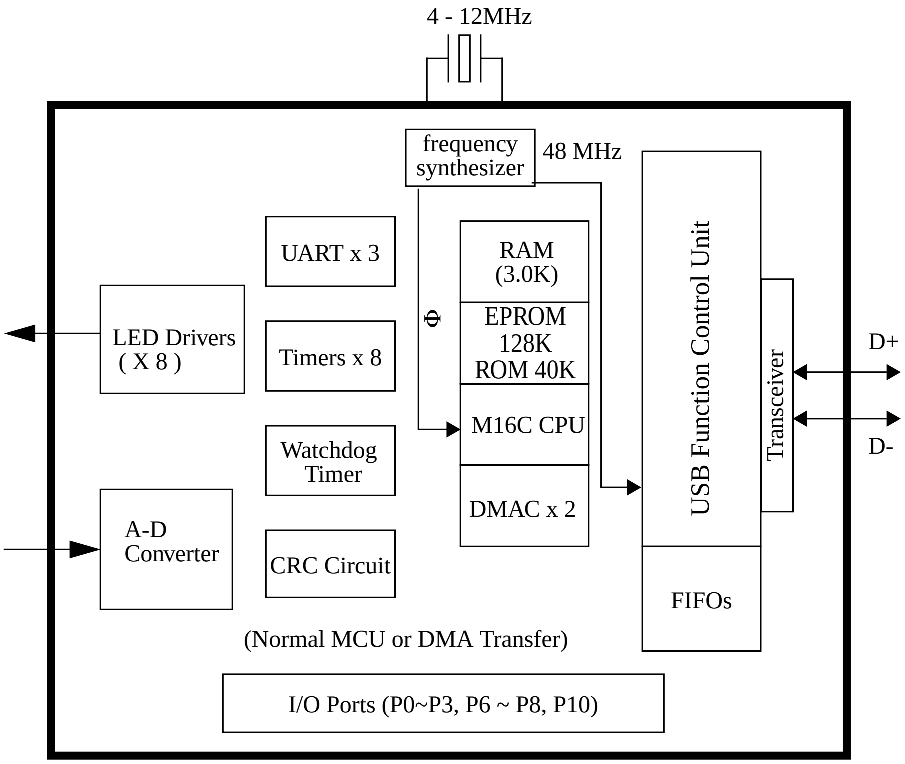


Figure 4: **M30240 block diagram**

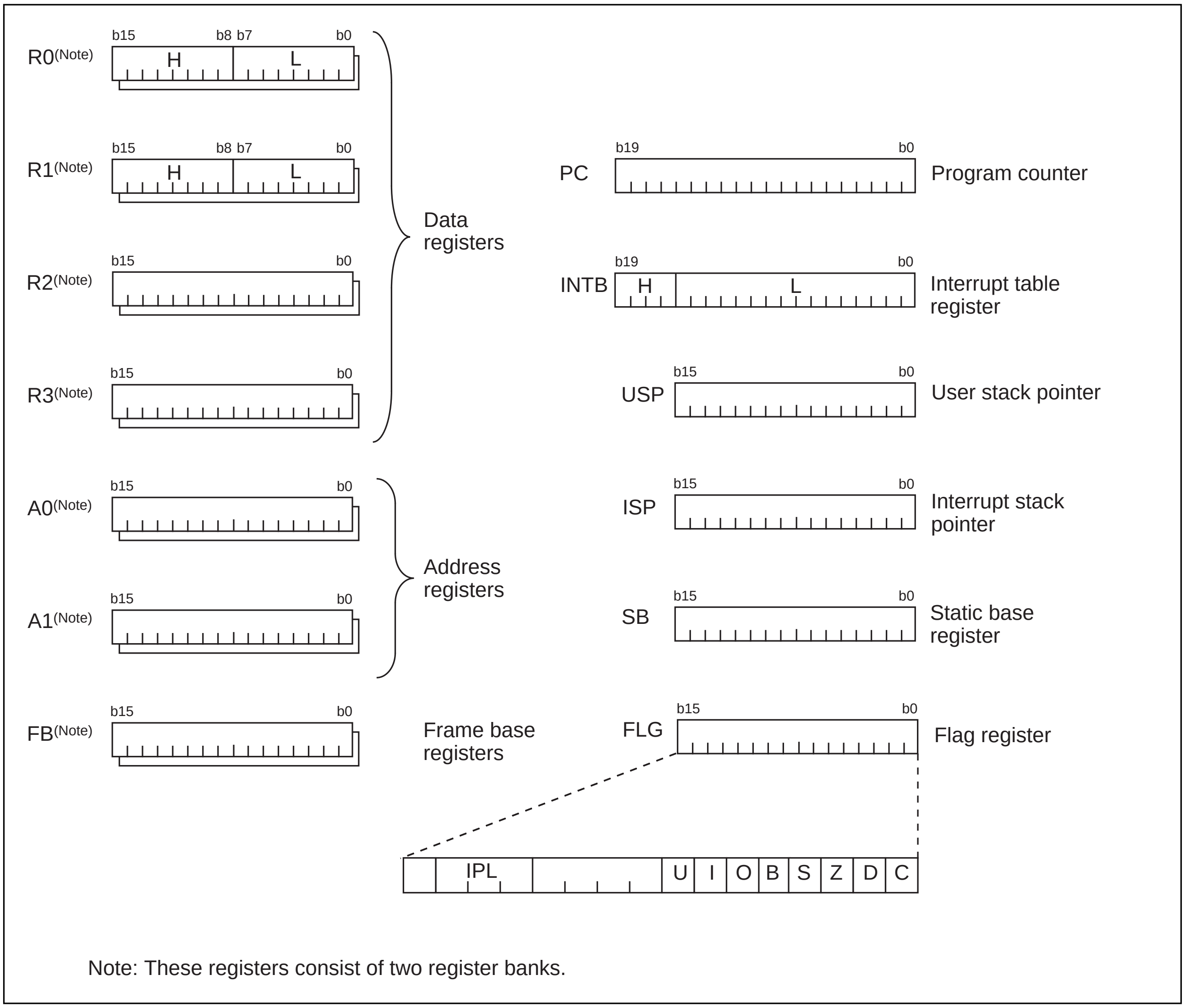
2.0 Operation of Functional Blocks

The M16C/24 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 USB, timers, serial I/O, DMAC, CRC calculation circuit, A-D converter, and I/O ports.

The following explains each unit.

2.1 Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 5. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.



Note: These registers consist of two register banks.

Figure 5: 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 each can be used as separate 8-bit data registers, high-order bits as (R0H/R1H), and low-order bits as (R0L/R1L). In some instructions, registers R2 and R0, as well as R3 and R1, can be used as 32-bit data registers (R2R0/R3R1).

(2) Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 16 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address 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. INTB can be used as separate registers of four high-order bits and 16 low-order bits.

(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 6 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; otherwise, cleared to “0”.

- **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 interrupts 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 M16C software manual for details.

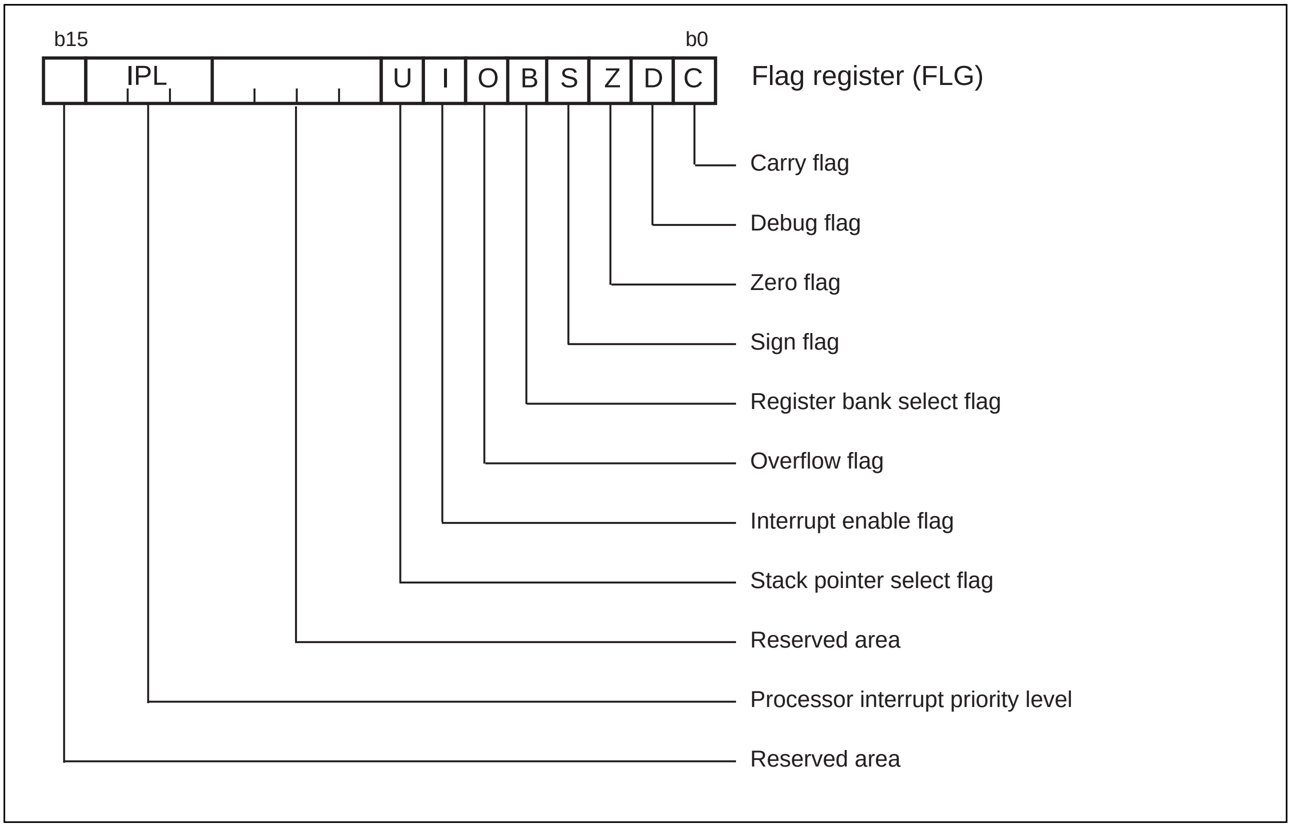
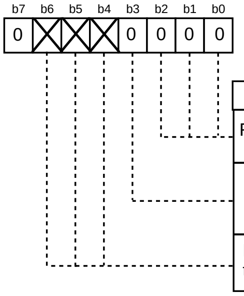


Figure 6: Flag register (FLG)

2.2 Processor Mode

Figure 7 shows the processor mode registers 0 and 1.

Processor mode register 0 (Note 1)

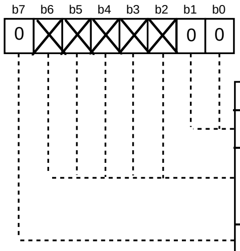


Symbol	Address	When reset
PM0	0004_{16}	00_{16} (Note)

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.	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

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)



Symbol	Address	When reset
PM1	0005_{16}	$00XXXXX0_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

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

Figure 7: Processor mode registers 0 and 1

2.3 Memory

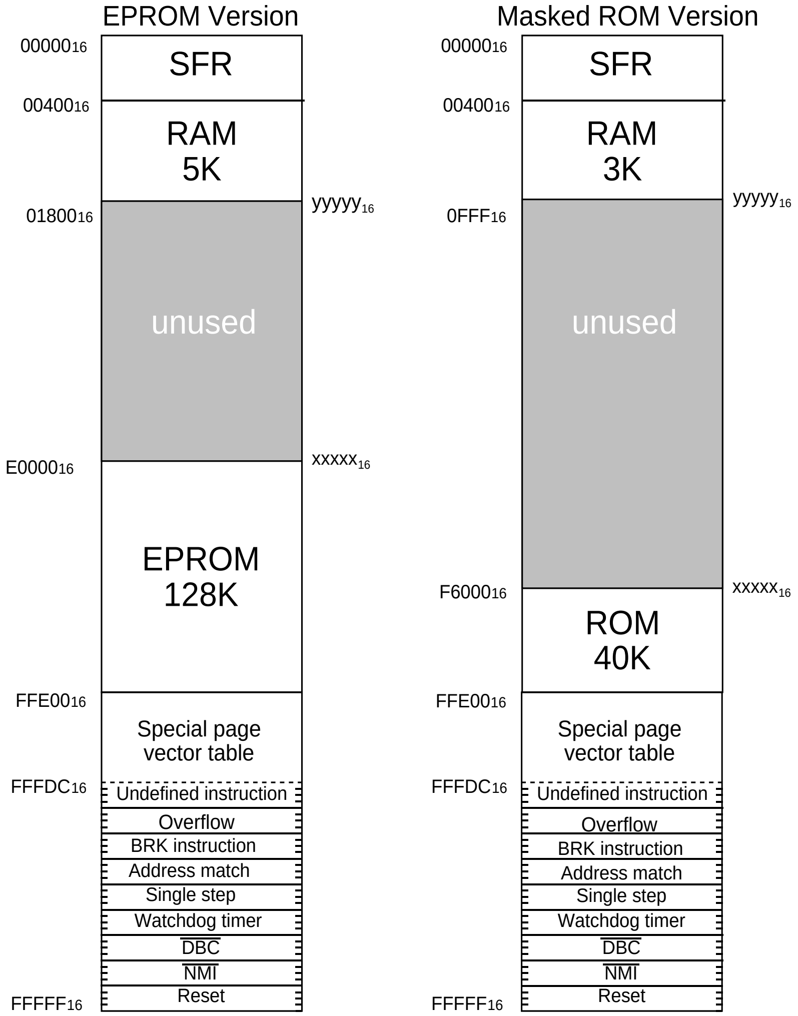


Figure 8: Memory Map

Figure 8 is a memory map of the M16C/24 group. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$. Addresses above $xxxxx_{16}$ are ROM. For example, in the M30240EC-XXXFP, there is 128K bytes of internal ROM from $E0000_{16}$ to $FFFFF_{16}$. The special page vector table is mapped from $FFE00_{16}$ to $FFFDB_{16}$. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as two-byte instructions, reducing the number of program steps.

The vector table for fixed interrupts such as the reset and $\overline{\text{NMI}}$ are mapped from $FFFDC_{16}$ to $FFFFF_{16}$. The starting addresses of the interrupt routines are stored here. The address of the vector table for software interrupts can be set as desired using the internal register (INTB). See Section 2.12 on interrupts for further details.

Addresses below $yyyyy_{16}$ are RAM. For example, in M30240EC-XXXFP, 5K bytes of internal RAM are mapped to the space from 00400_{16} to $017FF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.The SFR area is mapped to 00000_{16} to $003FF_{16}$. This area accommodates control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers. Section 2.4 describes the SFR area for peripheral unit control registers. Any part of the SFR area that is unoccupied is reserved and cannot be used for other purposes.

2.4 SFR MAP

The table below shows the peripheral control registers, their addresses, names, acronyms, and values after reset.

Address	Register name	Acronym	Value after reset
0000_{16}			
0001_{16}			
0002_{16}			
0003_{16}			
0004_{16}	Processor mode register 0	PM0	00_{16}
0005_{16}	Processor mode register 1	PM1	0 0 ▨ 0
0006_{16}	System clock control register 0	CM0	00_{16}
0007_{16}	System clock control register 1	CM1	20_{16}
0008_{16}			
0009_{16}	Address match interrupt enable register	AIER	▨ 0 0
$000A_{16}$	Protect register	PRCR	▨ 0 0 0
$000B_{16}$			
$000C_{16}$	USB control register	USBC	00_{16}
$000D_{16}$			
$000E_{16}$	Watchdog timer start register	WDTS	
$000F_{16}$	Watchdog timer control register	WDC	0 0 0 ? ? ? ? ?
0010_{16}	Address match interrupt register 0	RMAD0	00_{16}
0011_{16}			00_{16}
0012_{16}			▨ 0 0 0 0
0013_{16}			
0014_{16}	Address match interrupt register 1	RMAD1	00_{16}
0015_{16}			00_{16}
0016_{16}			▨ 0 0 0 0
0017_{16}			
0018_{16}			
0019_{16}			
$001A_{16}$			
$001B_{16}$			
$001C_{16}$			
$001D_{16}$			
$001E_{16}$	Reserved		
$001F_{16}$	USB attach / detach register		00_{16}
0020_{16}	DMA0 source pointer	SAR0	
0021_{16}			
0022_{16}			
0023_{16}			
0024_{16}	DMA0 destination pointer	DAR0	
0025_{16}			
0026_{16}			
0027_{16}			
0028_{16}	DMA0 transfer counter	TCR0	
0029_{16}			
$002A_{16}$			
$002B_{16}$			
$002C_{16}$	DMA0 control register	DM0CON	0 0 0 0 0 ? 0 0
$002D_{16}$			
$002E_{16}$			
$002F_{16}$			
0030_{16}	DMA1 source pointer	SAR1	
0031_{16}			
0032_{16}			
0033_{16}			
0034_{16}	DMA1 destination pointer	DAR1	
0035_{16}			
0036_{16}			
0037_{16}			
0038_{16}	DMA1 transfer counter	TCR1	
0039_{16}			
$003A_{16}$			
$003B_{16}$			
$003C_{16}$	DMA1 control register	DM1CON	0 0 0 0 0 ? 0 0
$003D_{16}$			
$003E_{16}$			
$003F_{16}$			

SFR MAP

Address	Register name	Acronym	Value after reset
0040$_{16}$			
0041$_{16}$			
0042$_{16}$			
0043$_{16}$			
0044$_{16}$	Suspend interrupt control register	SUSPIC	▨▨▨▨ ? 0 0 0
0045$_{16}$			
0046$_{16}$	Resume interrupt control register	RSMIC	▨▨▨▨ ? 0 0 0
0047$_{16}$	USB SOF interrupt control register	SOFIC	▨▨▨▨ ? 0 0 0
0048$_{16}$			
0049$_{16}$			
004A$_{16}$	Bus collision detection interrupt control register	BCNIC	▨▨▨▨ ? 0 0 0
004B$_{16}$	DMA0 interrupt control register	DM0IC	▨▨▨▨ ? 0 0 0
004C$_{16}$	DMA1 interrupt conrol register	DM1IC	▨▨▨▨ ? 0 0 0
004D$_{16}$	Key input interrupt control register	KUPIC	▨▨▨▨ ? 0 0 0
004E$_{16}$	A-D conversion interrupt control register	ADIC	▨▨▨▨ ? 0 0 0
004F$_{16}$	UART2 transmit interrupt control register	S2TIC	▨▨▨▨ ? 0 0 0
0050$_{16}$	UART2 receive interrupt control register	S2RIC	▨▨▨▨ ? 0 0 0
0051$_{16}$	UART0 transmit interrupt control register	S0TIC	▨▨▨▨ ? 0 0 0
0052$_{16}$	UART0 receive interrupt control register	S0RIC	▨▨▨▨ ? 0 0 0
0053$_{16}$	UART1 transmit interrupt control register	S1TIC	▨▨▨▨ ? 0 0 0
0054$_{16}$	UART1 receive interrupt control register	S1RIC	▨▨▨▨ ? 0 0 0
0055$_{16}$	TIMER A0 interrupt control register	TA0IC	▨▨▨▨ ? 0 0 0
0056$_{16}$	TIMER A1 interrupt control register	TA1IC	▨▨▨▨ ? 0 0 0
0057$_{16}$	TIMER A2 interrupt control register	TA2IC	▨▨▨▨ ? 0 0 0
0058$_{16}$	TIMER A3 interrupt control register	TA3IC	▨▨▨▨ ? 0 0 0
0059$_{16}$	TIMER A4 interrupt control register	TA4IC	▨▨▨▨ ? 0 0 0
005A$_{16}$	TIMER B0 interrupt control register	TB0IC	▨▨▨▨ ? 0 0 0
005B$_{16}$	TIMER B1 interrupt control register	TB1IC	▨▨▨▨ ? 0 0 0
005C$_{16}$	Reset interrupt control register	RSTIC	▨▨▨▨ ? 0 0 0
005D$_{16}$	INT0 interrupt control register	INT0IC	▨▨ 0 0 ? 0 0 0
005E$_{16}$	INT1 interrupt control register	INT1IC	▨▨ 0 0 ? 0 0 0
005F$_{16}$	USB function interrupt control register	USBFIC	▨▨▨▨ ? 0 0 0
	- - -		
0300$_{16}$	USB address register	USBA	00$_{16}$
0301$_{16}$	USB power management register	USBPM	00$_{16}$
0302$_{16}$	USB interrupt status register 1	USBIS1	00$_{16}$
0303$_{16}$	USB interrupt status register 2	USBIS2	00$_{16}$
0304$_{16}$	USB interrupt enable register 1	USBER1	FF$_{16}$
0305$_{16}$	USB interrupt enable register 2	USBER2	33$_{16}$
0306$_{16}$	USB frame number register low	USBSOFL	00$_{16}$
0307$_{16}$	USB frame number register high	USBSOFH	00$_{16}$
0308$_{16}$	USB ISO control register	USBISOC	0 0 ▨▨▨▨▨▨
0309$_{16}$	USB DMA0 source register	USBSAR0	00$_{16}$
030A$_{16}$	USB DMA1 source register	USBSAR1	00$_{16}$
030B$_{16}$	USB endpoint enable	USBEPEN	? 0 0 0
030C$_{16}$			
030D$_{16}$			
030E$_{16}$			
030F$_{16}$			
0310$_{16}$	USB reserved		
0311$_{16}$	USB EP 0 control/status register		00$_{16}$
0312$_{16}$	USB reserved		
0313$_{16}$	USB EP 0 max packet size register		08$_{16}$
0314$_{16}$	USB reserved		
0315$_{16}$	USB EP 0 OUT write count		00$_{16}$
0316$_{16}$	USB reserved		
0317$_{16}$	USB reserved		
0318$_{16}$	USB reserved		
0319$_{16}$	USB EP 1 IN control/status register		00$_{16}$
031A$_{16}$	USB EP 1 OUT control/status register		00$_{16}$
031B$_{16}$	USB EP 1 IN max packet size register		00$_{16}$
031C$_{16}$	USB EP 1 OUT max packet size register		00$_{16}$
031D$_{16}$	USB EP 1 OUT write count		00$_{16}$
031E$_{16}$	USB reserved		
031F$_{16}$	USB reserved		

SFR MAP

Address	Register name	Acronym	Value after reset
0320_{16}	USB reserved		
0321_{16}	USB EP 2 IN control/status register		00_{16}
0322_{16}	USB EP 2 OUT control/status register		00_{16}
0323_{16}	USB EP 2 IN max packet size register		00_{16}
0324_{16}	USB EP 2 OUT max packet size register		00_{16}
0325_{16}	USB EP 2 OUT write count		00_{16}
0326_{16}	USB reserved		
0327_{16}	USB reserved		
0328_{16}	USB reserved		
0329_{16}	USB EP 3 IN control/status register		00_{16}
$032A_{16}$	USB EP 3 OUT control/status register		00_{16}
$032B_{16}$	USB EP 3 IN max packet size register		00_{16}
$032C_{16}$	USB EP 3 OUT max packet size register		00_{16}
$032D_{16}$	USB EP 3 OUT write count		00_{16}
$032E_{16}$	USB reserved		00_{16}
$032F_{16}$	USB reserved		
0330_{16}	USB reserved		
0331_{16}	USB EP 4 IN control/status register		00_{16}
0332_{16}	USB EP 4 OUT control/status register		00_{16}
0333_{16}	USB EP 4 IN max packet size register		00_{16}
0334_{16}	USB EP 4 OUT max packet size register		00_{16}
0335_{16}	USB EP 4 OUT write count		00_{16}
0336_{16}	USB reserved		
0337_{16}	USB reserved		
0338_{16}	USB EP 0 FIFO		
0339_{16}	USB EP 1 FIFO		
$033A_{16}$	USB EP 2 FIFO		
$033B_{16}$	USB EP 3 FIFO		
$033C_{16}$	USB EP 4 FIFO		
$033D_{16}$	reserved		
$033E_{16}$	reserved		
$033F_{16}$	reserved		
0340_{16}			
0341_{16}			
0342_{16}			
0343_{16}			
0344_{16}			
0345_{16}			
0346_{16}			
0347_{16}			
0348_{16}			
0349_{16}			
$034A_{16}$			
$034B_{16}$			
$034C_{16}$			
$034D_{16}$			
$034E_{16}$			
$034F_{16}$			
0350_{16}			
0351_{16}			
0352_{16}			
0353_{16}			
0354_{16}			
0355_{16}			
0356_{16}			
0357_{16}			
0358_{16}			
0359_{16}			
$035A_{16}$			
$035B_{16}$			
$035C_{16}$			
$035D_{16}$			
$035E_{16}$			
$035F_{16}$			

Address	Register name	Acronym	Value after reset
0370_{16}			
0371_{16}			
0372_{16}			
0373_{16}			
0374_{16}			
0375_{16}			
0376_{16}			
0377_{16}	Reserved		
0378_{16}	UART2 transmit / receive mode register	U2MR	00_{16}
0379_{16}	UART2 bit rate generator	U2BRG	
$037A_{16}$	UART2 transmit buffer register	U2TB	
$037B_{16}$			
$037C_{16}$	UART2 transmit /receive control register 0	U2C0	08_{16}
$037D_{16}$	UART2 transmit / receive control register 1	U2C1	02_{16}
$037E_{16}$	UART2 receive buffer register	U2RB	
$037F_{16}$			
0380_{16}	Count start flag	TABSR	00_{16}
0381_{16}	Clock prescaler reset flag	CPSRF	0 ▨
0382_{16}	One-shot start flag	ONSF	0 0 ▨ 0 0 0 0 0
0383_{16}	Trigger select register	TRGSR	00_{16}
0384_{16}	Up-down flag	UDF	00_{16}
0385_{16}			
0386_{16}	Timer A0	TA0	
0387_{16}			
0388_{16}	Timer A1	TA1	
0389_{16}			
$038A_{16}$	Timer A2	TA2	
$038B_{16}$			
$038C_{16}$	Timer A3	TA3	
$038D_{16}$			
$038E_{16}$	Timer A4	TA4	
$038F_{16}$			
0390_{16}	Timer B0	TB0	
0391_{16}			
0392_{16}	Timer B1	TB1	
0393_{16}			
0394_{16}	Timer B2	TB2	
0395_{16}			
0396_{16}	Timer A0 mode register	TA0MR	00_{16}
0397_{16}	Timer A1 mode register	TA1MR	00_{16}
0398_{16}	Timer A2 mode register	TA2MR	00_{16}
0399_{16}	Timer A3 mode register	TA3MR	00_{16}
$039A_{16}$	Timer A4 mode register	TA4MR	00_{16}
$039B_{16}$	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
$039E_{16}$			
$039F_{16}$			
$03A0_{16}$	UART0 transmit / receive mode register	U0MR	00_{16}
$03A1_{16}$	UART0 bit rate generator	U0BRG	
$03A2_{16}$	UART0 transmit buffer register	U0TB	
$03A3_{16}$			
$03A4_{16}$	UART0 transmit / receive control register 0	U0C0	08_{16}
$03A5_{16}$	UART0 transmit / receive control register 1	U0C1	02_{16}
$03A6_{16}$	UART0 receive buffer register	U0RB	
$03A7_{16}$			
$03A8_{16}$	UART1 transmit / receive mode register	U1MR	00_{16}
$03A9_{16}$	UART1 bit rate generator	U1BRG	
$03AA_{16}$	UART1 transmit buffer register	U1TB	
$03AB_{16}$			
$03AC_{16}$	UART1 transmit / receive control register 0	U1C0	08_{16}
$03AD_{16}$	UART1 transmit / receive control register 1	U1C1	02_{16}
$03AE_{16}$	UART1 receive buffer register	U1RB	
$03AF_{16}$			

Address	Register name	Acronym	Value after reset
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 cause select register	DM0SL	00$_{16}$
03B9$_{16}$			
03BA$_{16}$	DMA1 cause select register	DM1SL	00$_{16}$
03BB$_{16}$			
03BC$_{16}$	CRC data register	CRCD	
03BD$_{16}$			
03BE$_{16}$	CRC input register	CRCIN	
03BF$_{16}$			
03C0$_{16}$	A-D register 0	AD0	
03C1$_{16}$			
03C2$_{16}$	A-D register 1	AD1	
03C3$_{16}$			
03C4$_{16}$	A-D register 2	AD2	
03C5$_{16}$			
03C6$_{16}$	A-D register 3	AD3	
03C7$_{16}$			
03C8$_{16}$	A-D register 4	AD4	
03C9$_{16}$			
03CA$_{16}$	A-D register 5	AD5	
03CB$_{16}$			
03CC$_{16}$	A-D register 6	AD6	
03CD$_{16}$			
03CE$_{16}$	A-D register 7	AD7	
03CF$_{16}$			
03D0$_{16}$			
03D1$_{16}$			
03D2$_{16}$			
03D3$_{16}$			
03D4$_{16}$	A-D control register 2	ADCON2	▨ ▨ ▨ ▨ ▨ ▨ ▨ 0
03D5$_{16}$			
03D6$_{16}$	A-D control register 0	ADCON0	0 0 0 0 0 ? ? ?
03D7$_{16}$	A-D conrol register 1	ADCON1	00$_{16}$
03D8$_{16}$			
03D9$_{16}$			
03DA$_{16}$			
03DB$_{16}$	Frequency synthesizer clock control	FSCCR	00$_{16}$
03DC$_{16}$	Frequency synthesizer control	FSC	60$_{16}$
03DD$_{16}$	Frequency synthesizer multiplier control	FSM	FF$_{16}$
03DE$_{16}$	Frequency synthesizer prescaler control	FSP	FF$_{16}$
03DF$_{16}$	Frequency synthesizer divider	FSD	FF$_{16}$
03E0$_{16}$	Port P0	P0	
03E1$_{16}$	Port P1	P1	
03E2$_{16}$	Port P0 direction register	PD0	00$_{16}$
03E3$_{16}$	Port P1 direction register	PD1	00$_{16}$
03E4$_{16}$	Port P2	P2	
03E5$_{16}$	Port P3	P3	
03E6$_{16}$	Port P2 direction register	PD2	00$_{16}$
03E7$_{16}$	Port P3 direction register	PD3	00$_{16}$
03E8$_{16}$			
03E9$_{16}$			
03EA$_{16}$			
03EB$_{16}$			
03EC$_{16}$	Port P6	P6	
03ED$_{16}$	Port P7	P7	
03EE$_{16}$	Port P6 direction register	PD6	00$_{16}$
03EF$_{16}$	Port P7 direction register	PD7	00$_{16}$

Address	Register name	Acronym	Value after reset
03F0$_{16}$	Port P8	P8	
03F1$_{16}$			
03F2$_{16}$	Port P8 direction register	PD8	00$_{16}$
03F3$_{16}$			
03F4$_{16}$	Port P10	P10	
03F5$_{16}$			
03F6$_{16}$	Port P10 direction register	PD10	00$_{16}$
03F7$_{16}$			
03F8$_{16}$			
03F9$_{16}$			
03FA$_{16}$	P2 drive capacity	P2DR	
03FB$_{16}$	PWM drive capacity	PWMDR	
03FC$_{16}$	Pull-up control register 0	PUR0	00$_{16}$
03FD$_{16}$	Pull-up control register 1	PUR1	00$_{16}$
03FE$_{16}$			
03FF$_{16}$			

2.5 Reset

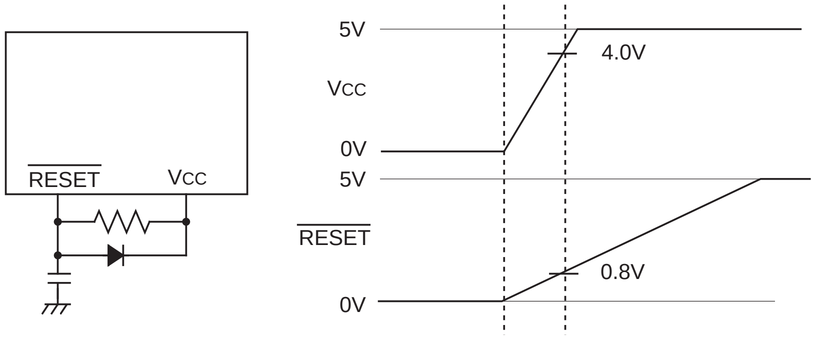
There are two types of resets: hardware and software. In both cases, operation is the same after the reset. (See “Software Reset” for further details regarding software resets.) This section explains on hardware resets.

Note: The USB peripheral is only reset during a hardware reset; software resets do not affect the USB peripheral.

When the supply voltage is within the range where operation is guaranteed, a reset is effected by holding the reset pin level “L” (0.2VCC max.) for at least 20 XIN cycles. When the reset pin level is then returned to the “H” level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure 9 shows an example of a reset circuit. Figure 10 shows the reset sequence.

.



Example when f(XIN) = 10MHz and VCC = 5V.

Figure 9: Reset circuit

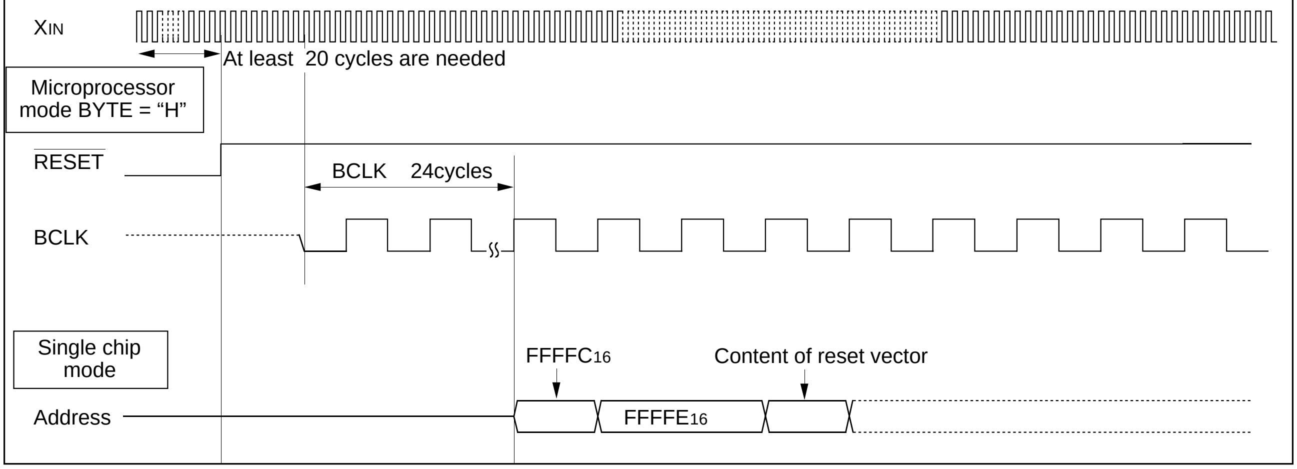


Figure 10: Reset sequence

When the RESET pin level = “L”, all ports change to input mode (floating.) Table 3 shows the status of the other pins while the RESET pin level is “L”.

Table 3: Main clock-generating circuits

Functions	Main clock-generating circuit
Use of clock	• CPU’s operating clock source • Internal peripheral units’ operating clock source
Usable oscillator	Ceramic or crystal oscillator
Pins to connect oscillator	XIN, XOUT
Oscillation stop/restart function	Available
Oscillator status immediately after reset	Oscillating

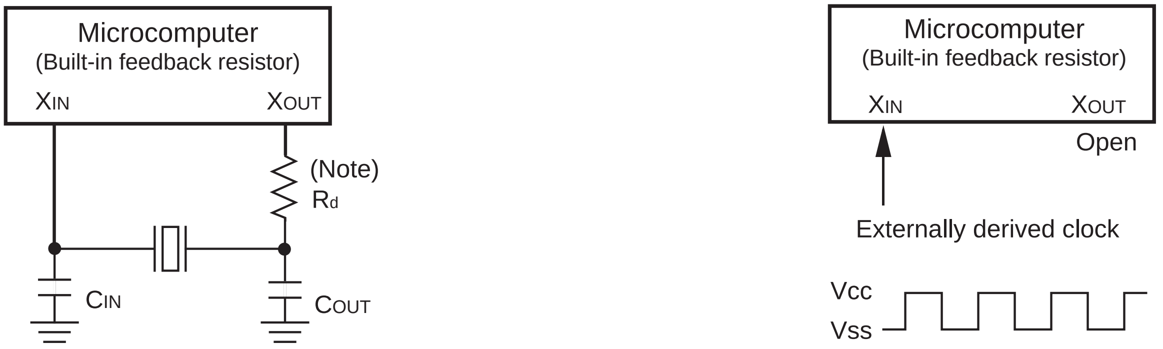
2.6 Software Reset

Writing "1" to bit 3 of processor mode register 0 (address 0004_{16}) applies a (software) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are preserved.

2.7 Clock-Generating Circuit

The clock-generating circuit contains one oscillator circuit that supplies the operating clock sources to the CPU and internal peripheral units.Example of oscillator circuit

Figure 11 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. Circuit constants in Figure 11 vary with each oscillator used. Use circuit constant values recommended by the oscillator manfacturer.



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.

Figure 11: Examples of clock source

2.8 Clock Control

Figure 12 shows the block diagram of the clock-generating circuit.

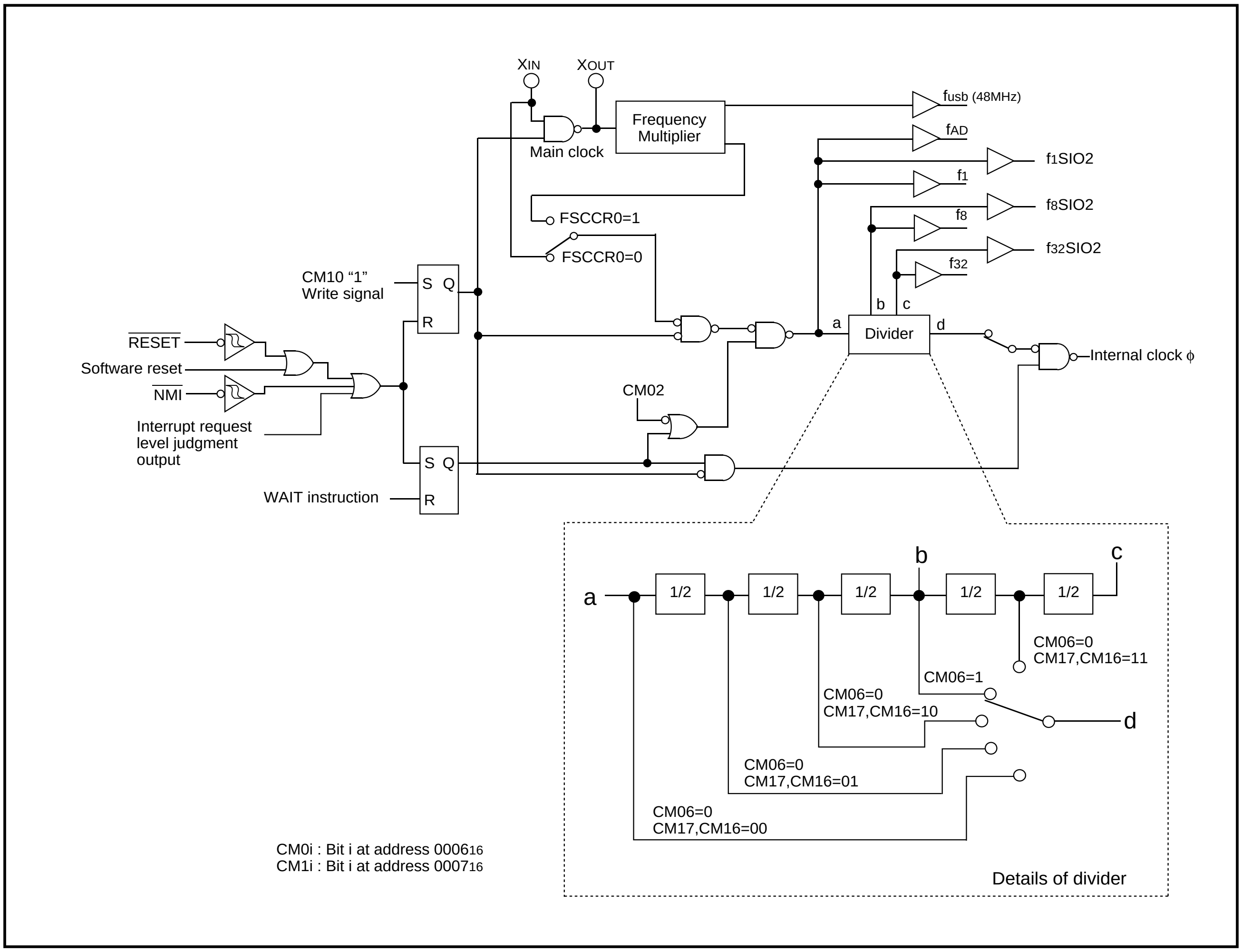


Figure 12: Clock-generating circuit

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 internal clock ϕ. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock reduces the power dissipation.

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the XOUT pin can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the XOUT pin reduces the power dissipation. This bit defaults to “1” when shifting to stop mode and after a reset.

(2) Internal clock ϕ

The internal clock ϕ is the clock that drives the CPU, and is either the main clock or is derived by dividing the main clock by 2, 4, 8, or 16. The internal clock ϕ is derived by dividing the main clock by 8 after a reset.

When shifting to stop mode, the main clock division select bit (bit 6 at 0006_{16}) is set to “1”.

(3) Peripheral function clock

- **f1, f8, f32**

The clock for the peripheral devices is derived from the main clock or by dividing it by 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.

- **fAD**

This clock has the same frequency as the main clock and is used for A-D conversion.

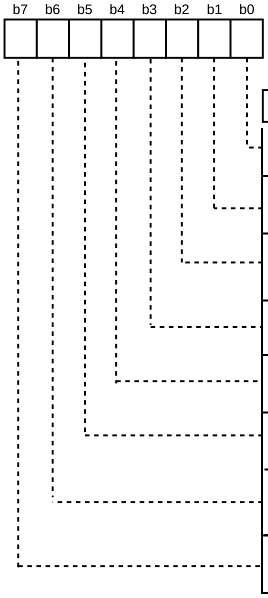
(4) Clock Output

In single-chip mode, the clock output function select bits (bits 0 and 1 at address 0006_{16}) enable f8 or f32 to be output from the P37/CLKOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) is set to “1”, the output of f8 and f32 stops when a WAIT instruction is executed.

Figure 13 shows the system clock control registers 0 and 1.

Clock Control

System clock control register 0 (Note 1)

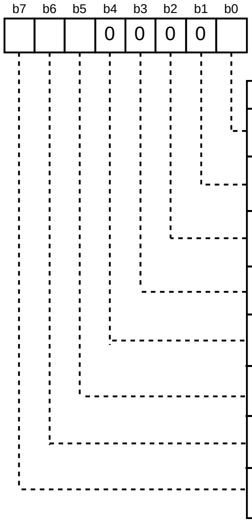


Symbol	Address	When reset
CM0	0006$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit	b1 b0 0 0 : I/O port P3$_7$ 0 1 : Not used 1 0 : f$_8$ output 1 1 : f$_{32}$ output	○	○
CM01			○	○
CM02	WAIT peripheral function clock stop bit	0 : Do not stop f$_1$, f$_8$, f$_{32}$ in wait mode 1 : Stop f$_1$, f$_8$, f$_{32}$ in wait mode	○	○
	Reserved bit	Always set to "0"	○	○
	Reserved bit	Always set to "0"	○	○
CM05	Main clock (X$_{IN}$-X$_{OUT}$) stop bit (Note 3) (Note 4)	0 : On 1 : Off	○	○
CM06	Main clock division select bit 0 (Note 2)	0 : CM16 and CM17 valid 1 : Division by 8 mode	○	○
	Reserved bit	Always set to "0"	○	○

Note 1: Set bit 0 of the protect register (address 000A$_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shiffing to stop mode.
Note 3: When entering power saving mode, main clock stops using this bit. When returning from stop mode and operating with X$_{IN}$, set this bit to “0”. When main clock oscillation is operating by itself, set system clock select bit (CM07) to “1” before setting this bit to “1”.
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.

System clock control register 1 (Note 1)



Symbol	Address	When reset
CM1	0007$_{16}$	20$_{16}$

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit	0 : Clock on 1 : All clocks off (stop mode)	○	○
	Reserved bit	Always set to “0”	○	○
	Reserved bit	Always set to “0”	○	○
	Reserved bit	Always set to “0”	○	○
	Reserved bit	Always set to “0”	○	○
CM15	X$_{IN}$-X$_{OUT}$ drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
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	○	○
CM17				

Note 1: Set bit 0 of the protect register (address 000A$_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shiffing to stop mode.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 0006$_{16}$) is “0”. If “1”, division mode is fixed at 8.

Figure 13: System clock control registers 0 and 1

2.9 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 VCC remains above 2V.

Because the oscillation of internal clock ϕ, f1 to f32, and fAD stops in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A operates, 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. Table 4 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled.

When shifting to stop mode, the main clock division select bit 0 (bit 6 at 0006_{16}) is set to “1”.

Table 4: Port status during stop mode

Pin		Single-chip mode
Port		Retains status before stop mode
CLKOUT	When f8, f32 selected	Retains status before stop mode

2.10 Wait Mode

When a WAIT instruction is executed, the internal clock ϕ stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the internal clock ϕ 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. Table 5 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts using as internal clock ϕ the clock that had been selected when the WAIT instruction was executed.

Table 5: Port status during wait mode

Pin		Single-chip mode
Port		Retains status before stop mode
CLKout	When f8, f32 selected	Does not stop when the WAIT peripheral function clock stop bit is “0” When the WAIT peripheral function clock stop bit is “0”, the status immediately prior to entering wait mode is maintained.

Status Transition Of Internal Clock ϕ

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for internal clock ϕ. Table 6 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

After a reset, operation defaults to division by 8 mode. When shifting to stop mode, the main clock division select bit 0 (bit 6 at address 0006_{16}) is set to “1”. The following shows the operational modes of internal clock

(1) Division by 2 mode

The main clock is divided by 2 to obtain the internal clock ϕ.

(2) Division by 4 mode

The main clock is divided by 4 to obtain the internal clock ϕ.

(3) Division by 8 mode

The main clock is divided by 8 to obtain the internal clock ϕ. Note that oscillation of the main clock must have stabilized before transferring from this mode to another mode.

(4) Division by 16 mode

The main clock is divided by 16 to obtain the internal clock ϕ.

(5) No-division mode

The main clock is used as internal clock

Table 6: Operating modes dictated by settings of system clock control registers 0 and 1

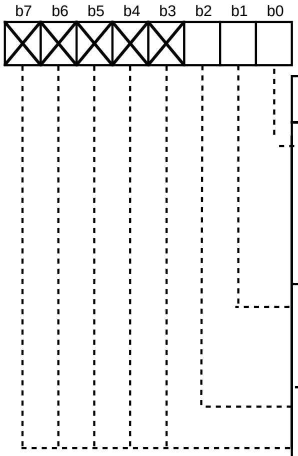
CM17	CM16	CM06	CM05	CM04	Operating mode of internal clock
0	1	0	0	Invalid	Division by 2 mode
1	0	0	0	Invalid	Division by 4 mode
Invalid	Invalid	1	0	Invalid	Division by 8 mode
1	1	0	0	Invalid	Division by 16 mode
0	0	0	0	Invalid	No-division mode

2.11 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 14 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}) and frequency synthesizer registers can only be changed when the respective bit in the protect register is set to “1”.

The system clock control registers 0 and 1 write-enable bit (bit 0 at $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at $000A_{16}$) do not automatically return to “0” after a value has been written to an address. The program must therefore be written to return these bits to “0”.

Protect register



Symbol
PRCR

Address
$000A_{16}$

When reset
$XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16}) and frequency synthesizer registers (addresses $03DB_{16}$ and $03DF_{16}$)	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
Reserved bit		Must always be set to "0"	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

Figure 14: Protect register

2.12 Interrupts

Table 7 and Table 8 show the interrupt sources and vector table addresses. When an interrupt is received, the program is executed from the address shown by the respective interrupt vector.

The vector table addresses for the interrupts in Table 7 are fixed (interrupt vector addresses). These interrupts are not affected by the interrupt enable flag (I flag) (non-maskable interrupts).

The vector table addresses for the interrupts in Table 8 are variable, being determined as relative to the fixed address in the interrupt table register (INTB). These interrupts can be enabled or disabled using the interrupt enable flag (I flag) (maskable interrupts). 64 vectors can be set in the interrupt table register (INTB). Any of software interrupts 0 to 63 can be assigned to each vector. By using the INT instruction to specify a software interrupt number, the program can be executed starting at the address indicated by the respective vector. The BRK instruction interrupt has interrupt vectors in both the fixed vector address and variable vector address. When the contents of FFFE4$_{16}$ through FFFE7$_{16}$ are all “FF$_{16}$”, the program is executed from the address shown in the BRK instruction interrupt vector in the variable vector address.

Specify the starting address of the interrupt program in the interrupt vector. Figure 15 shows the format for specifying the address.

Table 7: Interrupt vectors (fixed interrupt vector addresses)

Interrupt source	Vector table addresses Address(L) to Address(H)	Remarks
Undefined instruction	FFFDC$_{16}$ to FFFDF$_{16}$	Interrupt on UND instruction
Overflow	FFFE0$_{16}$ to FFFE3$_{16}$	Interrupt on INTO instruction
BRK instruction	FFFE4$_{16}$ to FFFE7$_{16}$	If the vector is filled with FF$_{16}$, program execution starts from the address shown by the vector in the variable vector table
Address Match	FFFE8$_{16}$ to FFFEB$_{16}$	There is an address-matching interrupt enable bit
Single Step (Note)	FFFEC$_{16}$ to FFFEF$_{16}$	Do not use
Watchdog timer	FFFF0$_{16}$ to FFF3$_{16}$	
$\overline{\text{DBC}}$ (Note)	FFFF4$_{16}$ to FFFF7$_{16}$	Do not use
$\overline{\text{NMI}}$	FFFF8$_{16}$ to FFFFB$_{16}$	External interrupt by NMI pin
Reset	FFFFC$_{16}$ to FFFFF$_{16}$	

Note: Interrupts used for debugging purposes only

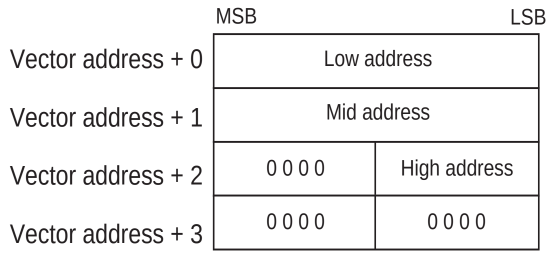


Figure 15: Format for specifying interrupt vector addresses

Table 8: Interrupt vectors (variable interrupt vector addresses)

Software interrupt number	Vector table addresses Address(L) to Address(H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked by I flag
Software interrupt number 4	+16 to +19	USB Suspend	
Software inturrupt number 6	+24 to +27	Resume	
Software inturrupt nubmer 7	+28 to +31	USB Start of Frame	
Software interrupt number 10	+40 to +43	Bus collision detection	
Software interrupt number 11	+44 to +47	DMA0	
Software interrupt number 12	+48 to +51	DMA1	
Software interrupt number 13	+52 to +55	Key input interrupt	
Software interrupt number 14	+56 to +59	A-D	
Software interrupt number 15	+60 to +63	UART2 transmit	
Software interrupt number 16	+64 to +67	UART2 receive	
Software interrupt number 17	+68 to +71	UART0 transmit	
Software interrupt number 18	+72 to +75	UART0 receive	
Software interrupt number 19	+76 to +79	UART1 transmit	
Software interrupt number 20	+80 to +83	UART1 receive	
Software interrupt number 21	+84 to +87	Timer A0	
Software interrupt number 22	+88 to +91	Timer A1	
Software interrupt number 23	+92 to +95	Timer A2	
Software interrupt number 24	+96 to +99	Timer A3	
Software interrupt number 25	+100 to +103	Timer A4	
Software interrupt number 26	+104 to +107	Timer B0	
Software interrupt number 27	+108 to +111	Timer B1	
Software interrupt number 28	+112 to +115	USB Reset	
Software interrupt number 29	+116 to +119	INT0	
Software interrupt number 30	+120 to +123	INT1	
Software interrupt number 31	+124 to +127	USB Function	
Software interrupt number 32 to Software interrupt number 63	+252 to +255	Software interrupt	Cannot be masked by I flag

Note 1:Address relative to address in interrupt table base address register (INTB)

(1) Interrupt control registers

Peripheral I/O interrupts have their own interrupt control registers. Table 9 shows the addresses of the interrupt control registers. Figure 16 shows the interrupt control registers.

The interrupt request bit is set by hardware to “0” when an interrupt request is received. The interrupt request bit can also be set by software to “0”. (Do not set to “1”.)

INT0 and INT1 are triggered by the edges of external inputs. The edge polarity is selected using the polarity select bit. (Other interrupts are described elsewhere.)

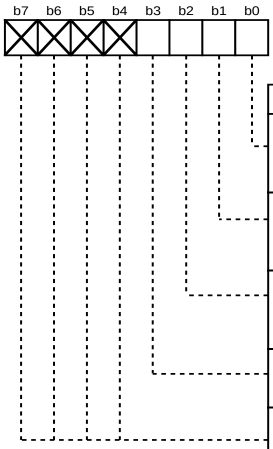
An interrupt must first be enabled before it can be used to cancel stop mode.

Table 9: Addresses in interrupt control register

Interrupt control register	Symbol name	Address	Interrupt control register	Symbol name	Address
Suspend-Interrupt	SUSPIC	0044_{16}	UART1 receive	S1RIC	0054_{16}
Resume interrupt	RSMIC	0046_{16}	Timer A0	TA0IC	0055_{16}
USB Start Of Frame	SOFIC	0047_{16}	Timer A1	TA1IC	0056_{16}
Bus collision detection	BCNIC	$004A_{16}$	Timer A2	TA2IC	0057_{16}
DMA0	DM0IC	$004B_{16}$	Timer A3	TA3IC	0058_{16}
DMA1	DM1IC	$004C_{16}$	Timer A4	TA4IC	0059_{16}
Key input interrupt	KUPIC	$004D_{16}$	Timer B0	TB0IC	$005A_{16}$
A-D	ADIC	$004E_{16}$	Timer B1	TB1IC	$005B_{16}$
UART2 transmit	S2TIC	$004F_{16}$	Reset interrupt	RSTIC	$005C_{16}$
UART2 receive	S2RIC	0050_{16}	INT0	INT0IC	$005D_{16}$
UART0 transmit	S0TIC	0051_{16}	INT1	INT1IC	$005E_{16}$
UART0 receive	S0RIC	0052_{16}	USB Function	USBFIC	$005F_{16}$
UART1 transmit	S1TIC	0053_{16}			

Interrupt control register

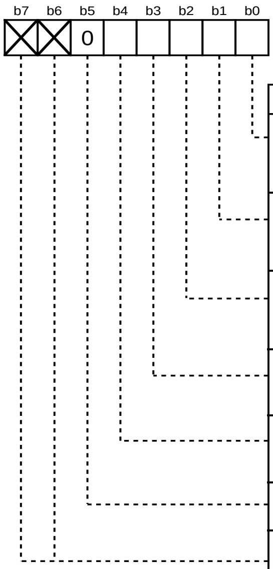
Symbol	Address	When reset
SUSPIC	0044_{16}	$XX00X000_2$
RSMIC	0046_{16}	$XXXXX000_2$
BCNIC	$004A_{16}$	$XXXXX000_2$
DMiIC(i=0, 1)	$004B_{16}$, $004C_{16}$	$XXXXX000_2$
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
SiTIC(i=0 to 2)	0051_{16}, 0053_{16}, $004F_{16}$	$XXXXX000_2$
SiRIC(i=0 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 $005B_{16}$	$XXXXX000_2$
RSTIC	$005C_{16}$	$XXXXX000_2$
USBFIC	$005F_{16}$	$XXXXX000_2$



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)
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

Note: This bit can only be reset (= 0), but cannot be set (= 1).

Symbol	Address	When reset
INTiIC(i=3)	$005D_{16}$, $005E_{16}$	$XX00X000_2$
SOFIC	0047_{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	○	○ (Note 2)
Reserved bit		Always set to “0”	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

Note 1: This bit can only be reset (= 0), but cannot be set (= 1).
Note 2: For SOFIC, (address 0047_{16}) , a "0" should always be written.

Figure 16: Interrupt control registers

(2) Interrupt priority

The order of priority when two or more interrupts are generated simultaneously is determined by both hardware and software.

The interrupt priority levels determined by hardware are reset > $\overline{\text{NMI}}$ > $\overline{\text{DBC}}$ > wacthdog timer > peripheral I/O interrupts > single-step > address matching interrupt.

The interrupt priority levels determined by software are set in the interrupt control registers.

Figure 17 shows the circuit that judges the interrupt hardware priority level. When two or more interrupts are generated simultaneously, the interrupt with the higher software priority is selected. However, if the interrupts have the same software priority level, the interrupt is selected according to the hardware priority set in the circuit.

The selected interrupt is accepted only when the priority level is higher than the processor interrupt priority level (IPL) in the flag register (FLG) and the interrupt enable flag (I flag) is “1”. Note that the reset, $\overline{\text{NMI}}$, $\overline{\text{DBC}}$, watchdog timer, single-step, address-match, BRK instruction, overflow, and undefined instruction interrupts are accepted regardless of the interrupt enable flag (I flag).

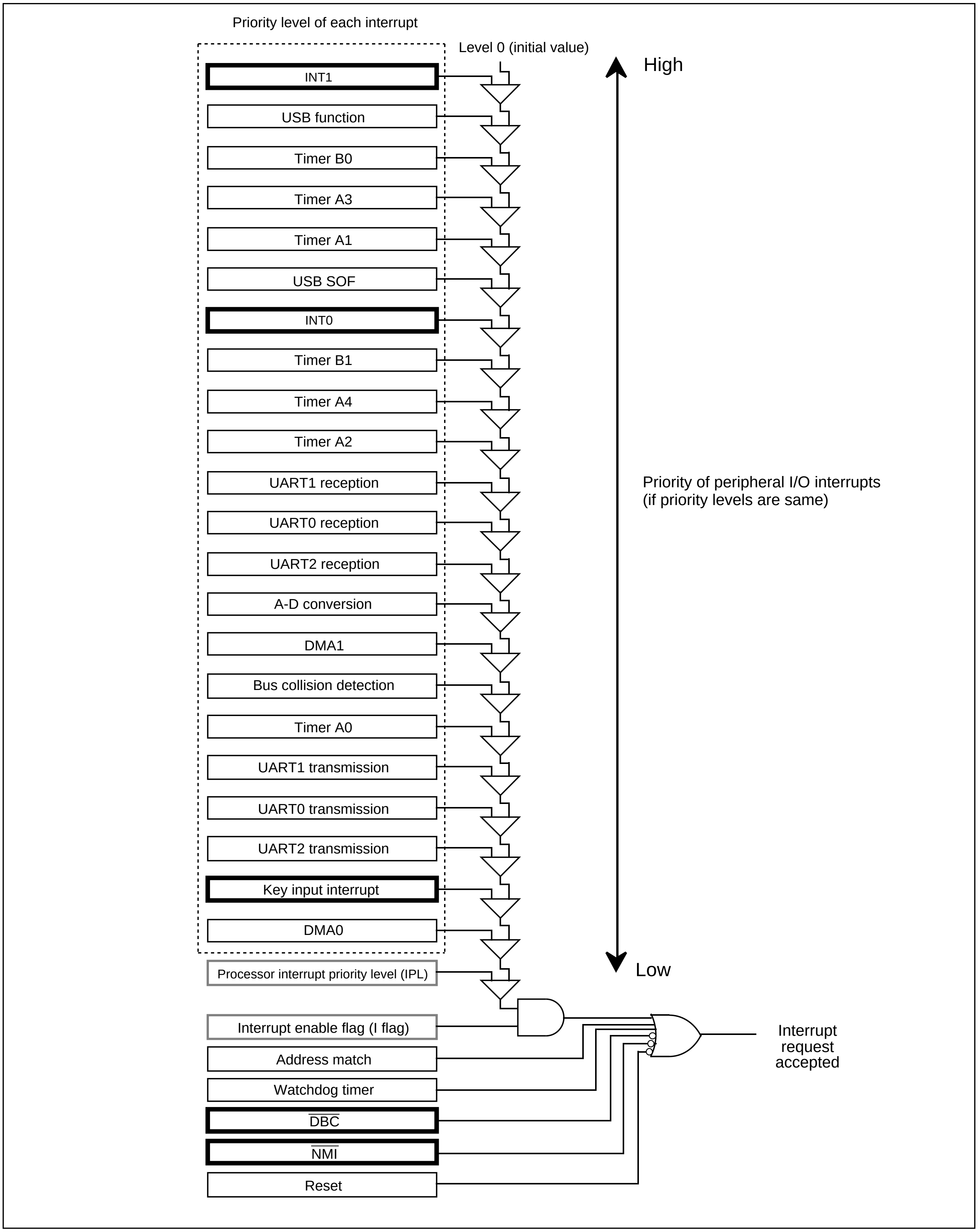


Figure 17: Interrupt resolution circuit

(3) Flag changes

When an interrupt request is received, the stack pointer select flag (U flag) changes to “0” and the flag register (FLG) and program counter (PC) are saved to the stack area indicated by the interrupt stack pointer (ISP). Thereafter, the interrupt enable flag (I flag) and debug flag (D flag) change to “0” and the processor interrupt priority level (IPL) at the flag register (FLG) is replaced by the priority level of the received interrupt. However, when interrupt requests are received for software interrupts 32 to 63, the flag register (FLG) and program counter (PC) are saved to the stack shown by the stack pointer select flag (U flag) at the time the interrupt was received. The stack pointer select flag (U flag) does not change. The value of the processor interrupt priority level (IPL) in the flag register (FLG) differs in the case of reset, $\overline{\mathrm{NMI}}$, $\overline{\mathrm{DBC}}$, watchdog timer, single-step, address-match, BRK instruction, overflow, and undefined instruction interrupts. Table 10 shows how the IPL changes when interrupt requests are received.

Table 10: Change of IPL state when interrupt request are accepted

Interrupt	Change of IPL
Reset	Level 0 (“000_2”) is set
$\overline{\mathrm{NMI}}$	Level 7 (“111_2”) is set
$\overline{\mathrm{DBC}}$	Does not change
Watchdog timer	Level 7 (“111_2”) is set
Single step	Does not change
Address match	Does not change
Software interrupt	Does not change

2.13 $\overline{\mathrm{NMI}}$ Interrupt

An $\overline{\mathrm{NMI}}$ interrupt is generated when the input to the P85/$\overline{\mathrm{NMI}}$ pin changes from “H” to “L”. The $\overline{\mathrm{NMI}}$ interrupt is a non-maskable external interrupt. The pin level can be checked in the port P85 register (bit 5 at address $03F0_{16}$).

This pin cannot be used as a normal port input.

Notes:

(1) When not intending to use the $\overline{\mathrm{NMI}}$ function, be sure to connect the $\overline{\mathrm{NMI}}$ pin to VCC. Because the $\overline{\mathrm{NMI}}$ interrupt is non-maskable, it cannot be disabled.

(2) When the $\overline{\mathrm{NMI}}$ pin input is “L”, do not set the microcomputer in stop mode or wait mode. The $\overline{\mathrm{NMI}}$ interrupt is triggered by the falling edge, so the “L” level does not need to be maintained longer than necessary.

2.14 Key-Input Interrupt

If the direction register of any of pin of Port0 or Port1 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. Figure 18 shows the block diagram of the key-input interrupt.

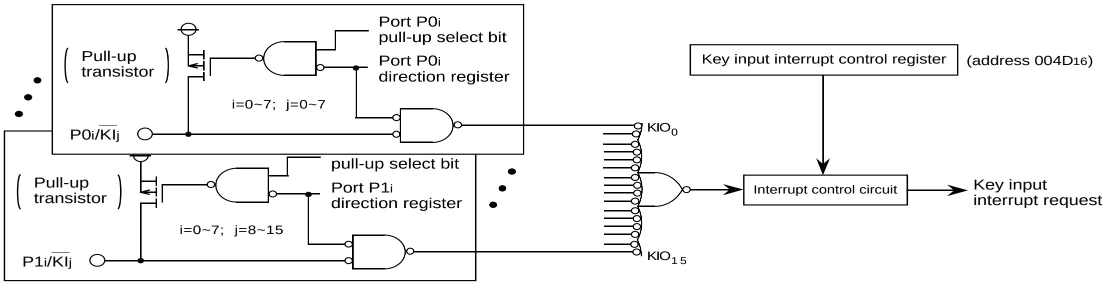


Figure 18: Block diagram of key input interrupt

(1) Enabling/disabling the key-input interrupt

The key-input interrupt can be enabled and disabled using the key-input interrupt register ($004D_{16}$). The key-input interrupt is affected by the interrupt priority level (IPL) and the interrupt enable flag (I flag).

(2) Occurrence timing of the key-input interrupt

With key-input interrupt acceptance enabled, ports P0 and P1, which are set to input, become key-input interrupt pins ($\overline{\text{KI0}}$ through $\overline{\text{KI15}}$). A key-input interrupt occurs when a falling edge is input to a key-input interrupt pin. At this moment, the level of other key-input interrupt pins must be “H”. No interrupt occurs when the level of any other key-input interrupt pins is “L”.

(3) How to determine a key-input interrupt

A key-input interrupt occurs when a falling edge is input to one of 16 pins, but each pin has the same vector address. Therefore, read the input level of ports P0 and P1 in the key-input interrupt routine to determine the interrupted pin.

(4) Registers related to the key-input interrupt

Figure 19 shows the memory map of key-input interrupt-related registers.

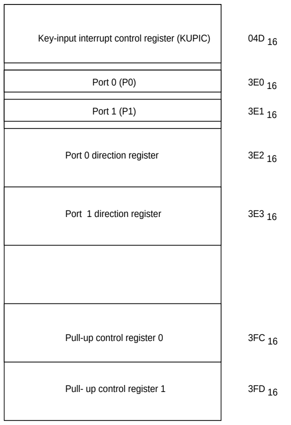
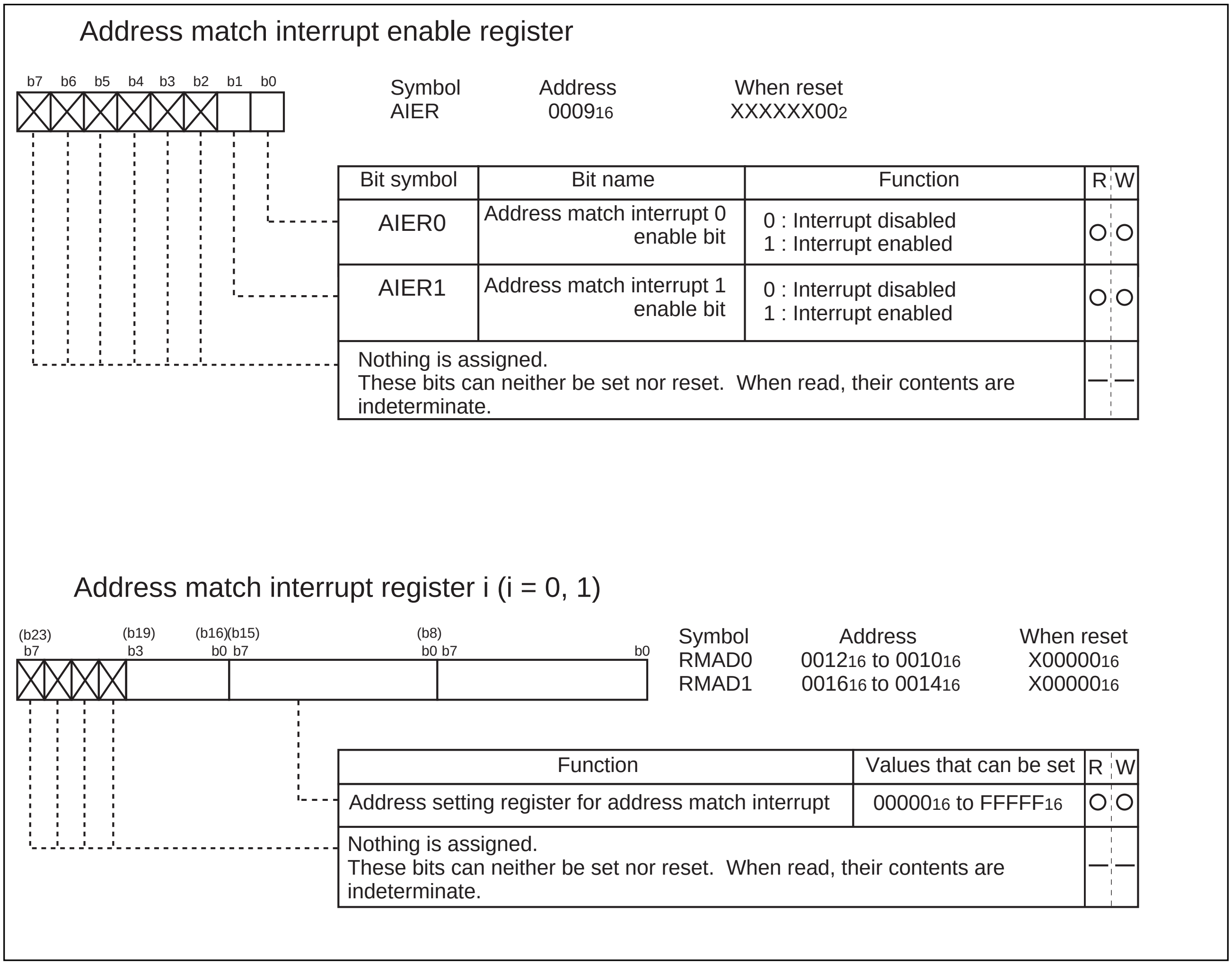

Key-input interrupt control register (KUPIC)
04D 16
Port 0 (P0)
3E0 16
Port 1 (P1)
3E1 16
Port 0 direction register
3E2 16
Port 1 direction register
3E3 16
Pull-up control register 0
3FC 16
Pull- up control register 1
3FD 16

Figure 19: Memory map of key-input interrupt-related registers

2.15 Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two 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).

Figure 20 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. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

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

Symbol	Address	When reset
RMAD0	0012$_{16}$ to 0010$_{16}$	X00000$_{16}$
RMAD1	0016$_{16}$ to 0014$_{16}$	X00000$_{16}$

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000$_{16}$ to FFFFF$_{16}$	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.		—	—

Figure 20: Address match interrupt-related registers

2.16 Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter that decrements using the clock derived by dividing the internal clock ϕ using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. Bit 7 of the watchdog timer control register (address $000F_{16}$) selects the prescaler division ratio (by 16 or 128). Table 11 shows the periodic table for the watchdog timer

.

Table 11: Watchdog timer periodic table (XIN = 10MHz)

CM06	CM17	CM16	Internal clock ϕ	WDC7	Period
0	0	0	10MHz	0	Approx. 52.4ms (Note)
				1	Approx. 419.2ms (Note)
0	0	1	5MHz	0	Approx. 104.9ms (Note)
				1	Approx. 838.8ms (Note)
0	1	0	2.5MHz	0	Approx. 209.7ms (Note)
				1	Approx. 1.68s (Note)
0	1	1	0.625MHz	0	Approx. 838.8ms (Note)
				1	Approx. 6.71s (Note)
1	Invalid	Invalid	1.25MHz	0	Approx. 419.2ms (Note)
				1	Approx. 3.35s (Note)

Note: Error is generated by the prescaler

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 $000E_{16}$).

Figure 21 shows the block diagram of the watchdog timer. Figure 22 shows the watchdog timer-related registers.

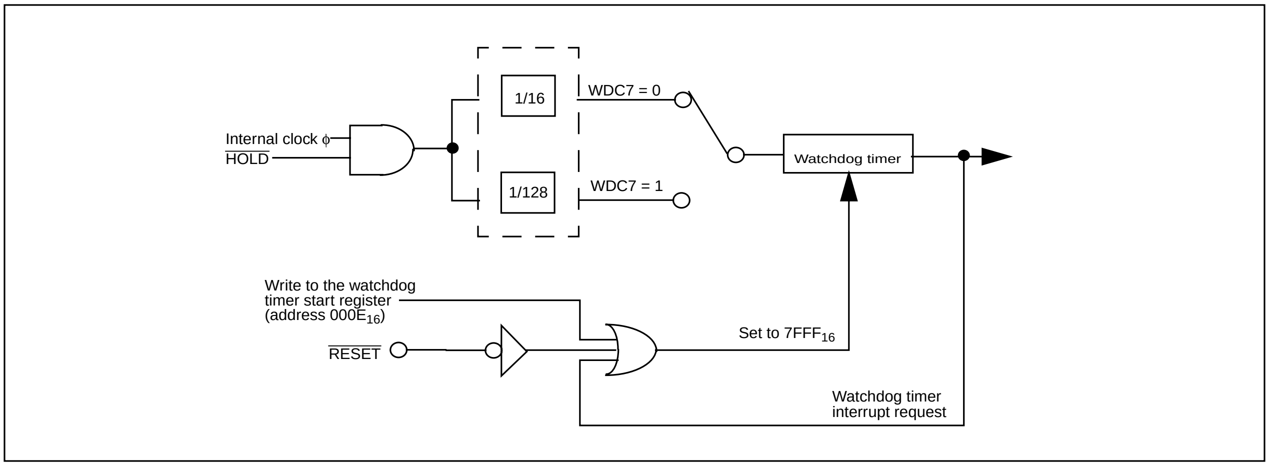
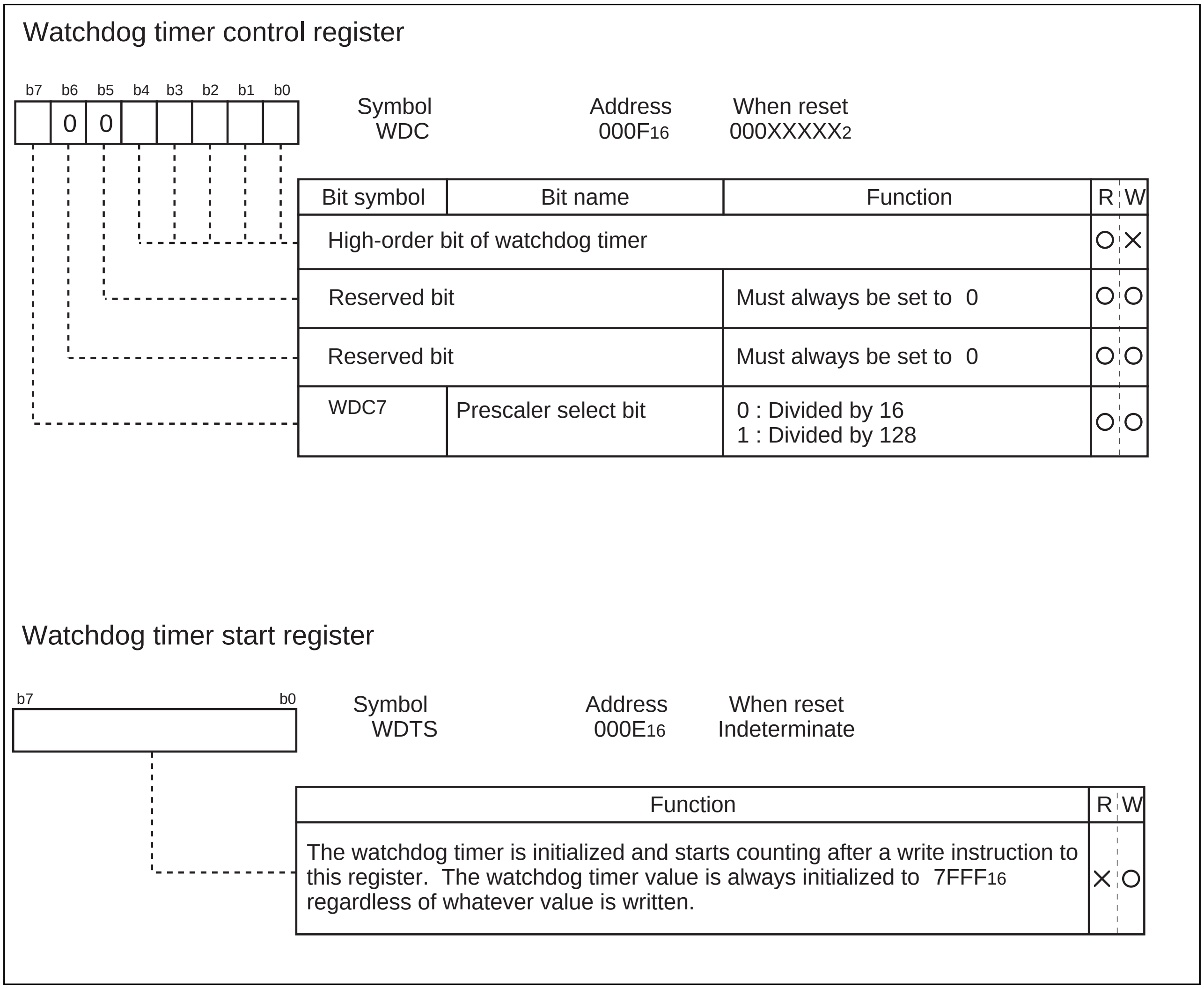

Internal clock ϕ
$\overline{\text{HOLD}}$
1/16
1/128
WDC7 = 0
WDC7 = 1
Watchdog timer
Write to the watchdog timer start register (address $000E_{16}$)
$\overline{\text{RESET}}$
Set to $7FFF_{16}$
Watchdog timer interrupt request

Figure 21: Block diagram of watchdog timer



Watchdog timer control register

Symbol WDC
Address $000F_{16}$
When reset $000XXXXX_2$

Bit symbol	Bit name	Function	R	W
High-order bit of watchdog timer			○	×
	Reserved bit	Must always be set to 0	○	○
	Reserved bit	Must always be set to 0	○	○
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	○	○

Watchdog timer start register

Symbol WDTS
Address $000E_{16}$
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 $7FFF_{16}$ regardless of whatever value is written.	×	○

Figure 22: Watchdog timer control and start registers

2.17 Frequency Synthesizer Circuit

The Frequency Synthesizer Circuit generates a 48MHz clock needed by the USB block and a clock f_{SYN} that are both a multiple of the external input reference clock f_{IN}. A block diagram of the circuit is shown in Figure 23.

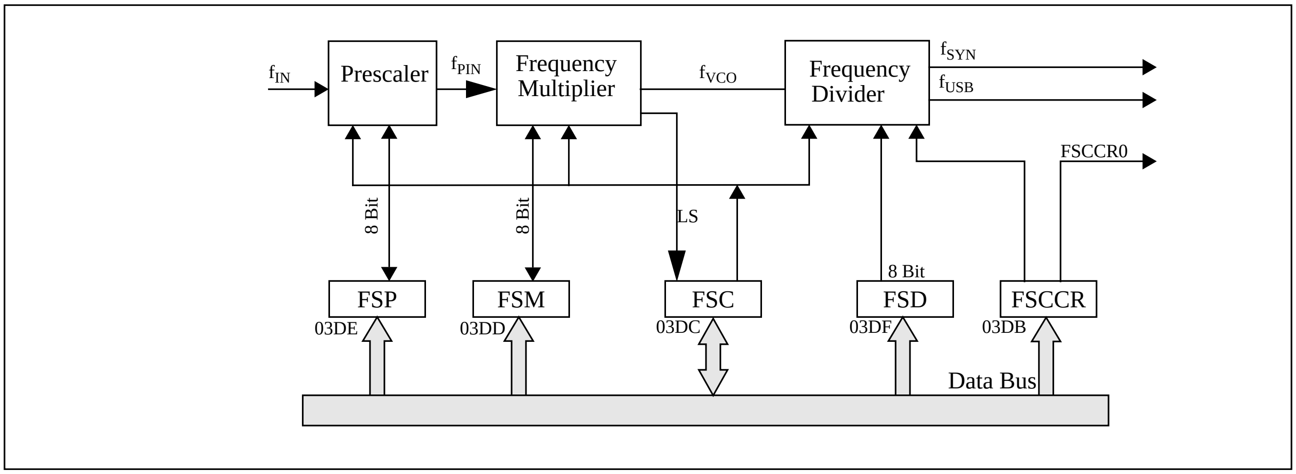


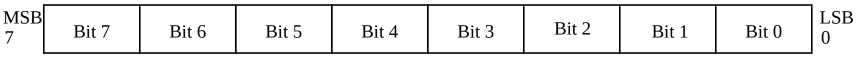
Figure 23: Frequency Synthesizer Circuit

The frequency synthesizer consists of a prescaler, frequency multiplier macro, a frequency divider macro, and five registers, namely FSP, FSM, FSC, FSD, and FSCCR. Clock f_{IN} is prescaled down using FSP to generate f_{PIN}. f_{PIN} is multiplied using FSM to generate an f_{VCO} clock which is then divided using FSD to produce the clock f_{SYN}. The f_{VCO} clock is optimized for 48 MHz operation and is buffered and sent out of the frequency synthesizer block as signal f_{USB}. This signal is used by the USB block.

(1) Prescaler

Clock f_{PIN} is a divided down version of clock f_{IN} (see Figure 24). The relationship between f_{PIN} and the clock input to the prescaler (f_{IN}) is as follows:

- $f_{PIN} = f_{IN} / 2(n+1)$ where n is a decimal number between 0 and 254.
  Setting FSP to 255 disables the prescaler and $f_{PIN} = f_{IN}$.



Address: $03DE_{16}$
Access: R/W
Reset: FF_{16}

f_{PIN}	FSP		f_{IN}
	Dec(n)	Hex(n)	
12 MHz	255	FF	12.00 MHz
1 MHz	5	05	12.00 MHz
2 MHz	2	02	12.00 MHz
3 MHz	1	01	12.00 MHz
6 MHz	0	00	12.00 MHz

$$f_{IN}/2(n+1) = f_{PIN}$$

Figure 24: Frequency Synthesizer Prescaler Register (FSP)

(2) Multiplier

Clock f_{VCO} is a multiplied up version of clock f_{PIN} (See Figure 25). The relationship between f_{VCO} and the clock input to the multiplier (f_{PIN}) from the prescaler is as follows:

- $f_{VCO} = f_{PIN} \times 2(n+1)$ where *n* is the decimal equivalent of the value loaded in FSM.
  Setting FSM to 255 disables the multiplier and $f_{VCO} = f_{PIN}$.

Note: *n* must be chosen such that f_{VCO} equals 48 MHz.

MSB 7								LSB 0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Address: $03DD_{16}$
Access: R/W
Reset: FF_{16}

f_{PIN}	FSM		f_{VCO}
	Dec(n)	Hex(n)	
320 kHz	74	4A	48.00 MHz
2 MHz	11	0B	48.00 MHz
4 MHz	5	05	48.00 MHz
6 MHz	3	03	48.00 MHz
12 MHz	1	01	48.00 MHz

$$f_{PIN} \times 2(n+1) = f_{VCO}$$

Figure 25: Frequency Synthesizer Multiply Register (FSM)

(3) Divider

Clock f_{SYN} is a divided down version of clock f_{VCO} (See Figure 26). The relationship between f_{SYN} and the clock input to the divider (f_{VCO}) from the multiplier is as follows:

- $f_{SYN} = f_{VCO} / 2(m+1)$ where m is the decimal equivalent of the value loaded in FSD.
  Setting FSD to 255 disables the divider and $f_{SYN} = f_{VCO}$.

MSB 7								LSB 0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Address: $03DF_{16}$
Access: R/W
Reset: FF_{16}

f_{VCO}	FSD		f_{SYN}
	Dec(m)	Hex(m)	
48.00 MHz	1	01	12.00 MHz
48.00 MHz	127	7F	187.50 KHz

$$f_{VCO}/2(m+1) = f_{SYN}$$

Figure 26: Frequency Synthesizer Divide Register (FSD)

The FSC0 bit in the FSC Control Register enables the frequency synthesizer block. When disabled (FSC0 = "0"), f_{VCO} is held at either a high or low state. When the frequency synthesizer control bit is active (FSC0 = "1"), a lock status (LS = "1") indicates that f_{SYN} and f_{VCO} are the correct frequency. The LS and FSCO control bits in the FSC Control register are shown in Figure 27.

When using the frequency synthesizer, a low-pass filter must be connected to the LPF pin.

Once the frequency synthesizer is enabled, a delay of 2-5ms is recommended before the output of the frequency synthesizer is used. This is done to allow the output to stabilize. It is also recommended that none of the registers be modified once the frequency synthesizer is enabled as it will cause the output to be temporarily (2-5ms) unstable. The CPU and USB clock sources are selecxted via the Frequency Synthesizer Clock Control register (FSCCR). See Figure 28

MSB 7							LSB 0
LS	CHG1	CHG0	Reserved	Reserved	VCO1	VCO0	FSE

Address: $03DC_{16}$
Access: R/W
Reset: 60_{16}

FSE Frequency Synthesizer Enable - Bit 0
0: Disabled
1: Enabled

VCO1,0 VCO Gain Control - Bits 2,1

Bit 2	Bit 1	
0	0:	Lowest Gain (recommended)
0	1:	Low Gain
1	0:	High Gain
1	1:	Highest Gain

Bits 4,3 Reserved (Read/Write “0”)

CHG1,0 LPF Current Control - Bits 6,5

Bit 6	Bit 5	
0	0:	Disabled
0	1:	Low Current
1	0:	Intermediate Current (recommended)
1	1:	High Current

LS Frequency Synthesizer Lock Status (Read Only; Write “0”) - Bit 7
0: Unlocked
1: Locked

Figure 27: Frequency Synthesizer Control Register (FSC)

MSB 7							LSB 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FSCCR1	FSCCR0

Address: $03DB_{16}$
Access: R/W
Reset: 00_{16}

FSCCR0 Clock Source Selection- Bit 0
0: Xin
1: f_{SYN}

FSCCR1 USB Clock Source Selection- Bit 1
0: $f_{USB} = f_{VCO}$
1: $f_{USB} = f_{Xin}$

Bits 7-2 Reserved (Read/Write “0”)

Figure 28: Frequency Synthesizer Clock Control Register (FSCCR)

2.18 Universal Serial Bus

The Universal Serial Bus (USB) has the following features:

- Complete USB Specification (version 1.0) Compatibility
- Error-handling capabilities
- FIFOs:
  - Endpoint 0: IN 32-byte OUT 32-byte
  - Endpoint 1: IN 128-byte OUT 128-byte
  - Endpoint 2: IN 32-byte OUT 32-byte
  - Endpoint 3: IN 32-byte OUT 32-byte
  - Endpoint 4: IN 32-byte OUT 32-byte
- Nine Endpoints - Control endpoint (Endpoint 0 - bidirectional) plus four IN and four OUT endpoints
- Complete Device Configuration
- Supports All Device Commands
- Supports Full-Speed Functions
- Support of All USB Transfer Types:
  - Isochronous
  - Bulk
  - Control
  - Interrupt
- Suspend/Resume Operation
- On-chip USB Transceiver with voltage converter
- Start-of-frame interrupt and output pin

USB Function Control Unit (USB FCU)

The implementation of the USB by this device is accomplished chiefly through the device's USB Function Control Unit (See Figure 29). The Function Control Unit's overall purpose is to handle the USB packet protocol layer. The Function Control Unit notifies the MCU that a valid token has been received. When this occurs, the data portion of the token is routed to the appropriate FIFO. The MCU transfers the data to, or from, the host by interacting with that endpoint's FIFO and CSR register.

The USB Function Control Unit is composed of five sections:

- Serial Interface Engine (SIE)
- Generic Function Interface (GFI)
- Serial Engine Interface Unit (SIU)
- Microcontroller Interface (MCI)
- USB Transceiver

Serial Interface Engine

The SIE interfaces to the USB serial data and handles deserialization/serialization of data, NRZI encoding decoding, clock extraction, CRC generation and checking, bit stuffing, and other specifications pertaining to the USB protocol such as handling inter-packet time-outs and packet ID (PID) decoding.

Generic Function Interface

The GFI handles the all USB standard requests from the host through the control endpoint (endpoint zero), handles Bulk, Isochronous and Interrupt transfers through endpoints 1-4. The GFI handles read pointer reversal for re-transmit the current data set; write pointer reversal for reception of the last data set; data toggle synchronization.

Serial Engine Interface Unit

The SIU block decodes the Address and Endpoint fields from the USB host.

Microcontroller Interface

The MCI block handles the Microcontroller interface and performs address decoding and synchronization of control signals.

USB Transceiver

The USB transceiver, designed to interface with the physical layer of the USB, is compliant with the USB Specification (version 1.0) for high speed devices. It consists of two 6-ohm drivers, a receiver, and schmitt triggers for single-ended receive signals.

The transceiver also includes a voltage converter. The voltage converter can supply 3.0 - 3.6V to the transmitter when the rest of the chip (CPU, USB FCU) operates at 4.15 - 5.25V. To enable the voltage converter, set bit 4 of the USB Control Register (USBC) to a “1”. To disable the voltage converter, set bit 4 of the USBC to a “0”. Refer to Section 5.5 “USB Transceiver” for more detailed information.

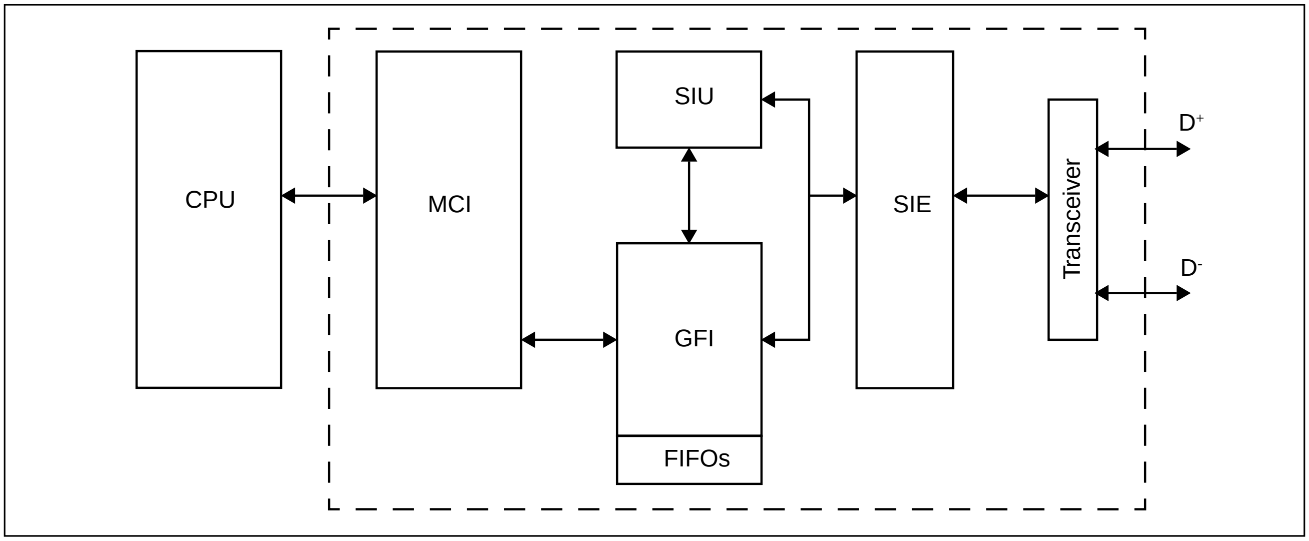


Figure 29: USB Function Control Unit Block Diagram

USB Interrupts

There are two types of USB interrupts in this device: the first type is the USB function (including overrun/underrun USB, reset, suspend and resume) interrupt, used to control the flow of data and USB power control; the second type is start-of-frame (SOF) interrupt, used to monitor the transfer of isochronous (ISO) data.

USB Function Interrupt

Endpoints 1-4 each have two interrupt status bits associated with them to control the data transfer or to report a STALL/UNDER_RUN/OVER_RUN condition. The EPx_OUT_INT bit is set when the USB FCU successfully receives a packet of data, or sets the FORCE_STALL bit, or the OVER_RUN bit of the Endpoint x OUT CSR. The EPx_IN_INT bit is set when the USB FCU successfully sends a packet of data, or sets the UNDER_RUN bit of the Endpoint x IN CSR. Endpoint 0 - the control endpoint - has one interrupt status bit associated with it to control the data transfer or report a STALL condition. The EP0_INT is set when the USB FCU successfully receives/sends a packet of data, or sets the SETUP_END bit, the FORCE_STALL bit, or clears the DATA_END bit in the Endpoint 0 IN CSR. Each endpoint interrupt is enabled by setting the corresponding bit in the USB Interrupt Enable Register 1 and 2. The USB Interrupt Status Register 1 and 2 are used to indicate pending interrupts for a given endpoint. The USB FCU sets the interrupt status bits. The CPU writes a “1” to clear the corresponding status bit. By writing back the same value it read, the CPU will clear all the existing interrupts. The CPU must read then write both status registers, writing status register 1 first and status register 2 second to guarantee proper operation.

The suspend interrupt status bit is set if a USB suspend signal is received. If the device is in suspend mode, the resume interrupt status bit is set when a USB resume signal is received. There is a single interrupt enable bit for both of suspend and resume interrupts (bit 7 of the interrupt enable register 2).

The USB reset interrupt status bit is set if a USB reset signal is received. When this bit is set, all USB internal registers is reset to their default values except this bit itself. This bit is cleared by the CPU writing a “0” to it. When the CPU detects a USB reset interrupt, it needs to re-initialize the USB block in order to accept packets from the host.

The Over/Underrun status bit is set (applicable to endpoints used for isochronous data transfer), when an overrun condition occurs in an endpoint (CPU is too slow to unload the data from the FIFO), or when an underrun condition occurs in an endpoint (CPU is too slow to load the data to the FIFO).

The USB Function Interrupt (sum of all individual function interrupts) is enabled by setting the corresponding bit in the Interrupt Control Register of the Interrupt Control Unit.

USB SOF Interrupt

The USB SOF (Start-Of-Frame) interrupt is used to control the transfer of isochronous data. The USB FCU generates a start-of-frame interrupt when a start-of-frame packet is received. The USB SOF interrupt is enabled by setting the corresponding bit in the Interrupt Control Register of the Interrupt Control Unit.

USB Endpoint FIFOs

The USB FCU has an IN (transmit) FIFO and an OUT (receive) FIFO for each endpoint. Both FIFOs support up to two separate data sets of variable size (except Endpoint 0), and provide the ability of back-to-back transmission and reception. Throughout this specification, the terms “IN FIFO” and “OUT FIFO” refer these FIFOs associated with the current endpoint.

In the event of a bad transmission/reception, the USB FCU handles all the read/write pointer reversal and data set management tasks when it is applicable.

IN (Transmit) FIFOs

The CPU/DMA writes data to the endpoint’s IN FIFO location specified by the FIFO write pointer, which automatically increments by "1" after a write.

Endpoint 0 IN FIFO Operation:

The CPU writes a “1” to the IN_PKT_RDY bit after it finishes writing a packet of data to the IN FIFO. The USB FCU clears the IN_PKT_RDY bit after the packet is successfully transmitted to the host (ACK is received from the host) or the SETUP_END bit of the IN CSR is set to a “1”.

Endpoint 1-4 IN FIFO Operation when AUTO_SET (bit 7 of IN CSR) = “0”:

MAXP > half of the IN FIFO size: The CPU writes a “1” to IN_PKT_RDY bit after the CPU/DMAC finishes writing a packet of data to the IN FIFO. The USB FCU clears TX_NOT_EMPTY bit after the packet is successfully transmitted to the host (ACK is received from the host). The CPU should only write data to the IN FIFO if the TX_NOT_EMPTY bit of the IN CSR is a “0”.

MAXP <= half of the IN FIFO size: The CPU writes a “1” to the IN_PKT_RDY bit after the CPU/DMAC finishes writing a packet of data to the IN FIFO. If only one packet of data is the FIFO TX_NOT_EMPTY bit gets set to a “1” and the IN_PKT_RDY bit gets clear to a ‘0’. If two packets of data in the FIFO, then the TX_NOT_EMPTY bit gets set to a “1” and the IN_PKT_RDY bit stays as a “1” (the FIFO can hold up to two data packets at the same time in this configuration, for back-to-back transmission). The CPU should only write data to the IN FIFO if the IN_PKT_RDY bit of the IN CSR is a “0”.

Endpoint 1-4 IN FIFO Operation when AUTO_SET (bit 7 of IN CSR) = “1”:

MAXP > half of the IN FIFO size: When the number of bytes of data equal to the MAXP (maximum packet size) is written to the IN FIFO by the CPU/DMAC, the USB FCU sets the TX_NOT_EMPTY bit to a “1” automatically. The USB FCU clears the TX_NOT_EMPTY bit after the packet is successfully transmitted to the host (ACK is received from the host). The CPU should only write data to the IN FIFO if the TX_NOT_EMPTY bit of the IN CSR is a “0”.

MAXP <= half of the IN FIFO size: When the number of bytes of data equal to the MAXP (maximum packet size) is written to the IN FIFO by the CPU/DMAC, the USB FCU sets the TX_NOT_EMPTY/IN_PKT_RDY bits to a “1” automatically depends on FIFO status. If only one packet of data is the FIFO TX_NOT_EMPTY bit gets set to a ‘1’ and the IN_PKT_RDY bit get clear to a “0”. If two packets of data in the FIFO then both the TX_NOT_EMPTY bit gets set to a “1” and the IN_PKT_RDY bit gets set to a “1” (the FIFO can hold up to two data packets at the same time in this configuration, for back-to-back transmission). The CPU should only write data to the IN FIFO if the IN_PKT_RDY bit of the IN CSR is a “0”.

A software or a hardware flush acts as if a packet is being successfully transmitted out to the host. If there is one packet in the IN FIFO, a flush causes the IN FIFO to be empty, if there are two packets in the IN FIFO, a flush causes the older packet to be flushed out from the IN FIFO. Flush updates the IN FIFO status (IN_PKT_RDY and TX_NOT_EMPTY bits).

The status of the endpoint 1-4 IN FIFO for both of the above cases, could be obtained from the IN CSR as shown in Table 12 .

Table 12: TA FIFO Status

IN_PKT_RDY	TX_NOT_EMPTY	TX FIFO Status
0	0	No data packet in TX FIFO
0	1	One data packet in TX FIFO if MAXP <= half of the FIFO size.
X	1	One data packet in TX FIFO if MAXP >= half of the FIFO size.
1	0	Invalid
1	1	Two data packets in TX FIFO if MAXP <= half of the FIFO size

Interrupt Endpoints:

Any endpoint can be used for interrupt transfers. For normal interrupt transfers, the interrupt transactions behave the same as bulk transactions, i.e., no special setting is required. The IN endpoints may also be used to communicate rate feedback information for certain types of isochronous functions. This is done by setting the INTPT bit in the IN CSR register of the corresponding endpoint. When the INTPT bit is set, the data toggle bits is changed after each packet is sent to the host without regard to the presence or type of handshakepacket.

The following outlines the operation sequence for an IN endpoint used to communicate rate feedback information:

1. Set MAXP > 1/2 of the endpoint’s FIFO size;
2. Set INTPT bit of the IN CSR;
3. Flush the old data in the FIFO;
4. Load interrupt status information and set IN_PKT_RDY bit in the IN CSR;
5. Repeat steps 3 & 4 for all subsequent interrupt status updates.

Out (Receive) FIFOs

The USB FCU writes data to the endpoint’s OUT FIFO location specified by the FIFO write pointer, which automatically increments by one after a write. When the USB FCU has successfully received a data packet, it sets the OUT_PKT_RDY bit to a “1” in the OUT CSR. The CPU/DMAC only reads data from the OUT FIFO if the OUT_PKT_RDY bit of the OUT CSR is a “1”.

Endpoint 0 OUT FIFO Operation:

The USB FCU sets the OUT_PKT_RDY bit to a ‘1’ after it has successfully received a packet of data from the host. The CPU writes a “0” to the OUT_PKT_RDY bit after the packet of data is unloaded from the OUT FIFO by the CPU.

Endpoint 1-4 OUT FIFO Operation when AUTO_CLR (bit 7 of OUT CSR) = “0”:

MAXP > half of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit to a “1” after it has successfully received a packet of data from the host. The CPU writes a “0” to the OUT_PKT_RDY bit after the packet of data is unloaded from the OUT FIFO by the CPU/DMAC.

MAXP <= half of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit to a “1” after it has successfully received a packet of data from the host. The CPU writes a “0” to the OUT_PKT_RDY bit after the packet of data is unloaded from the OUT FIFO by the CPU/DMAC. In this configuration, the FIFO can store up to two data packets at the same time, for back-to-back reception. Therefore, the OUT_PKT_RDY bit may remain set after the CPU writes a “0” to it if there is another packet in the OUT FIFO.

Endpoint 1-4 OUT FIFO Operation when AUTO_CLR (bit 7 of OUT CSR) = “1”:

MAXP > half of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit to a “1” after it has successfully received a packet of data from the host. The USB FCU clears the OUT_PKT_RDY bit to a ‘0’ automatically when the number of bytes of data equal to the MAXP (maximum packet size) is unloaded from the OUT FIFO by the CPU/DMAC.

MAXP <= half of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit to a “1” after it has successfully received a packet of data from the host. The USB FCU clears the OUT_PKT_RDY bit to a “0” automatically when the number of bytes of data equal to the MAXP (maximum packet size) is unloaded from the OUT FIFO by the CPU/DMAC. In this configuration, the FIFO can hold up to two data packets at the same time, for back-to-back reception. Therefore, the OUT_PKT_RDY bit may remain set after one packet (size equal to MAXP) of data is unloaded if there is another packet in the OUT FIFO.

A software flush acts as if a packet is being unloaded from the OUT FIFO. If there is one packet in the OUT FIFO, a flush causes the OUT FIFO to be empty, if there are two packets in the OUT FIFO, a flush causes the older packet to be flushed out from the OUT FIFO.

USB Special Function Registers

The MCU controls USB operation through the use of special function registers (SFR). This section describes in detail each USB related SFR. Some USB special function registers have a mix of read/write, read only, and write only register bits. Additionally, the bits may be configured to allow the user to write only a “0” or a “1” to individual bits. When accessing these registers, writing a “0” to a register that can only be set to a “1” by the CPU has no effect on that register bit. Each figure and description of the special function registers details this operation.

USB attach / detach register

The USB attach / detach register is shown in Figure 30. The register is used to detach the USB function from a USB host without physically disconnecting the USB cable. The register is enabled in this special mode by setting PORT83_SECOND high, this forces Port 8_3 to operate as a pull-up for D+(it tri-states the port output driver and forces a “1” if Port 8_3 is read).When the attach/detach bit is high, an attach is detected by the host; when set low, a detach is registered by the host. A 1.5K ohm pull-up resistor must be added externally from port 8_3 to D+ to enable this mode. This mode is bypassed when EXTCAP is used to pull up D+ via a 1.5 K ohm resistor.

MSB 7							LSB 0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Attach	P83_2nd

Address: 001F$_{16}$
Access: R/W
Reset: 00$_{16}$

PORT83_SECOND:
- 0 Normal mode for Port 8_3
- 1 Forces Port 8_3 to operate as pull-up for D+ enable special register.

ATTACH / DETACH (enabled when bit 0 is set to 1):
- 0 Will cause the host to eventually detect a detach.
- 1 Will cause host to detect attach.

Bit 7:2 Reserved

Figure 30: USB attach / detach register

The USB Control Register**,** shown in Figure 31**,** is used to control the USB FCU (for Microsoft Legacy Application, please see Addendum). This register is not reset by a USB reset signaling. After the USB is enabled (USBC7 set to "1"), a minimum delay of 250ns (three 12 MHz clock periods) is needed before performing any other USB register read/write operations.

MSB 7	6	5	4	3	2	1	LSB 0
USBC7	USBC6	USBC5	USBC4	USBC3	Reserved	Reserved	Reserved

Address: 000C$_{16}$
Access: R/W
Reset: 00$_{16}$

Bit 2:0	Reserved
USBC3	Transceiver Voltage Converter High/Low Current Mode Selection Bit (bit 3) 0: High current mode, for USB normal operation 1: Low current mode, for USB suspend operation
USBC4	USB Transceiver Voltage Converter Enable Bit (bit 4) 0: USB transceiver voltage converter disabled 1: USB transceiver voltage converter enabled
USBC5	USB Clock Enable Bit (bit 5) 0: 48 MHz clock to the USB block is disabled. 1: 48 MHz clock to the USB block is enabled.
USBC6	USB $\overline{\text{SOF}}$ Port Select Bit (bit 6) 0: USB $\overline{\text{SOF}}$ output is disabled. P8$_6$ is used as GPIO pin. 1: USB $\overline{\text{SOF}}$ output is enabled
USBC7	USB Enable Bit (bit 7) 0: USB block is disabled, all USB internal registers are held at their default values. 1: USB block is enabled

Figure 31: USB Control Register (for Microsoft Legacy Application, see Addendum)

The USB Function Address Register, shown in Figure 32, maintains the 7-bit USB address assigned by the host. The USB FCU uses this register value to decode USB token packet addresses. At reset, when the device is not yet configured, the value is 00$_{16}$.

MSB 7	6	5	4	3	2	1	LSB 0
Reserved	FUNAD6	FUNAD5	FUNAD4	FUNAD3	FUNAD2	FUNAD1	FUNAD0

Address: 0300$_{16}$
Access: R/W
Reset: 00$_{16}$

FUNAD6:0	7-bit programmable Function Address (bits 6-0)
Bit 7	Reserved (Read/Write "0")

Figure 32: USB Function Address Register

The USB Power Management Register, shown in Figure 33, is used for power management in the USB FCU.

MSB 7	6	5	4	3	2	1	LSB 0
Reserved	Reserved	Reserved	Reserved	Reserved	WAKEUP	RESUME	SUSPEND

Address: 0301$_{16}$
Access: R/W
Reset: 00$_{16}$

SUSPEND	USB Suspend Detection Flag (bit 0) (Write "0" only or Read) 0: No USB suspend signal detected 1: USB suspend signal detected
RESUME	USB Resume Detection Flag (bit 1) (Write "0" only or Read) 0: No USB resume signal detected 1: USB resume signal detected
WAKEUP	USB Remote Wake-up Bit (bit 2) 0: End remote resume signaling 1: Remote resume signaling (If SUSPEND = "1")
Bit7:3	Reserved (Read/Write "0")

Figure 33: USB Power Management Register

USB Suspend Detection Flag

When the USB FCU receives a USB suspend signaling, it sets the SUSPEND bit and generates an interrupt. The CPU writes a “0” to clear this bit when the device is resumed by the host (resume interrupt is generated and Resume Detection Flag is set) or remote wake-up by itself (The CPU writes a “1” to Remote Wake-up Bit).

USB Resume Detection Flag

When the USB FCU is in suspend mode and receives a USB resume signaling, it sets the RESUME bit, and generates an interrupt. The CPU writes a “0” to clear this bit.

USB Remote Wake-up Bit

The CPU writes a “1” to the WAKEUP bit for remote wake-up. While this bit is set, and the USB FCU is in suspend mode, it generates a resume signaling to the host. The CPU must keep this bit set for a minimum of 10ms and a maximum of 15ms before writing a “0” to this bit.

The USB FCU is able to generate a USB function interrupt as discussed in “USB Interrupt” section .

USB Interrupt Status Registers**,** shown in Figures 34 and 35, are used to indicate the condition that caused a USB function interrupt to the CPU. A “1” indicates the corresponding condition caused a USB function interrupt. The USB Interrupt Status Registers can be cleared by writing back to the register the same value that was read. To ensure proper operation, the CPU reads both USB interrupt status registers, then write back the same values it read to these two registers for clearing the status bits. The CPU must write the USB Interrupt Status Register 1 first, then the USB Interrupt Status Register 2. The registers cannot be cleared by writing a “0” to the bits that are a “1”.

MSB 7									LSB 0
	INTST7	INTST6	INTST5	INTST4	INTST3	INTST2	Reserved	INTST0	

Address: 0302_{16}
Access: R/W
Reset: 00_{16}

INTST0	USB Endpoint 0 Interrupt Status Flag (bit 0)
Bit 1	Reserved (Read/Write “0”)
INTST2	USB Endpoint 1 IN Interrupt Status Flag (bit 2)
INTST3	USB Endpoint 1 OUT Interrupt Status Flag (bit 3)
INTST4	USB Endpoint 2 IN Interrupt Status Flag (bit 4)
INTST5	USB Endpoint 2 OUT Interrupt Status Flag (bit 5)
INTST6	USB Endpoint 3 IN Interrupt Status Flag (bit 6)
INTST7	USB Endpoint 3 OUT Interrupt Status Flag (bit 7)

0: No interrupt request issued
1: Interrupt request issued

Figure 34: USB Interrupt Status Register 1

INTST0 is set to a “1” by the USB FCU if (in Endpoint 0 CSR):

- Successfully receives a packet of data
- Successfully sends a packet of data
- EP0CSR3 (DATA_END) bit is cleared
- EP0CSR4 (FORCE_STALL) bit is set
- EP0CSR5 (SETUP_END) bit is set

INTST2, INTST4, INTST6 or INTST8 is set to a “1” by the USB FCU if (in Endpoint x IN CSR):

- Successfully sends a packet of data
- INXCSR1 (UNDER_RUN) bit is set

INTST3, INTST5, INTST7 or INTST9 is set to a “1” by the USB FCU if (in Endpoint xOUT CSR):

- Successfully receives a packet of data
- OUTXCSR1 (OVER_RUN) bit is set
- OUTXCSR4 (FORCE_STALL) bit is set

MSB 7							LSB 0
INTST15	INTST14	INTST13	INTST12	Reserved	Reserved	INTST9	INTST8

Address: 0303_{16}
Access: R/W
Reset: 00_{16}

INTST8	USB Endpoint 4 In Interrupt Status Flag (bit 0)
INTST9	USB Endpoint 4 Out Interrupt Status Flag (bit 1)
Bit 3:2	Reserved (Read/Write “0”)
INTST12	USB Overrun/Underrun Interrupt Status Flag (bit 4)
INTST13	USB Reset Interrupt Status Flag (bit 5)
INTST14	USB Resume Signaling Interrupt Status Flag (bit 6)
INTST15	USB Suspend Signaling Interrupt Status Flag (bit 7)

0: No interrupt request issued
1: Interrupt request issued

Figure 35: USB Interrupt Status Register 2

INTST12 is set to a “1” by the USB FCU if an overrun or underrun condition occurs in any of the endpoints.

INTST13 is set to a “1” by the USB FCU if a USB reset signaling from the host is received. All other USB internal registers is reset to their default values.

INTST14 is set to a “1” by the USB FCU if a USB resume signaling is received from the host.

INTST15 is set to a “1” by the USB FCU if a USB suspend signaling is received from the host.

The USB Interrupt Enable Register**s,** shown in Figure 36 and Figure 37, are used to enable the corresponding interrupt status conditions, which can generate a USB function interrupt. If the bit to a corresponding interrupt condition is “0”, that condition does not generate a USB function interrupt. If the bit is a “1”, that condition can generate a USB function interrupt. Upon reset, all USB interrupt status conditions are enabled except bit 7 of USB Interrupt Enable Register 2 (that is, suspend and resume interrupt is disabled).

MSB 7							LSB 0
INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	Reserved	INTEN0

Address: 0304_{16}
Access: R/W
Reset: FF_{16}

INTEN0	USB Endpoint 0 In Interrupt Enable Bit (bit 0)
Bit 1	Reserved (Read/Write “0”)
INTEN2	USB Endpoint 1 IN Interrupt Enable Bit (bit 2)
INTEN3	USB Endpoint 1 OUT Interrupt Enable Bit (bit 3)
INTEN4	USB Endpoint 2 IN Interrupt Enable Bit (bit 4)
INTEN5	USB Endpoint 2 OUT Interrupt Enable Bit (bit 5)
INTEN6	USB Endpoint 3 IN Interrupt Enable Bit (bit 6)
INTEN7	USB Endpoint 3 OUT Interrupt Enable Bit (bit 7)

0: Interrupt disabled
1: Interrupt enabled

Figure 36: USB Interrupt Enable Register 1

MSB 7							LSB 0
Reserved	Reserved	Reserved	INTEN12	Reserved	Reserved	INTEN9	INTEN8

Address: 0305_{16}
Access: R/W
Reset: 33_{16}

INTEN8	USB Endpoint 4 IN Interrupt Enable Bit (bit 0)
INTEN9	USB Endpoint 4 OUT Interrupt Enable Bit (bit 1)
Bit 3:2	Reserved (Read/Write “0”)
INTEN12	USB Overrun/Underrun Interrupt Enable Bit (bit 4)
Bit 5	Reserved
Bit 6	Reserved
Bit 7	Reserved

0: Interrupt disabled
1: Interrupt enabled

Figure 37: USB Interrupt Enable Register 2

The USB Frame Number Low Register, shown in Figure 38, contains the lower 8 bits of the 11-bit frame number received from the host. The USB Frame Number High Register, shown in Figure 39 contains the upper 3 bits of the 11-bit frame number received from the host.

MSB 7							LSB 0
FN7	FN6	FN5	FN4	FN3	FN2	FN1	FN0

Address: 0306_{16}
Access: R
Reset: 00_{16}

FN7:0	Lower 8 bits of the 11-bit frame number issued with a SOF token

Figure 38: USB Frame Number Low Register

MSB 7							LSB 0
Reserved	Reserved	Reserved	Reserved	Reserved	FN10	FN9	FN8

Address: 0307_{16}
Access: R
Reset: 00_{16}

FN10:8	Upper 3 bits of the 11-bit frame number issued with a SOF token
Bits 7:3	Reserved (Read “0”)

Figure 39: USB Frame Number High Register

The USB ISO Control Register, shown in Figure 40, contains two global bits, ISO_UPD and AUTO_FL for endpoints 1-4 regarding the isochronous data transfer.

If ISO_UPD = “0”, a data packet in an endpoint’s IN FIFO is always ‘ready to transmit’ upon receiving the next IN_TOKEN from the host (with matched address & endpoint number). If ISO_UPD = “1” and the ISO bit of the corresponding endpoint’s IN CSR is set, then the internal ‘ready to transmit’ signal to the transmit control logic is delayed until the next SOF. In this way, the data loaded in frame n is transmitted out in frame n+1. The ISO_UPD bit is a global bit for endpoints 1 to 4, and works with isochronous pipes only.

If AUTO_FL = “1”, ISO_UPD = "1", and a particular IN endpoint’s ISO bit is set, then at the time the USB FCU detects a SOF packet, if the corresponding IN endpoint’s IN_PKT_RDY = “1”, the USB FCU automatically flushes the oldest packet from the IN FIFO. In this case, IN_PKT_RDY = “1” indicates that two data packets are in the IN FIFO. Since, for ISO transfer, double buffering is a requirement, MAXP must be set to less than or equal to 1/2 of the FIFO size.

MSB 7								LSB 0
	ISO_UPD	AUTO_FL	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Address: 0308_{16}
Access: R/W
Reset: 00_{16}

Bits 5:0	Reserved (Read/Write “0”)
AUTO_FL	AUTO_FLUSH Bit (bit 6) 0: Hardware auto FIFO flush disabled 1: Hardware auto FIFO flush enabled
ISO_UPD	ISO_UPDATE Bit (bit 7) 0: ISO_UPDATE disabled 1: ISO_UPDATE enabled

Figure 40: **USB ISO Control Register**

The USB DMAx Request Registers, shown in Figures 41 and 42, are used to select the USB Endpoint x FIFO read/write request as DMAC channel 0 or channel 1 request source. Figure 43 shows the USB enpoint enable register. The USB DMA0 (DMA1) Request Register has only one bit set at any given time. If multiple bits are set, then no request is selected.

MSB 7								LSB 0
	DMA0R7	DMA0R6	DMA0R5	DMA0R4	DMA0R3	DMA0R2	DMA0R1	DMA0R0

Address: 0309_{16}
Access: R/W
Reset: 00_{16}

DMA0R0	Endpoint 1 IN FIFO Write Request Selection bit (bit 0)
DMA0R1	Endpoint 2 IN FIFO Write Request Selection bit (bit 1)
DMA0R2	Endpoint 3 IN FIFO Write Request Selection bit (bit 2)
DMA0R3	Endpoint 4 IN FIFO Write Request Selection bit (bit 3)
DMA0R4	Endpoint 1 OUT FIFO Read Request Selection bit (bit 4)
DMA0R5	Endpoint 2 OUT FIFO Read Request Selection bit (bit 5)
DMA0R6	Endpoint 3 OUT FIFO Read Request Selection bit (bit 6)
DMA0R7	Endpoint 4 OUT FIFO Read Request Selection bit (bit 7)

0: Not selected
1: Selected

Figure 41: **USB DMA0 Request Register**

MSB 7								LSB 0
	DMA1R7	DMA1R6	DMA1R5	DMA1R4	DMA1R3	DMA1R2	DMA1R1	DMA1R0

Address: $030A_{16}$
Access: R/W
Reset: 00_{16}

DMA1R0	Endpoint 1 IN FIFO Write Request Selection bit (bit 0)
DMA1R1	Endpoint 2 IN FIFO Write Request Selection bit (bit 1)
DMA1R2	Endpoint 3 IN FIFO Write Request Selection bit (bit 2)
DMA1R3	Endpoint 4 IN FIFO Write Request Selection bit (bit 3)
DMA1R4	Endpoint 1 OUT FIFO Read Request Selection bit (bit 4)
DMA1R5	Endpoint 2 OUT FIFO Read Request Selection bit (bit 5)
DMA1R6	Endpoint 3 OUT FIFO Read Request Selection bit (bit 6)
DMA1R7	Endpoint 4 OUT FIFO Read Request Selection bit (bit 7)

0: Not selected
1: Selected

Figure 42: **USB DMA1 Request Register**

MSB 7								LSB 0
	EP4_IN	EP4_OUT	EP3_IN	EP3_OUT	EP2_IN	EP2_OUT	EP1_IN	EP1_OUT

Address: $030B_{16}$
Access: R/W
Reset: ff_{16}

EP1_OUT	Endpoint 1 IN FIFO Write Request Selection bit (bit 0)
EP1_IN	Endpoint 2 IN FIFO Write Request Selection bit (bit 1)
EP2_OUT	Endpoint 3 IN FIFO Write Request Selection bit (bit 2)
EP2_IN	Endpoint 4 IN FIFO Write Request Selection bit (bit 3)
EP3_OUT	Endpoint 1 OUT FIFO Read Request Selection bit (bit 4)
EP3_IN	Endpoint 2 OUT FIFO Read Request Selection bit (bit 5)
EP4_OUT	Endpoint 3 OUT FIFO Read Request Selection bit (bit 6)
EP4-IN	Endpoint 4 OUT FIFO Read Request Selection bit (bit 7)

0: Not selected
1: Selected

Figure 43: **USB Endpoint enable register**

The **Endpoint 0 CSR** (Control & Status Register), shown in Figure 44, contains the control and status information of Endpoint 0.

MSB 7								LSB 0
EP0CSR7	EP0CSR6	EP0CSR5	EP0CSR4	EP0CSR3	EP0CSR2	EP0CSR1	EP0CSR0	

Address: 0311_{16}
Access: R/W
Reset 00_{16}

EP0CSR0 OUT_PKT_RDY Flag (bit 0) (Read Only - Write “ 0”)
0: Out packet is not ready
1: Out packet is ready

EP0CSR1 IN_PKT_RDY Bit (bit 1) (Write “ 1” only or Read)
0: In packet is not ready
1: In packet is ready

EP0CSR2 SEND_STALL Bit (bit 2) (Write “ 1” only or Read)
0: No action
1: Stall Endpoint 0 by the CPU

EP0CSR3 DATA_END Bit (bit 3) (Write “ 1” only or Read)
0: No action
1: Last packet of data transferred from/to the FIFO

EP0CSR4 FORCE_STALL Flag (bit 4) (Write “ 0” only or Read)
0: No action
1: Stall Endpoint 0 by the USB FCU

EP0CSR5 SETUP_END flag (bit 5) (Read Only - Write “ 0”)
0: No action
1: Control transfer ended before the specific length of data is transferred during the data phase

EP0CSR6 SERVICED_OUT_PKT_RDY Bit (bit 6) (Write Only - Read “ 0”)
0: No change
1: Clear the OUT_PKT_RDY bit (EP0CSR0)

EP0CSR7 SERVICED_SETUP_END Bit (bit 7) (Write Only - Read “ 0”)
0: No change
1: Clear the STUP_END bit (EP0CSR5)

Figure 44: USB Endpoint 0 CSR

EP0CSR0 (OUT_PKT_RDY):

The USB FCU sets this bit to a “1” upon receiving a valid SETUP/OUT token from the host. The CPU clears this bit after unloading the FIFO, by way of writing a “1” to EP0CSR6. The CPU does not clear the OUT_PKT_RDY bit before finishes decoding the host request. If EP0CSR2 (SEND_STALL) needs to be set - the CPU decodes an invalid or unsupported request - the setting EP0CSR6 = “1” & EP0CSR2 = “1” is done in a same CPU write.

EP0CSR1 (IN_PKT_RDY):

The CPU writes a “1” to this bit after it finishes writing a packet of data to the endpoint 0 FIFO. The USB FCU clears this bit after the packet is successfully transmitted to the host, or the EP0CSR5 (SETUP_END) bit is set.

EP0CSR2 (SEND_STALL):

The CPU writes a “1” to this bit if it decodes an invalid or unsupported standard device request from the host. The USB FCU returns a STALL handshake for all subsequent IN/OUT transactions (during control transfer data or status stages) while this bit is set. The CPU writes a “0” to clear this bit.

EP0CSR3 (DATA_END):

For control transfers, the CPU writes a “1” to this bit when it writes (IN data phase) or reads (OUT data phase) the last packet of data to or from the FIFO. This bit indicates to the USB FCU that the specific amount of data in the setup phase is transferred. The USB FCU advances to the status phase once this bit is set. When the status phase completes, the USB FCU clears this bit. When this bit is set to a “1”, and the host again requests or sends more data, the USB FCU returns a STALL handshake.

EP0CSR4 (FORCE_STALL):

The USB FCU sets this bit to a “1” if the host sends out a larger data packet than the MAXP size, or if during a data stage a command pipe is sent more data or is requested to return more data than was indicated in the setup stage (see description for EP0CSR3). The USB FCU returns a STALL handshake for all subsequent IN/OUT transactions (during data or status stages) while this bit is set. The CPU writes a “0” to clear this bit.

EP0CSR5 (SETUP_END):

The USB FCU sets this bit to a “1” if a control transfer has ended before the specific length of data is transferred during the data phase. The CPU clears this bit by writing a “1” to EP0CSR7. Once the CPU sees the SETUP_END bit set, it stops accessing the FIFO to service the previous setup transaction. If OUT_PKT_RDY is set at the same time SETUP_END is set, it indicates the previous setup transaction ended, and a new SETUP token is in the FIFO.

EP0CSR6 and EP0CSR7:

These bits are used to clear EP0CSR0 and EP0CSR5 respectively. Writing a “1” to these bits clears the corresponding register bit.

The USB Endpoint 0 MAXP, shown in Figure 45, indicates the maximum packet size (MAXP) of Endpoint 0 IN/OUT packet. The default value for Endpoint 0 MAXP is 8 bytes. The CPU can change this value, as negotiated with the host controller through the SET_DESCRIPTOR command.

MSB 7							LSB 0
Reserved	Reserved	EP0MXP5	EP0MXP4	EP0MXP3	EP0MXP2	EP0MXP1	EP0MXP0

EP0MXP4:0 Maximum packet size (MAXP) of Endpoint 0 IN/OUT packet.

Address: 0313_{16}
Access: R/W
Reset: 08_{16}

Figure 45: **USB Endpoint 0 MAXP**

The USB Endpoint 0 OUT WRT CNT register, shown in Figure 46, contains the number of bytes of the current data set in the OUT FIFO. The USB FCU sets the value in the Write Count Register after having successfully received a packet of data from the host. The CPU reads the register to determine the number of bytes to be read from the FIFO.

MSB 7							LSB 0
Reserved	Reserved	Reserved	W_CNT4	W_CNT3	W_CNT2	W_CNT1	W_CNT0

W_CNT4:0 Receive Byte Count.

Address: 0315_{16}
Access: R
Reset: 00_{16}

Figure 46: **USB Endpoint 0 OUT WRT CNT**

The USB Endpoint x IN CSR (Control & Status Register), shown in Figure 47, contains control and status information of the respective IN endpoint 1-4.

INXCSR0 (IN_PKT_RDY) and INXCSR5 (TX_FIFO_NOT_EMPTY):

These two bits are read together to determine IN FIFO status. A “1” can be written to the INXCSR0 bit by the CPU to indicate a packet of data is written to the FIFO (see “IN (Transmit) FIFO” operation for details).

INXCSR1 (UNDER_RUN):

This bit is used in ISO mode only to indicate to the CPU that a FIFO underrun has occurred. The USB FCU sets this bit to a “1” at the beginning of an IN token if no data packet is in the FIFO. Setting this bit causes the INST12 bit of the Interrupt Status Register 2 to set. The CPU writes a “0” to clear this bit.

INXCSR2 (SEND_STALL):

The CPU writes a “1” to this bit when the endpoint is stalled (transmitter halt). The USB FCU returns a STALL handshake while this bit is set. The CPU writes a “0” to clear this bit.

INXCSR3 (ISO):

The CPU writes a “1” to this bit to initialize the respective endpoint as an isochronous endpoint for IN transactions.

INXCSR4 (INTPT):

The CPU writes a “1” to this bit to initialize this endpoint as a status change endpoint for IN transactions. This bit is set only if the corresponding endpoint is to be used to communicate rate feedback information (see Chapter . IN (Transmit) FIFOs for details).

INXCSR5 (TX_FIFO_NOT_EMPTY):

The USB FCU sets this bit to a “1” when there is data in the IN FIFO. This bit in conjunction with IN_PKT_RDY bit provides the transmit FIFO status information (see “IN (Transmit) FIFO” operation for details).

INXCSR6 (FLUSH):

The CPU writes a “1” to this bit to flush the IN FIFO. If there is one packet in the IN FIFO, a flush causes the IN FIFO to be empty, if there are two packets in the IN FIFO, a flush causes the older packet to be flushed out from the IN FIFO. Setting the INXCSR6 (FLUSH) bit during transmission could produce unpredictable results.

INXCSR7 (AUTO_SET):

If the CPU sets this bit to a “1”, the IN_PKT_RDY bit is set automatically by the USB FCU after the number of bytes of data equal to the maximum packet size (MAXP) is written into the IN FIFO (see “IN (Transmit) FIFO” operation for details).

MSB 7							LSB 0
INXCSR7	INXCSR6	INXCSR5	INXCSR4	INXCSR3	INXCSR2	INXCSR1	INXCSR0

Address: 0319_{16}
Address: 0321_{16}
Address: 0329_{16}
Address: 0331_{16}

Access: R/W
Reset: 00_{16}

INXCSR0 IN_PKT_RDY Bit (bit 0) (Write “1” only or Read)
- 0: In packet is not ready
- 1: In packet is ready

INXCSR1 UNDER_RUN Flag (bit 1) (Write “0” only or Read)
- 0: No FIFO underrun
- 1: FIFO underrun has occurred

INXCSR2 SEND_STALL Bit (bit 2)
- 0: No action
- 1: Stall IN Endpoint X by the CPU

INXCSR3 ISO Bit (bit 3)
- 0: Select non-isochronous transfer
- 1: Select isochronous transfer

INXCSR4 INTPT Bit (bit 4)
- 0: Select non-rate feedback interrupt transfer
- 1: Select rate feedback interrupt transfer

INXCSR5 TX_NOT_EPT Flag (bit 5) (Read Only - Write “0”)
- 0: Transmit FIFO is empty
- 1: Transmit FIFO is not empty

INXCSR6 FLUSH Bit (bit 6) (Write Only - Read “0”)
- 0: No action
- 1: Flush the FIFO

INXCSR7 AUTO_SET Bit (bit 7)
- 0: AUTO_SET disabled
- 1: AUTO_SET enabled

Figure 47: USB Endpoint x IN CSR

The USB Endpoint x OUT CSR (Control & Status Register), shown in Figure 48, contains control and status information of the respective OUT endpoint 1-4.

OUTXCSR0 (OUT_PKT_RDY):

The USB FCU sets the this bit to a “1” after it successfully receives a packet of data from the host. This bit is cleared by the CPU or by the USB FCU after a packet of data is unloaded from the FIFO (see “OUT (Receive) FIFO” operation for details).

OUTXCSR1 (OVER_RUN):

This bit is used in ISO mode only to indicate to the CPU that a FIFO overrun has occurred. The USB FCU sets this bit to a “1” at the beginning of an OUT token if the OUTXCSR0 (OUT_PKT_RDY) bit is not cleared. Setting this bit causes the INST12 bit of the Interrupt Status Register 2 to set. The CPU writes a “0” to clear this bit.

OUTXCSR2 (SEND_STALL):

The CPU writes a “1” to this bit when the endpoint is stalled (receiver halt). The USB FCU returns a STALL handshake while this bit is set. The CPU writes a “0” to clear this bit.

OUTXCSR3 (ISO):

The CPU sets this bit to a “1” to initialize the respective endpoint as an Isochronous endpoint for OUT transactions.

OUTXCSR4 (FORCE_STALL):

The USB FCU sets this bit to a “1” if the host sends out a larger data packet than the MAXP size. The USB FCU returns a STALL handshake while this bit is set. The CPU writes a “0” to clear this bit.

OUTXCSR5 (DATA_ERR):

The USB FCU sets this bit to a “1” to indicate a CRC error or a bit stuffing error received in an ISO packet. The CPU writes a “0” to clear this bit.

OUTXCSR6 (FLUSH):

The CPU writes a “1” to this to flush the OUT FIFO. If there is one packet in the OUT FIFO, a flush causes the OUT FIFO to be empty, if there are two packets in the OUT FIFO, a flush causes the older packet to be flushed out from the OUT FIFO. Setting the OUTXCSR6 (FLUSH) bit during reception could produce unpredictable results.

OUTXCSR7 (AUTO_CLR):

If the CPU sets this bit to a “1”, the OUT_PKT_RDY bit is cleared automatically by the USB FCU after the number of bytes of data equal to the maximum packet size (MAXP) is unloaded from the OUT FIFO (see “OUT (Receive) FIFO” operation for details).

MSB 7								LSB 0
	OUTXCSR7	OUTXCSR6	OUTXCSR5	OUTXCSR4	OUTXCSR3	OUTXCSR2	OUTXCSR1	OUTXCSR0

Address: $031A_{16}$
Address: 0322_{16}
Address: $032A_{16}$
Address: 0332_{16}

Access: R/W
Reset: 00_{16}

OUTXCSR0 OUT_PKT_RDY Flag (bit 0) (Write “0” only or Read)
- 0: Out packet is not ready
- 1: Out packet is ready

OUTXCSR1 OVER_RUN Flag (bit 1) (Write “0” only or Read)
- 0: No FIFO overrun
- 1: FIFO overrun occurred

OUTXCSR2 SEND_STALL Bit (bit 2)
- 0: No action
- 1: Stall OUT Endpoint X by the CPU

OUTXCSR3 ISO Bit (bit 3)
- 0: Select non-isochronous transfer
- 1: Select isochronous transfer

OUTXCSR4 FORCE_STALL Flag (bit 4) (Write “0” only or Read)
- 0: No action
- 1: Stall Endpoint X by the USB FCU

OUTXCSR5 DATA_ERR Flag (bit 5) (Write “0” only or Read)
- 0: No error
- 1: CRC or bit stuffing error received in an ISO packet

OUTXCSR6 FLUSH Bit (bit 6) (Write Only - Read “0”)
- 0: No action
- 1: Flush the FIFO

OUTXCSR7 AUTO_CLR Bit (bit 7)
- 0: AUTO_CLR disabled
- 1: AUTO_CLR enabled

Figure 48: USB Endpoint x OUT CSR

The USB Endpoint x IN MAXP, shown in Figure 49, indicates the maximum packet size (MAXP) of an Endpoint x IN packet. The default values for Endpoints 1-4 are 0 bytes. The CPU can change this value, as negotiated with the host controller through the SET_DESCRIPTOR command.

MSB 7								LSB 0
	IMAXP7	IMAXP6	IMAXP5	IMAXP4	IMAXP3	IMAXP2	IMAXP1	IMAXP0

Address: $031B_{16}$
Address: 0323_{16}
Address: $032B_{16}$
Address: 0333_{16}

Access: R/W
Reset: 00_{16}

IMAXP7:0 Maximum packet size (MAXP) of Endpoint x IN packet.
For endpoints that support smaller FIFO size, unused bits are not implemented (always write “0” to those bits)

Figure 49: USB Endpoint x IN MAXP

The USB Endpoint x OUT MAXP, shown inFigure 50, indicates the maximum packet size (MAXP) of an Endpoint x OUT packet. The default values for endpoints 1-4 are 0 bytes. The CPU can change this value, as negotiated with the host controller through the SET_DESCRIPTOR command.

MSB 7							LSB 0
OMAXP7	OMAXP6	OMAXP5	OMAXP4	OMAXP3	OMAXP2	OMAXP1	OMAXP0

OMAXP7:0	Maximum packet size (MAXP) of Endpoint x OUT packet. For endpoints that support smaller FIFO size, unused bits are not implemented (always write “0” to those bits)

Address: $031C_{16}$
Address: 0324_{16}
Address: $032C_{16}$
Address: 0334_{16}

Access: R/W
Reset: 00_{16}

Figure 50: USB Endpoint x OUT MAXP

The **U**SB Endpoint x OUT WRT CNT register, shown in Figure 51, contains the number of bytes of the current data set in the OUT FIFO. The USB FCU sets the value in the Write Count Register after having successfully received a packet of data from the host. The CPU reads the register to determine the number of bytes to be read from the FIFO.

MSB 7							LSB 0
W_CNT7	W_CNT6	W_CNT5	W_CNT4	W_CNT3	W_CNT2	W_CNT1	W_CNT0

W_CNT7:0 Receive Byte Count.

Address: $031D_{16}$
Address: 0325_{16}
Address: $032D_{16}$
Address: 0335_{16}

Access: R
Reset: 00_{16}

Figure 51: USB Endpoint x OUT WRT CNT

The USB Endpoint x FIFO Register, shown in Figure 52, is the USB IN (transmit) and OUT (receive) FIFO data register. The CPU writes data to these registers for the corresponding Endpoint IN FIFO and reads data from these registers for the corresponding Endpoint OUT FIFO.

MSB 7							LSB 0
DATA_7	DATA_6	DATA_5	DATA_4	DATA_3	DATA_2	DATA_1	DATA_0

DATA_7:0 Endpoint x IN/OUT FIFO register

Address: 0338_{16}
Address: 0339_{16}
Address: $033A_{16}$
Address: $033B_{16}$
Address: $033C_{16}$

Access: R
Reset: N/A

Figure 52: USB Endpoint x FIFO Register

2.19 DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU.Table 13 shows the DMAC specifications. Figure 53 shows the block diagram of the DMAC. Figure 54, Figure 55 and Figure 56 show the registers used by the DMAC.

Table 13: DMAC specifications

Item	Specification
Number of channels	2 (cycle steal method)
Transfer memory space	•From any SFR, RAM, or ROM address to a fixed address •From a fixed address to any SFR or RAM address •From a fixed address to a fixed address (Note that DMA-related registers [0020_{16} to $003F_{16}$] cannot be accessed)
Maximum number 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 (INT0 can be selected by DMA0, INT1 by DMA1) Timer A0 to timer A4 interrupt requests Timer B0 to timer B1 interrupt requests UART0 transmission and reception interrupt requests UART1 transmission and reception interrupt requests UART2 transmission and reception interrupt requests A-D conversion interrupt requests USB function interrupt requests USB SOF 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 The DMA enable bit is cleared and transfer ends when an underflow occurs in the transfer counter Repeat transfer When an underflow occurs in the transfer counter, the value in the transfer counter reload register is reloaded into the transfer counter and the DMA transfer is repeated
DMA interrupt request generation timing	When an underflow occurs in the transfer counter
DMA startup	Single transfer Transfer starts when the DMA is requested after “1” is written to the DMA enable bit Repeat transfer Transfer starts when the DMA is requested after “1” is written to the DMA enable bit or after an underflow occurs in the transfer counter
DMA shutdown	When “0” is written to the DMA enable bit When, in single transfer mode, an underflow occurs in the transfer counter
Forward address pointer and reload timing for transfer counter	When DMA transfer starts, the value of whichever of the source or destination pointer that is set up as the forward pointer is reloaded into the forward address pointer. The value in the transfer counter reload register is reloaded into 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 sets up as the forward register is the same as reading the value of the forward address pointer.

Note:DMA transfer is not affected by any interrupt.

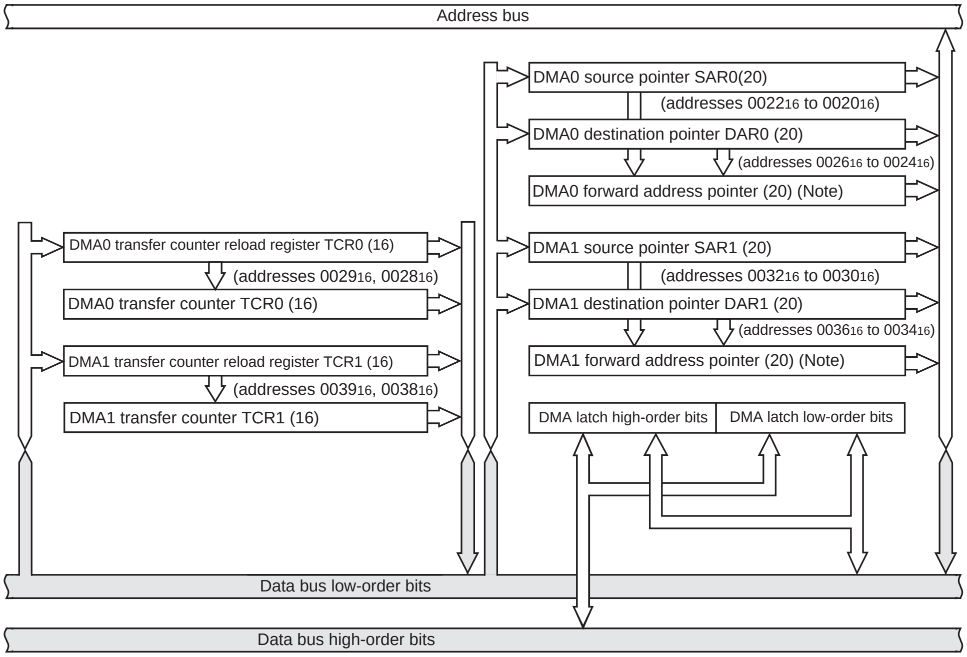
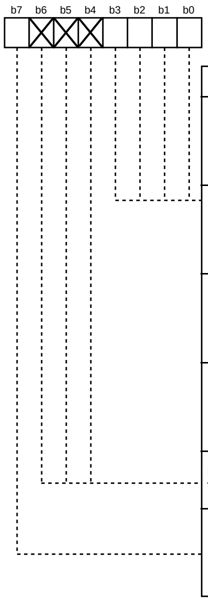

Address bus
DMA0 source pointer SAR0(20)
(addresses 002216 to 002016)
DMA0 destination pointer DAR0 (20)
(addresses 002616 to 002416)
DMA0 forward address pointer (20) (Note)
DMA1 source pointer SAR1 (20)
(addresses 003216 to 003016)
DMA1 destination pointer DAR1 (20)
(addresses 003616 to 003416)
DMA1 forward address pointer (20) (Note)
DMA latch high-order bits
DMA latch low-order bits
DMA0 transfer counter reload register TCR0 (16)
(addresses 002916, 002816)
DMA0 transfer counter TCR0 (16)
DMA1 transfer counter reload register TCR1 (16)
(addresses 003916, 003816)
DMA1 transfer counter TCR1 (16)
Data bus low-order bits
Data bus high-order bits

Figure 53: Block diagram of DMAC

DMAi request cause select register



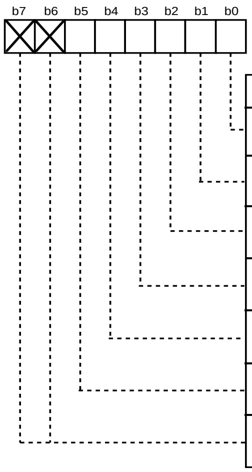
Symbol DMiSL (i=0,1) | Address 03B8$_{16}$,03BA$_{16}$ | When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b3 b2 b1 b0 0 0 0 0 : Falling edge of $\overline{INT0}$ / $\overline{INT1}$ pin (Note1) 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 0 1 1 1 : Timer B0 1 0 0 0 : Timer B1 1 0 0 1 : USB0/USB1 (Note 3) 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 / UART1 receive (Note 2)	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is “0”.			–	–
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”)	○	○

Note 1: Address 03B8$_{16}$ is for INT0, 03BA$_{16}$ is for INT1.
Note 2: Address 03B8$_{16}$ is for UART1 transmit, 03BA$_{16}$ is for UART1 receive.
Note 3: Address 03B8$_{16}$ is for USB0, 03BA$_{16}$ is for USB1.

Figure 54: DMAC register (1)

DMAi control register



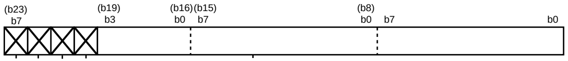
Symbol: DMAiCON(i=0,1)
Address: $002C_{16}$, $003C_{16}$
When reset: $00000X00_2$

Bit symbol	Bit name	Function	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. These bits can neither be set nor reset. When read, the value of these bits is “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 55: DMAC register (2)

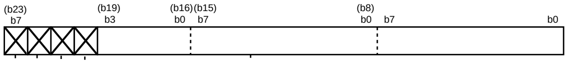
DMAi source pointer (i = 0, 1)



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

Function	Transfer count specification	R	W
Source pointer Stores the source address	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is “0”.		–	–

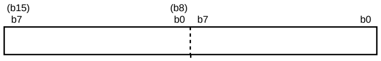
DMAi destination pointer (i = 0, 1)



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

Function	Transfer count specification	R	W
Destination pointer Stores the destination address	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is “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 56: 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 and the 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 is one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

(2) DMAC transfer cycles

Any combination of even or odd transfer read and write addresses is possible. Table 14 show the number of DMAC transfer cycles. Table 15 shows the corresponding coefficient values. Figure 57 shows an example of the transfer cycle for a source read.

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

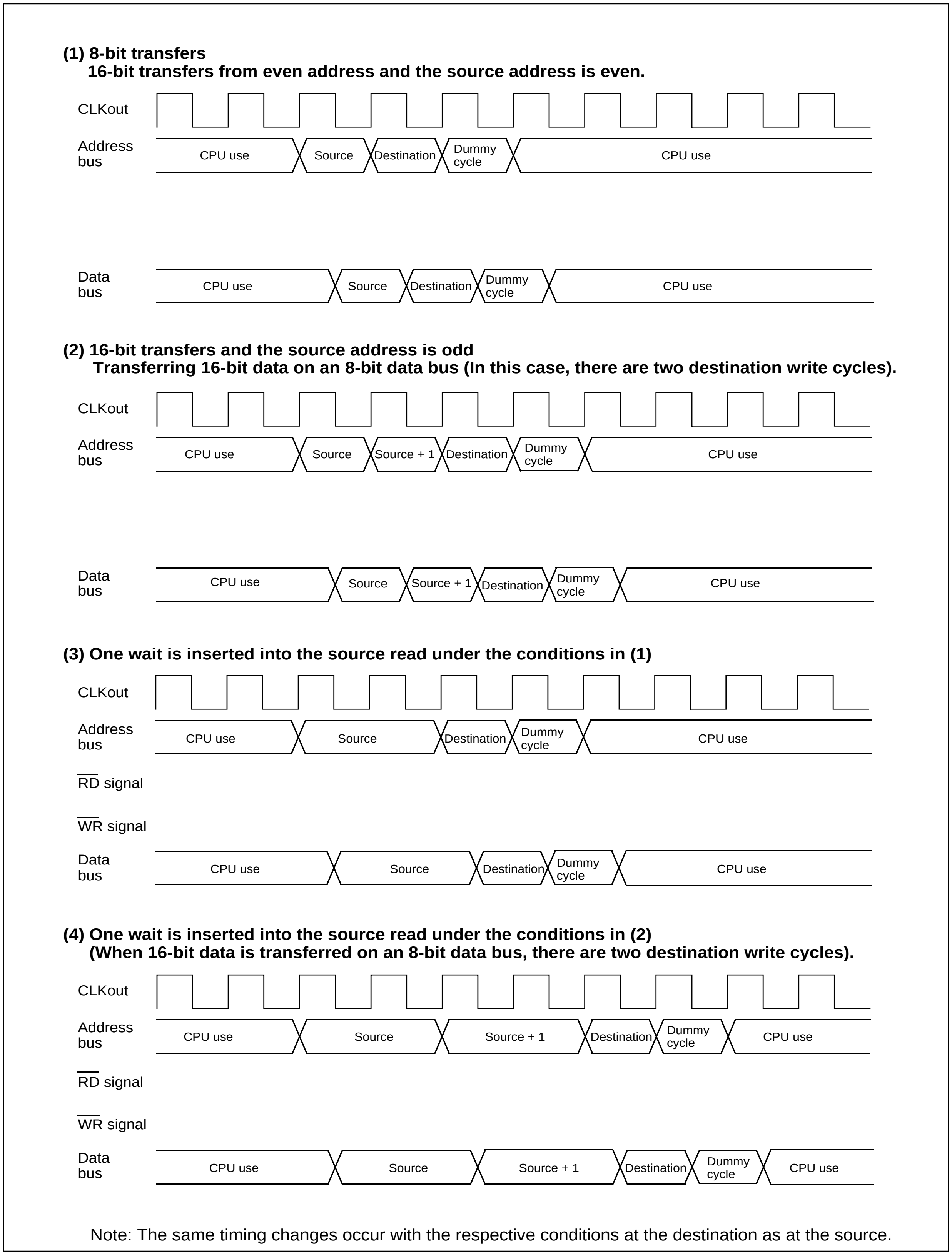
Number of transfer cycles per transfer unit = Number of read cycles x j + Number of write cycles x k

Table 14: Number of DMAC transfer cycles

Transfer unit	Access address	Single-chip mode	
		Number of read cycles	Number of write cycles
8-bit transfers (DMBIT=”1”)	Even	1	1
	Odd	1	1
16-bit transfers (DMBIT=”0”)	Even	1	1
	Odd	2	2

Table 15: Coefficients j,k

Internal memory		
Internal ROM/ RAM No wait	Internal ROM/ RAM with wait	SFR area
1	2	2



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

Figure 57: Example of the transfer cycle for a source read

2.20 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. Figure 58 shows the block diagram of Timers A and B.

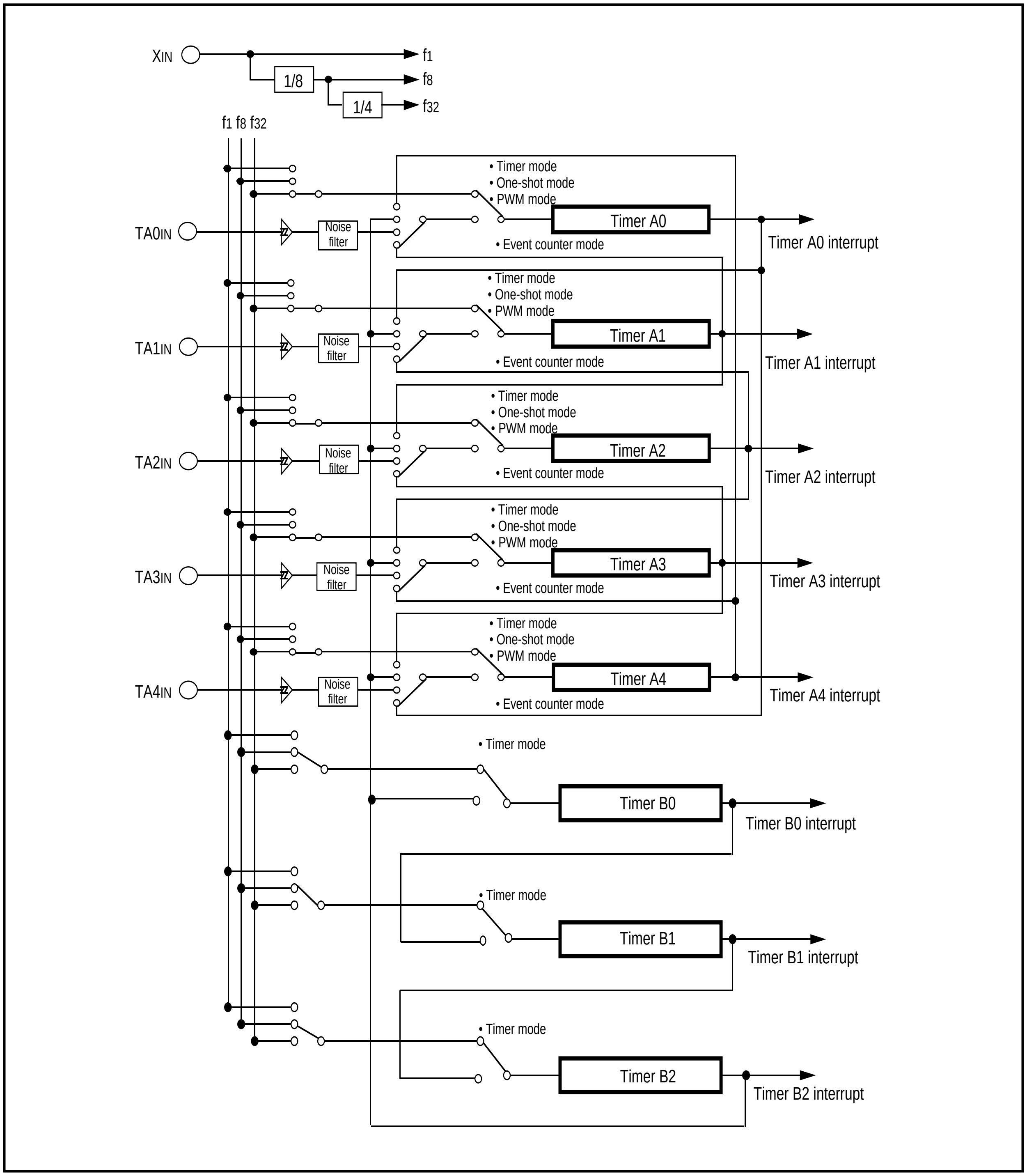


Figure 58: Timer A andTimer B block diagram

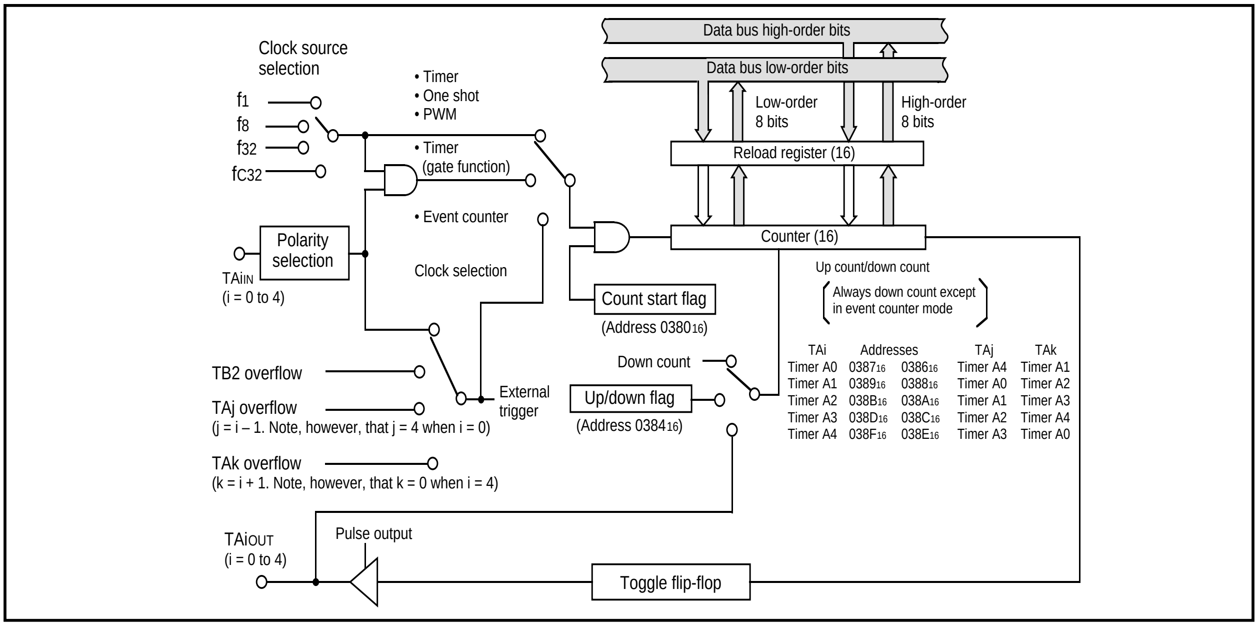
2.21 Timer A

Figure 59, Figure 60, Figure 61, and Figure 62 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.

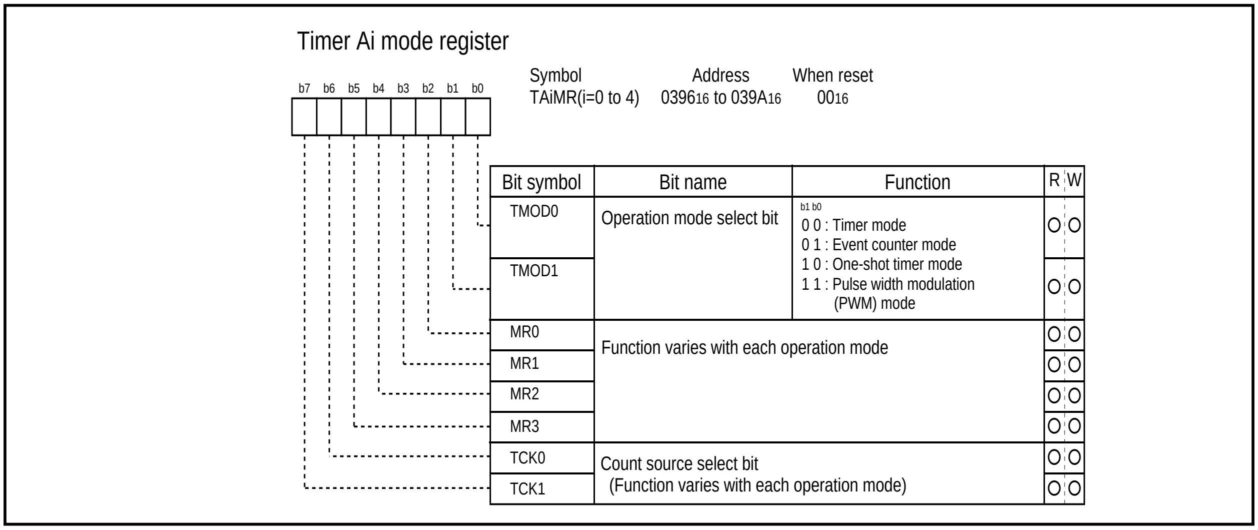
Timer A has the four operation modes listed as follows:

- 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.



TAi	Addresses		TAj	TAk
Timer A0	0387_{16}	0386_{16}	Timer A4	Timer A1
Timer A1	0389_{16}	0388_{16}	Timer A0	Timer A2
Timer A2	$038B_{16}$	$038A_{16}$	Timer A1	Timer A3
Timer A3	$038D_{16}$	$038C_{16}$	Timer A2	Timer A4
Timer A4	$038F_{16}$	$038E_{16}$	Timer A3	Timer A0

Figure 59: Block diagram of Timer A


b7 b6 b5 b4 b3 b2 b1 b0

Timer Ai mode register

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 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 60: Timer A related Registers (1)

Timer Ai register (Note)

(b15) b7 … (b8) b0 b7 … b0

Symbol	Address	When reset
TA0	0387_{16},0386_{16}	Indeterminate
TA1	0389_{16},0388_{16}	Indeterminate
TA2	$038B_{16}$,$038A_{16}$	Indeterminate
TA3	$038D_{16}$,$038C_{16}$	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 $FFFF_{16}$	○	○
Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	○	○
One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$	×	○
Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$	×	○
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	00_{16} to FE_{16} (Both high-order and low-order addresses)	×	○

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

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0380_{16}	00_{16}

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		○	○

Up/down flag

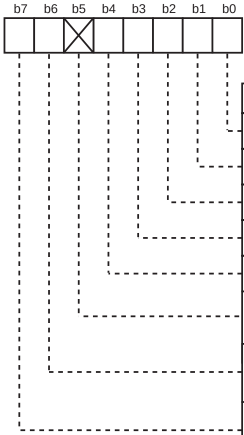
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UDF	0384_{16}	00_{16}

Bit symbol	Bit name	Function	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	○	○
TA1UD	Timer A1 up/down flag		○	○
TA2UD	Timer A2 up/down flag		○	○
TA3UD	Timer A3 up/down flag		○	○
TA4UD	Timer A4 up/down flag		○	○
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled When not using the two-phase pulse signal processing function, set the select bit to “0”	×	○
TA3P	Timer A3 two-phase pulse signal processing select bit		×	○
TA4P	Timer A4 two-phase pulse signal processing select bit		×	○

Figure 61: Timer A-related registers (2)

One-shot start flag

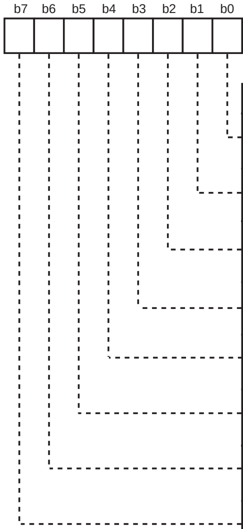


Symbol	Address	When reset
ONSF	0382_{16}	$00X00000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is 0	○	○
TA1OS	Timer A1 one-shot start flag		○	○
TA2OS	Timer A2 one-shot start flag		○	○
TA3OS	Timer A3 one-shot start flag		○	○
TA4OS	Timer A4 one-shot start flag		○	○
Nothing is assigned. This bit can neither be set nor reset. When read, its content is indeterminate.			—	—
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on $TA0_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	○	○
TA0TGH			○	○

Note: Set the corresponding port direction register to 0 .

Trigger select register

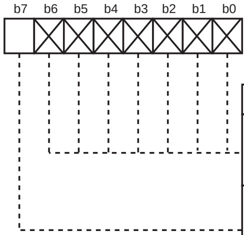


Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on $TA1_{IN}$ 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 $TA2_{IN}$ 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 $TA3_{IN}$ 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 $TA4_{IN}$ 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 .

Clock prescaler reset flag



Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is 0)	○	○

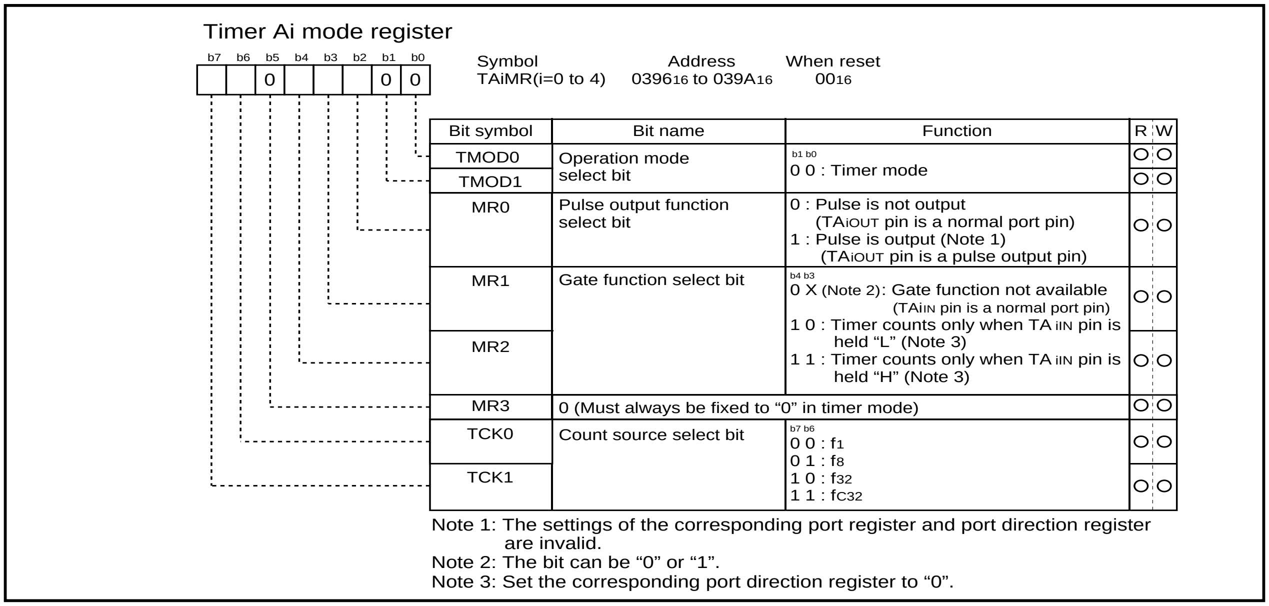
Figure 62: Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. See Table 16 below. Figure 63 shows the timer Ai mode register in timer mode.

Table 16: Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}
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 only to reload register (transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by TAiIN pin's input signal • Pulse output funtion Each time the timer underflows, the TAiOUT pin's polarity is reversed.



Timer Ai mode register

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 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 fixed to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
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 63: Timer Ai mode register in timer mode

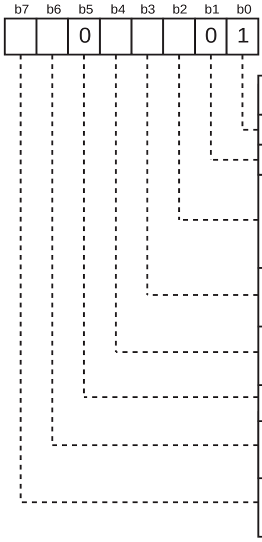
(2) Event counter mode

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 17 lists the timer specifications when counting a single-phase external signal. Figure 64 shows Timer Ai mode register in event counter mode, single-phase signal.

Table 17: Timer specification 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. (However, this does not apply when the free-run function is selected.)
Divide ration	1/ (FFF$_{16}$ - 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	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
Writer to timer	•When counting stopped When a value is written to timer Ai register, it si written to both reload register and counter •When counting in progress When a value is writtento timer Ai register, it si written to only reload register
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

Timer Ai mode register



Symbol	Address	When reset
TAiMR(i = 0, 1)	0396_{16}, 0397_{16}	00_{16}

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 fixed to 0 in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Invalid in event counter mode Can be 0 or 1		○	○

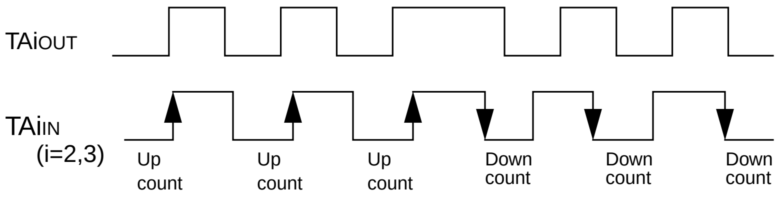
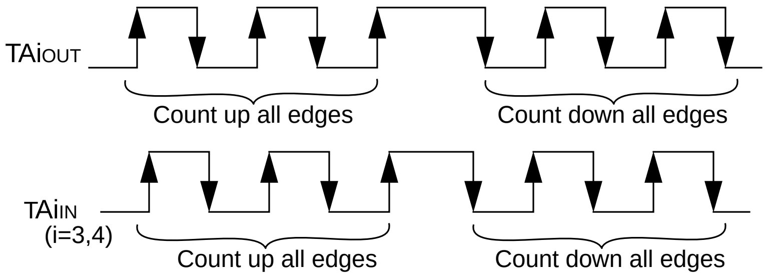
Note 1: In event counter mode, the count source is selected by the event / trigger select bit (addresses 0382_{16} and 0383_{16}).
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Valid only when counting an external signal.
Note 4: When an L signal 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 64: Timer Ai mode register in event counter mode, single signal

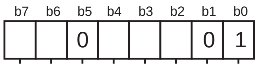
Table 18 shows the Timer specification in event counter mode when processing two-phase signal with timers A2, A3, and A4.

Figure 65 shows Timer Ai mode register in event counter mode when processing two-phase signal.

Table 18: Timer specification in even 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 overflow or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	1/ (FFF16 - 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 overflow or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Writer 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	•Normal processing operation The timer counts up rising edges or counts down falling edges on the TAiIN pin when input signal on the TAiOUT pin is “H”  TAiOUT TAiIN (i=2,3) Up count Up count Up count Down count Down count Down count •Multiply-by-4 processing operation 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 rising and falling edges on the TAiOUT and TAiIN pins.  TAiOUT Count up all edges Count down all edges TAiIN (i=3,4) Count up all edges Count down all edges
Note	This does not apply when the free-run function is selected.

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

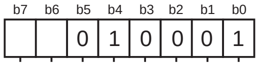


Symbol TAiMR(i = 2 to 4)
Address 0398_{16} to $039A_{16}$
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter 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	Count polarity select bit (Note 2)	0 : Counts external signal's falling edges 1 : Counts external signal's rising edges	○	○
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 3)	○	○
MR3	0 : (Must always be 0 in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse signal processing operation select bit (Note 4)(Note 5)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: This bit is valid when only counting an external signal.
Note 3: Set the corresponding port direction register to 0 .
Note 4: This bit is valid for the timer A3 mode register.
For timer A2 and A4 mode registers, this bit can be 0 or 1 .
Note 5: 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 .

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



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

Bit symbol	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.
For timer A2 and A4 mode registers, this bit can be 0 or 1 .
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 .

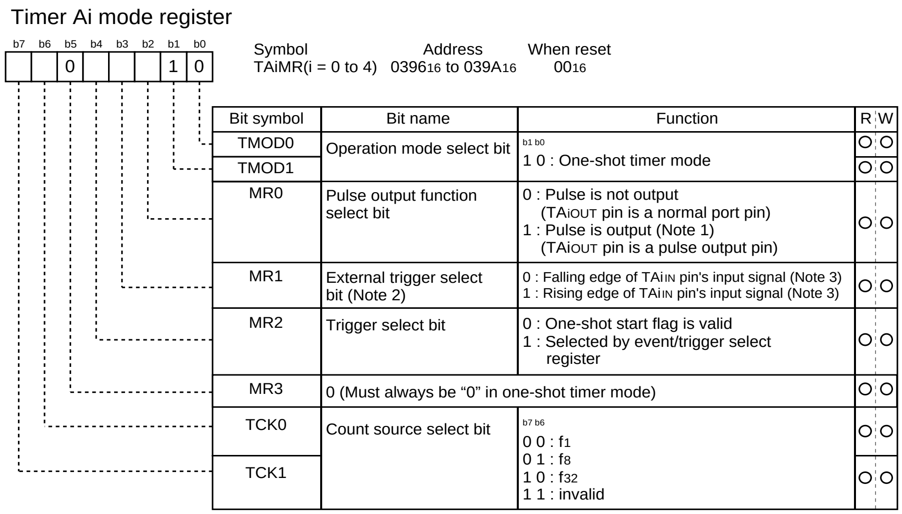
Figure 65: Timer Ai mode register in event counter mode, two-phase signal

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 19 .) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 66 shows the timer Ai mode register in one-shot mode.

Table 19: Timer specifications in one-shot timer mode

Item	Specification
Count source	f1, f8, f32
Count operation	•The timer counts down •When the count reaches000016, 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 000016 • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 000016
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT 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)



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	○	○
TMOD1			○	○
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)	○	○
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)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 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 0 1 : f_8 1 0 : f_{32} 1 1 : invalid	○	○
TCK1			○	○

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 66: Timer Ai mode register in one-shot mode

(4) Pulse-width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 20 .) In this mode, the counter functions as either a 16-bit pulse-width modulator or an 8-bit pulse-width modulator. Figure 67 shows the timer Ai mode register in pulse-width modulation mode. Figure 68 shows the example of how a 16-bit pulse-width modulator operates. Figure 69 shows the example of how an 8-bit pulsewidth modulator operates.

Table 20: Timer specifications in pulse-width modulation mode

Item	Specification
Count source	f1, f8, f32
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 / fin n : Set value •Cycle time(216-1) / 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 (28-1) (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 A register, it is written to only reload register (transferred to counter at next reload timer).

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

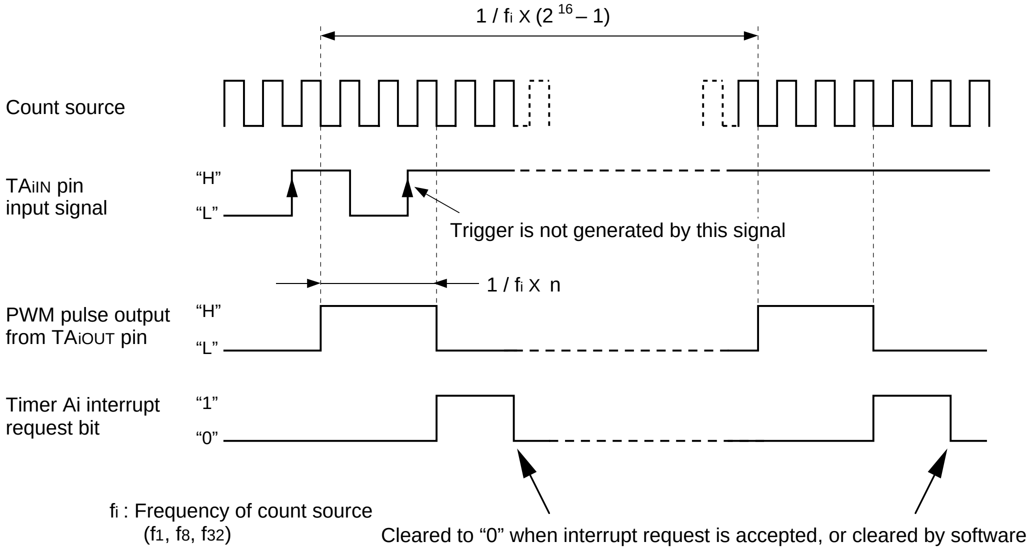
Symbol	Address	When reset
TAiMR(i=0 to 4)	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be “1” in PWM mode)		○	○
MR1	External trigger select bit (Note 1)	0: Falling edge of TAi_{IN} pin's input signal (Note 2) 1: Rising edge of TAi_{IN} pin's input signal (Note 2)	○	○
MR2	Trigger select bit	0: Count start flag is valid 1: Selected by event/trigger select register	○	○
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	○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TCK1			○	○

Note 1: Valid only when the TA_{iIN} 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 67: Timer Ai mode register in pulse-width modulation mode

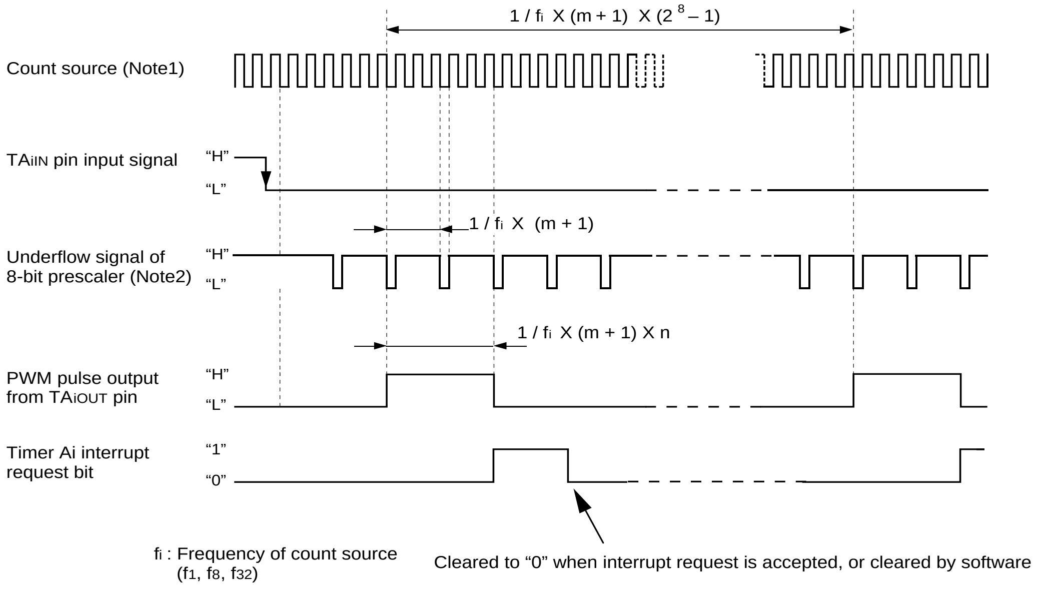
Condition : Reload register = 0003_{16}, when external trigger
(rising edge of TAi_{IN} pin input signal) is selected


1 / fi X (2 16 – 1)
Count source
TAiIN pin input signal
“H”
“L”
Trigger is not generated by this signal
1 / fi X n
PWM pulse output from TAiOUT pin
“H”
“L”
Timer Ai interrupt request bit
“1”
“0”
fi : Frequency of count source (f1, f8, f32)
Cleared to “0” when interrupt request is accepted, or cleared by software

Note: n = 0000_{16} to $FFFE_{16}$.

Figure 68: Example of how a 16-bit pulse-width modulator operates

Condition : Reload register high-order 8 bits = 02_{16}
Reload register low-order 8 bits = 02_{16}
External trigger (falling edge of TAiIN pin input signal) is selected


1 / fi X (m + 1) X (2^8 – 1)
Count source (Note1)
TAiIN pin input signal
"H"
"L"
1 / fi X (m + 1)
Underflow signal of 8-bit prescaler (Note2)
"H"
"L"
1 / fi X (m + 1) X n
PWM pulse output from TAiOUT pin
"H"
"L"
Timer Ai interrupt request bit
"1"
"0"

fi : Frequency of count source
(f1, f8, f32)

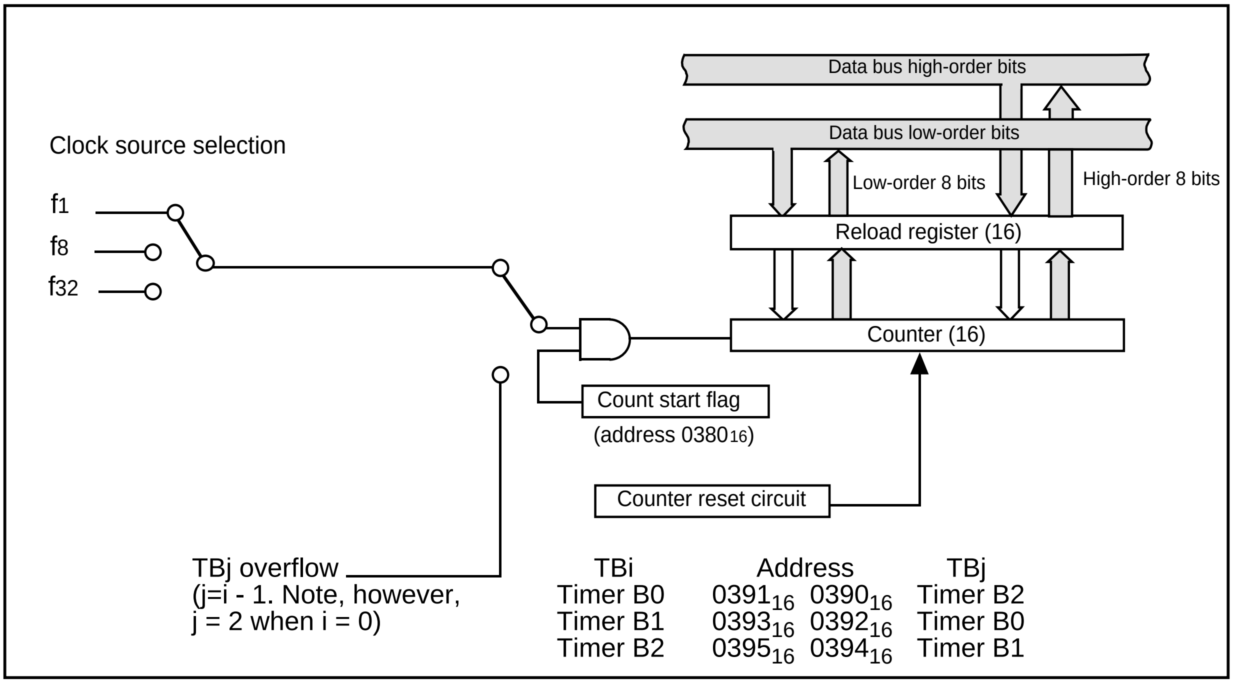
Cleared to "0" when interrupt request is accepted, or cleared by software

Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00_{16} to FE_{16}; n = 00_{16} to FE_{16}.

Figure 69: Example of how an 8-bit pulse-width modulator operates

2.22 Timer B

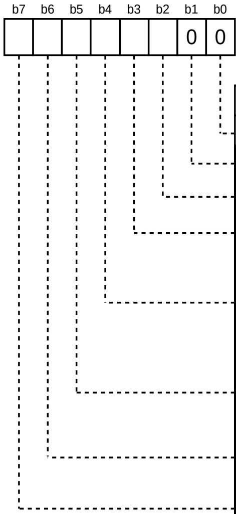
Figure 70 shows the block diagram of timer B. Figure 71 and Figure 72 show the timer B-related registers. Use the timer Bi mode register (i = 0 to 2) bits 0 and 1 to choose the desired mode. Timer B works in Timer mode only (i.e., the timer counts an in internal count source).



TBi	Address		TBj
Timer B0	0391_{16}	0390_{16}	Timer B2
Timer B1	0393_{16}	0392_{16}	Timer B0
Timer B2	0395_{16}	0394_{16}	Timer B1

Figure 70: Block diagram of Timer B

Timer Bi mode register



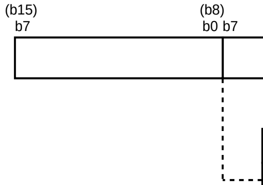
Symbol	Address	When reset
TBiMR(i=0 to 2)	$039B_{16}$ to $039D_{16}$	$00XX0000_2$

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 (Fixed to “0” in timer mode ; i = 0)		○ (Note 1)	○
	Nothing is assiigned (i = 1, 2). This bit can neither be set nor reset. When read, its content is indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. This bit can neither be set nor reset. When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : invalid	○	○
TCK1			○	○

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

Figure 71: Timer B-related registers

Timer Bi register (Note)

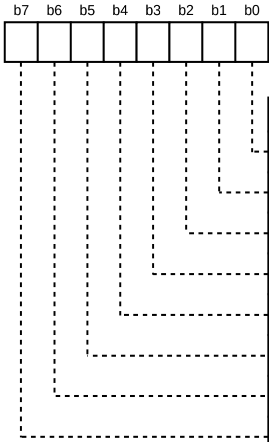


Symbol	Address	When reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate
TB2	0395_{16}, 0394_{16}	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000_{16} to $FFFF_{16}$	○	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	——	○	×

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

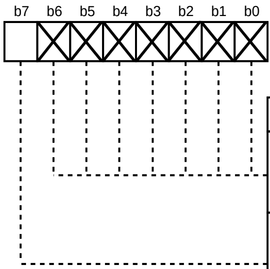
Count start flag



Symbol	Address	When reset
TABSR	0380_{16}	00_{16}

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	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 72: Timer B-related registers

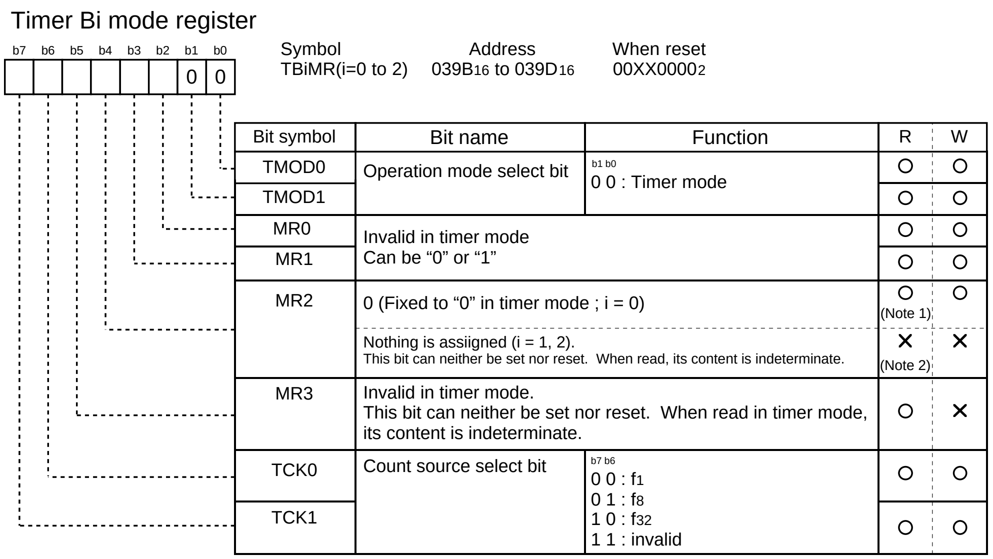
(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 21) Figure 73 shows the Timer Bi mode register in timer mode.

Table 21: Timer specifications in timer mode

Item	Specification
Count source	f1, f8, f32
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 (see Note)

Note: Timer B2 does not generate an interrupt; it is used only as a prescaler.



Timer Bi mode register

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 (Fixed to “0” in timer mode ; i = 0)		○ (Note 1)	○
	Nothing is assiigned (i = 1, 2). This bit can neither be set nor reset. When read, its content is indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. This bit can neither be set nor reset. When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : invalid	○	○
TCK1			○	○

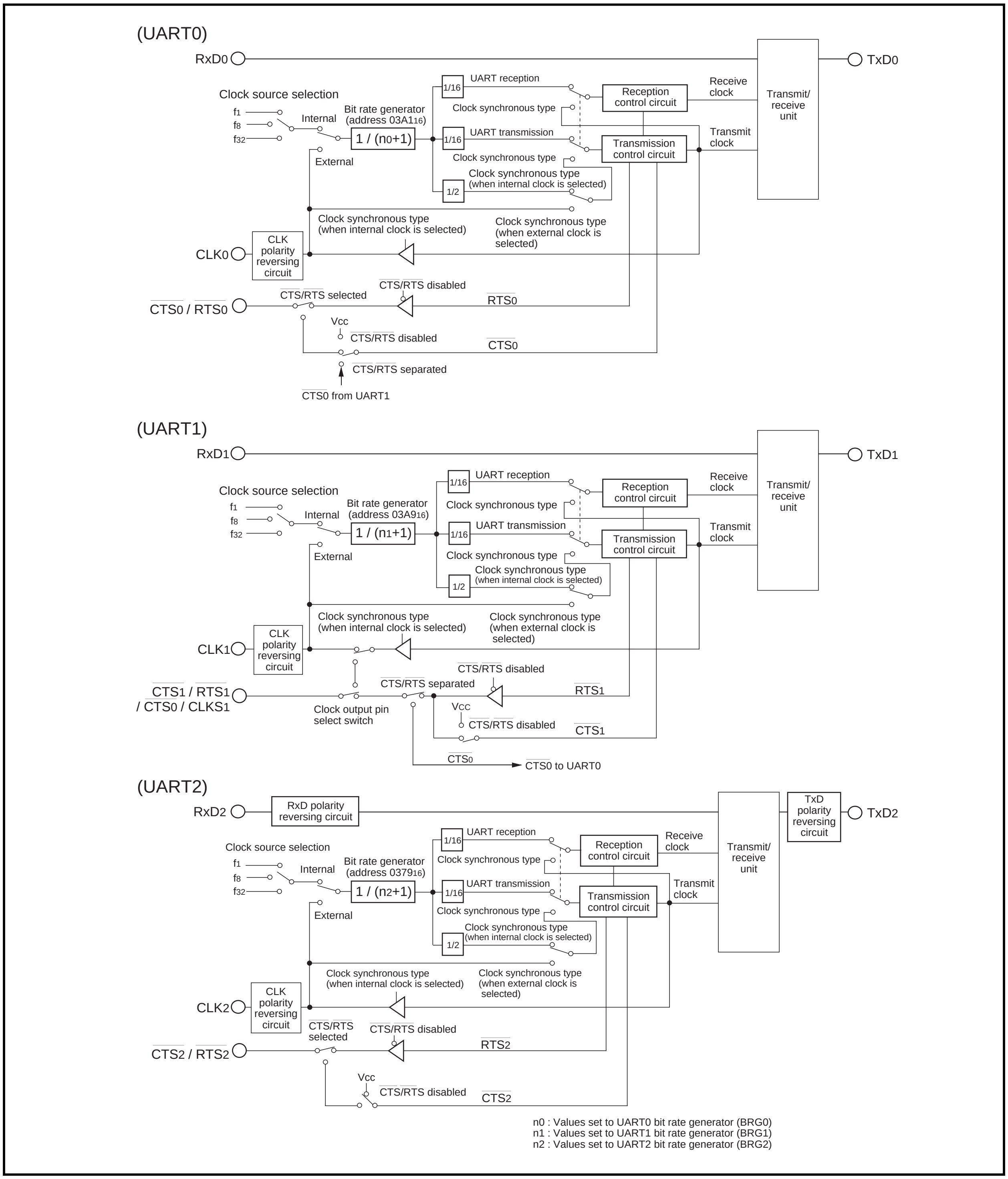
Note 1: Timer B0.
Note 2: Timer B1, timer B2.

Figure 73: Timer Bi mode register in timer mode

2.23 UART0 through UART2

Serial I/O is configured as three channels: UART0, UART1, and UART2. UART0, UART1, and UART2 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 74 shows the block diagram of UART0, UART1, and UART2.


(UART0)
RxD0
TxD0
Clock source selection
f1
f8
f32
Internal
External
Bit rate generator (address 03A116)
1 / (n0+1)
1/16
UART reception
Clock synchronous type
Reception control circuit
Receive clock
Transmit/ receive unit
1/16
UART transmission
Clock synchronous type
Transmission control circuit
Transmit clock
Clock synchronous type (when internal clock is selected)
1/2
Clock synchronous type (when internal clock is selected)
Clock synchronous type (when external clock is selected)
CLK0
CLK polarity reversing circuit
CTS/RTS disabled
CTS/RTS selected
CTS0 / RTS0
RTS0
Vcc
CTS/RTS disabled
CTS0
CTS/RTS separated
CTS0 from UART1
(UART1)
RxD1
TxD1
Clock source selection
f1
f8
f32
Internal
External
Bit rate generator (address 03A916)
1 / (n1+1)
1/16
UART reception
Clock synchronous type
Reception control circuit
Receive clock
Transmit/ receive unit
1/16
UART transmission
Clock synchronous type
Transmission control circuit
Transmit clock
Clock synchronous type (when internal clock is selected)
1/2
Clock synchronous type (when internal clock is selected)
Clock synchronous type (when external clock is selected)
CLK1
CLK polarity reversing circuit
CTS/RTS disabled
CTS/RTS separated
CTS1 / RTS1 / CTS0 / CLKS1
RTS1
Clock output pin select switch
Vcc
CTS/RTS disabled
CTS1
CTS0
CTS0 to UART0
(UART2)
RxD2
RxD polarity reversing circuit
TxD polarity reversing circuit
TxD2
Clock source selection
f1
f8
f32
Internal
External
Bit rate generator (address 037916)
1 / (n2+1)
1/16
UART reception
Clock synchronous type
Reception control circuit
Receive clock
Transmit/ receive unit
1/16
UART transmission
Clock synchronous type
Transmission control circuit
Transmit clock
Clock synchronous type (when internal clock is selected)
1/2
Clock synchronous type (when internal clock is selected)
Clock synchronous type (when external clock is selected)
CLK2
CLK polarity reversing circuit
CTS/RTS selected
CTS/RTS disabled
CTS2 / RTS2
RTS2
Vcc
CTS/RTS disabled
CTS2

n0 : Values set to UART0 bit rate generator (BRG0)
n1 : Values set to UART1 bit rate generator (BRG1)
n2 : Values set to UART2 bit rate generator (BRG2)

Figure 74: Block diagram of UARTi (i=0 to 2)

Figure 75 and Figure 76 show the block diagram of the transmit/receive unit.

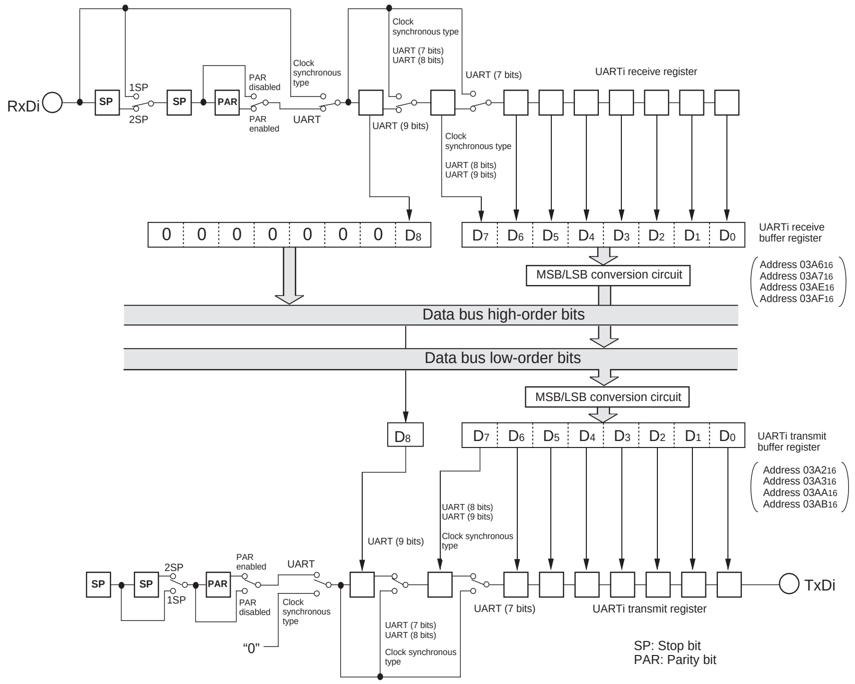


Figure 75: Block diagram of UAR2 (i=0,1) transmit/receive circuit

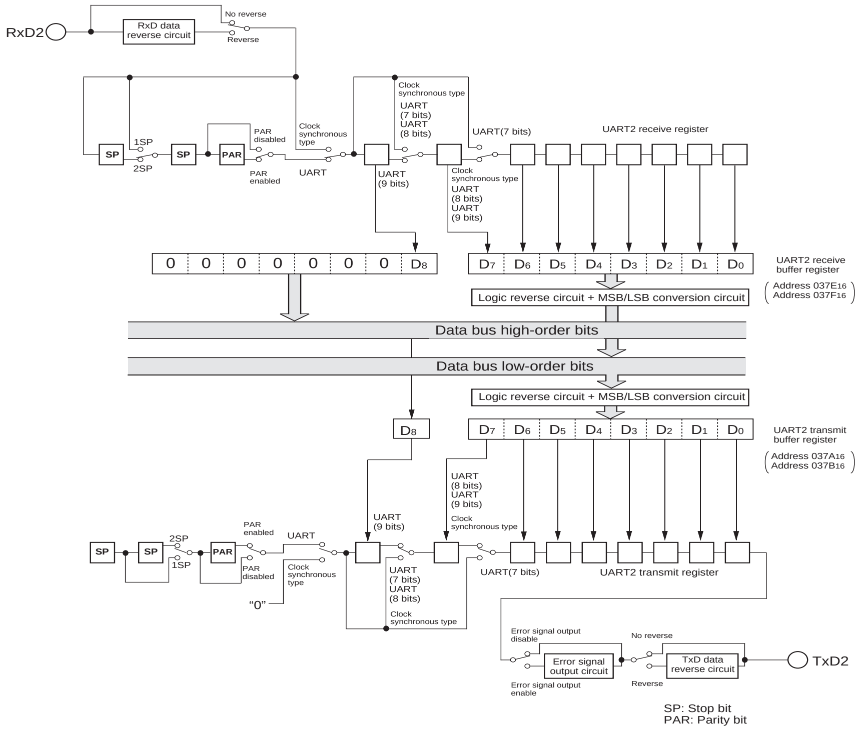


Figure 76: Block diagram of UART2 transmit/receive circuit

UART0 through UART2

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 and UART1 have almost the same functions.

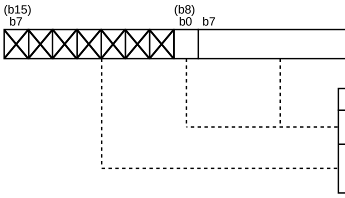
UART0 through UART2 are almost equal in their functions with minor exceptions.Table 22 shows the comparison of functions of UART0 through UART2, and Figure 77, Figure 78, Figure 79, Figure 80, and Figure 81 show the registers related to UARTi.

Table 22: Comparison of functions of UART0 throught UART2

Function	UART0	UART1	UART2
CLK polarity selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
LSB first / MSB first selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 2)
Continuous receive mode selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
Transfer clock output from multiple pins selection	Impossible	Possible (Note 1)	Impossible
Separate CTS/RTS pins	Possible	Impossible	Impossible
Serial data logic switch	Impossible	Impossible	Possible (Note 4)
Sleep mode selection	Possible (Note 3)	Possible (Note 3)	Impossible
TxD, RxD I/O polarity switch	Impossible	Impossible	Possible
TxD, RxD port output format	CMOS output	CMOS output	CMOS output
Parity error signal output	Impossible	Impossible	Possible (Note 4)
Bus collision detection	Impossible	Impossible	Possible

Note 1: Only during clock synchronous serial I/O mode.
Note 2: Only during clock synchronous serial I/O mode and 8-bit UART mode.
Note 3: Only during UART mode.
Note 4: Used for SIM interface.

UARTi transmit buffer register

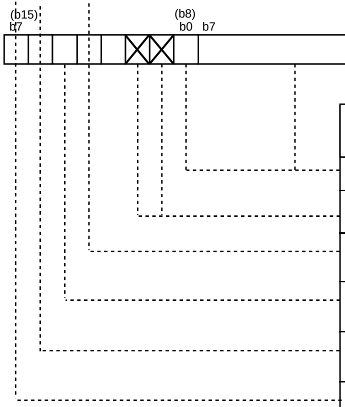


Symbol	Address	When reset
U0TB	03A3$_{16}$, 03A2$_{16}$	Indeterminate
U1TB	03AB$_{16}$, 03AA$_{16}$	Indeterminate
U2TB	037B$_{16}$, 037A$_{16}$	Indeterminate

Function	R	W
Transmit data (Note)	×	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.	—	—

Note: Bit 8 is set to “1” when I^{2}C mode is used.

UARTi receive buffer register

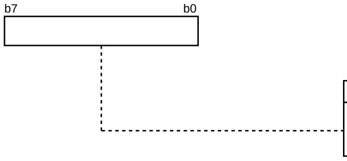


Symbol	Address	When reset
U0RB	03A7$_{16}$, 03A6$_{16}$	Indeterminate
U1RB	03AF$_{16}$, 03AE$_{16}$	Indeterminate
U2RB	037F$_{16}$, 037E$_{16}$	Indeterminate

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. These bits can neither be set nor reset. When read, the value of these bits is “0”.				—	—
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.

UARTi bit rate generator



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$_{16}$	×	○

Figure 77: Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiMR(i=0,1)	$03A0_{16}$, $03A8_{16}$	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 bit	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	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				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock	○	○
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	○	○
SLEP	Sleep select bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

UART2 transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

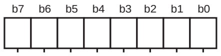
Symbol	Address	When reset
U2MR	0378_{16}	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : (Note) 0 1 1 : Inhibited 1 1 1 : Inhibited	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				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock	○	○
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”	○	○

Note: Bit 2 to bit 0 are set to “010_2” when I^2C mode is used.

Figure 78: Serial I/O-related registers (2)

UARTi transmit/receive control register 0

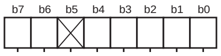


Symbol	Address	When reset
UiC0(i=0,1)	$03A4_{16}$, $03AC_{16}$	08_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit	Valid when bit 4 = “0” 0 : $\overline{\text{CTS}}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	Valid when bit 4 = “0” 0 : $\overline{\text{CTS}}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	○	○
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	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled 1 : CTS/RTS function disabled ($P6_0$ and $P6_4$ function as programmable I/O port)	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled 1 : CTS/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: The settings of the corresponding port register and port direction register are invalid.

UART2 transmit/receive control register 0



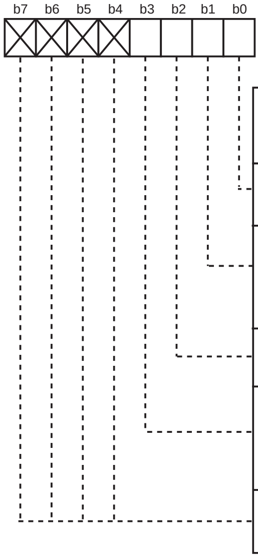
Symbol	Address	When reset
U2C0	$037C_{16}$	08_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit	Valid when bit 4 = “0” 0 : $\overline{\text{CTS}}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	Valid when bit 4 = “0” 0 : $\overline{\text{CTS}}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	○	○
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	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled 1 : CTS/RTS function disabled ($P7_3$ functions programmable I/O port)	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled 1 : CTS/RTS function disabled ($P7_3$ functions programmable I/O port)	○	○
Nothing is assigned. This bit can neither be set nor reset. When read, the value of this bit is “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 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding port direction register to “0”.
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.

Figure 79: Serial I/O-related registers (3)

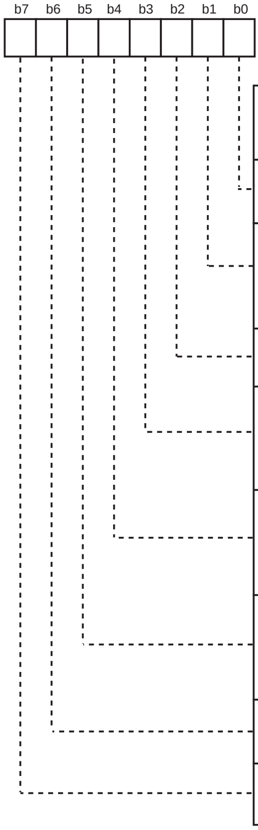
UARTi transmit/receive control register 1



Symbol	Address	When reset
UiC1(i=0,1)	03A5$_{16}$,03AD$_{16}$	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	○	○
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	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
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	○	×
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is “0”.				—	—

UART2 transmit/receive control register 1

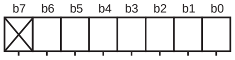


Symbol	Address	When reset
U2C1	037D$_{16}$	02$_{16}$

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	○	○
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	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
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	○	×
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)	○	○
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Invalid	○	○
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	0 : No reverse 1 : Reverse	○	○
U2ERE	Error signal output enable bit	Must be fixed to “0”	0 : Output disabled 1 : Output enabled	○	○

Figure 80: Serial I/O-related registers (4)

UART transmit/receive control register 2



Symbol	Address	When reset
UCON	03B0$_{16}$	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	Invalid	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Invalid	○	○
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 $\overline{CTS}$/$\overline{RTS}$ bit	0 : $\overline{CTS}$/$\overline{RTS}$ shared pin 1 : CTS/RTS separated	0 : $\overline{CTS}$/$\overline{RTS}$ shared pin 1 : $\overline{CTS}$/$\overline{RTS}$ separated	○	○
Nothing is assigned. This bit can neither be set nor reset. When read, its content is 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”.

Figure 81: Serial I/O-related registers (5)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table 23 and Table 24 list the specifications of the clock synchronous serial I/O mode. Figure 82 shows the UARTi transmit/receive mode register.

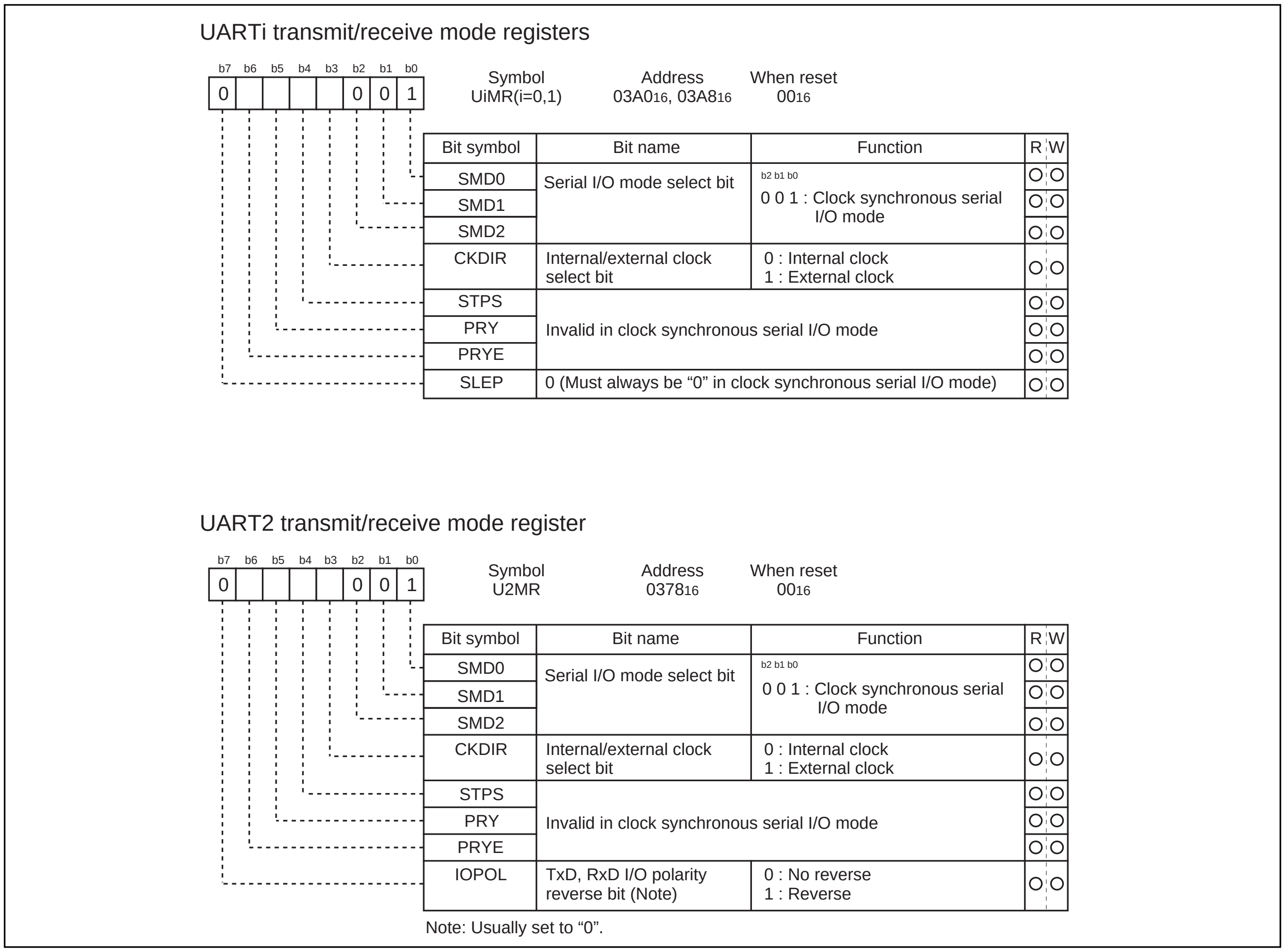
Table 23: Specifications of clock synchronous serial I/O mode (1)

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ = “0”) : fi=2(n+1) (Note 1) fi = f1, f8, f32 • When external clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ = “1”) : Input from CLKi pin (Note 2)
Transmission/reception control	• CTS function/RTS function/CTS, RTS function chosen to be invalid
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” • Furthermore, if external clock is selected, the following requirements must also be met: _ CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0”: CLKi input level = “H” _ CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”: CLKi 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” _ 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” • Furthermore, if external clock is selected, the following requirements must also be met: _ CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0”: CLKi input level = “H” _ CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”: CLKi input level = “L” • When transmitting _ Transmit interrupt cause select bit (bits 0, 1 at address 03B0$_{16}$, bit 4 at address 037D$_{16}$) = “0”: Interrupts requested when data transfer from UARTi _ Transmit interrupt cause select bit (bits 0, 1 at address 03B0$_{16}$, bit 4 at address 037D$_{16}$) = “1”: Interrupts requested when data transmission from
Error detection	• Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi

Note 1: “n” denotes the value 00$_{16}$ to FF$_{16}$ that is set to the UART bit rate generator.
Note 2: Maximum 5 Mbps.
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 is not set to “1”.

Table 24: Specifications of clock synchronous serial I/O mode (2)

Item	Specification
Select function	• CLK polarity selection Whether transmit data is output/input 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 simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection (UART1) (Note) UART1 transfer clock can be chosen by software to be output from one of the two pins set • Separate CTS/RTS pins (UART0) (Note) UART0 CTS and RTS pins each can be assigned to separate pins • Switching serial data logic (UART2) Whether to reverse data in writing to th etransmission buffer register or readin g the reception buffer register can be sele • Switching serial data logic (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.



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 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Must always be “0” in clock synchronous serial I/O mode)		○	○

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 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note)	0 : No reverse 1 : Reverse	○	○

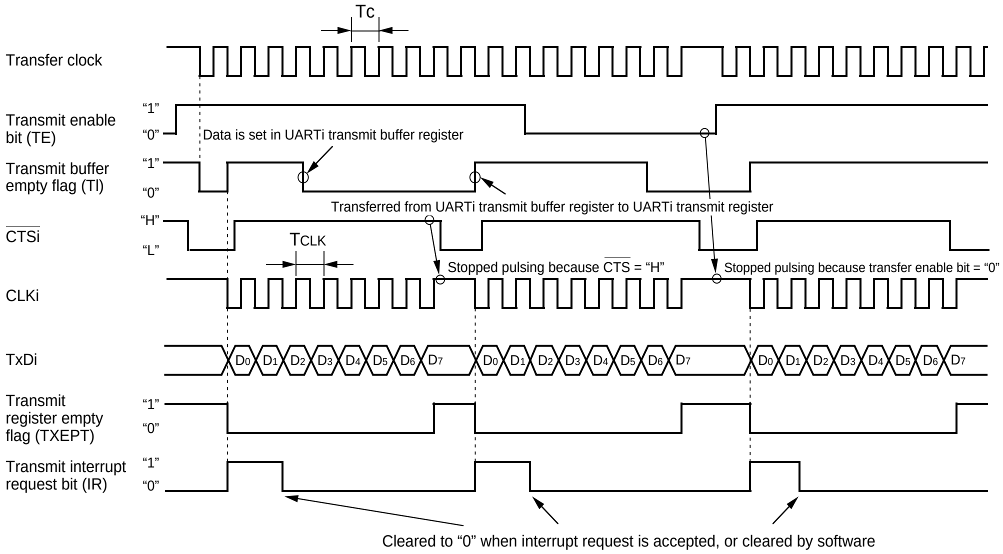
Note: Usually set to “0”.

Figure 82: UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 25 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 and the separate CTS/ RTS pins functions are not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a “H”. The typical clock synchronous timing diagrams are shown in Figure 83.

Table 25: Input/output pin functions in clock synchronous serial I/O mode

Pin name	Function	Method of selection
TxDi (P63, P67, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P62, P66, P71)	Serial data input	Port P62, P66, and P71 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 (P61, P65, P72)	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 P61, P65, and P72 direction register (bits 1 and 5 at address $03EE_{16}$, bit 2 at address $03EF_{16}$) = “0”
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ (P60,P64,P73)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” Port P60, P64 and P73 direction register (bits 0 and 4 at address $03EE_{16}$, bit 3 at address $03EF_{16}$) = “0”
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “1”
	Programmable I/O port	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “1”

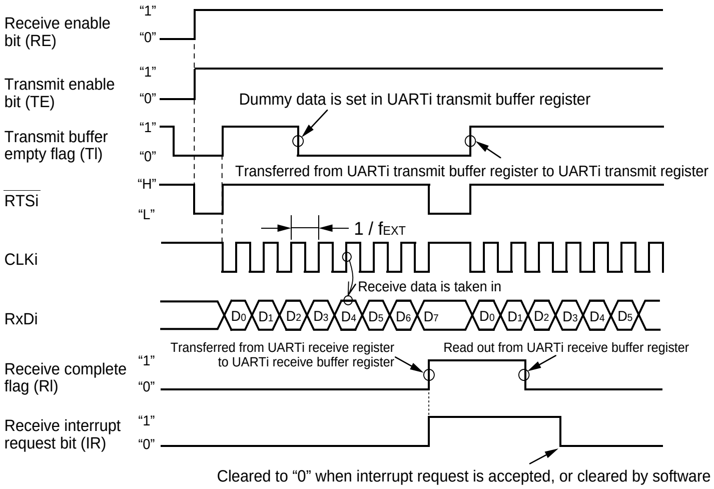

Tc
Transfer clock
Transmit enable bit (TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
CTSi
"H"
"L"
TCLK
Stopped pulsing because CTS = "H"
Stopped pulsing because transfer enable bit = "0"
CLKi
TxDi
D0 D1 D2 D3 D4 D5 D6 D7
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

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".

$T_C = T_{CLK} = 2(n + 1) / f_i$
f_i: frequency of BRGi count source (f_1, f_8, f_{32})
n: value set to BRGi

Example of receive timing (when external clock is selected)


Receive enable bit (RE)
"1"
"0"
Transmit enable bit (TE)
"1"
"0"
Dummy data is set in UARTi transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
RTSi
"H"
"L"
1 / fEXT
CLKi
Receive data is taken in
RxDi
D0 D1 D2 D3 D4 D5 D6 D7
Transferred from UARTi receive register to UARTi receive buffer register
Read out from UARTi receive buffer register
Receive complete flag (RI)
"1"
"0"
Receive interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

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".

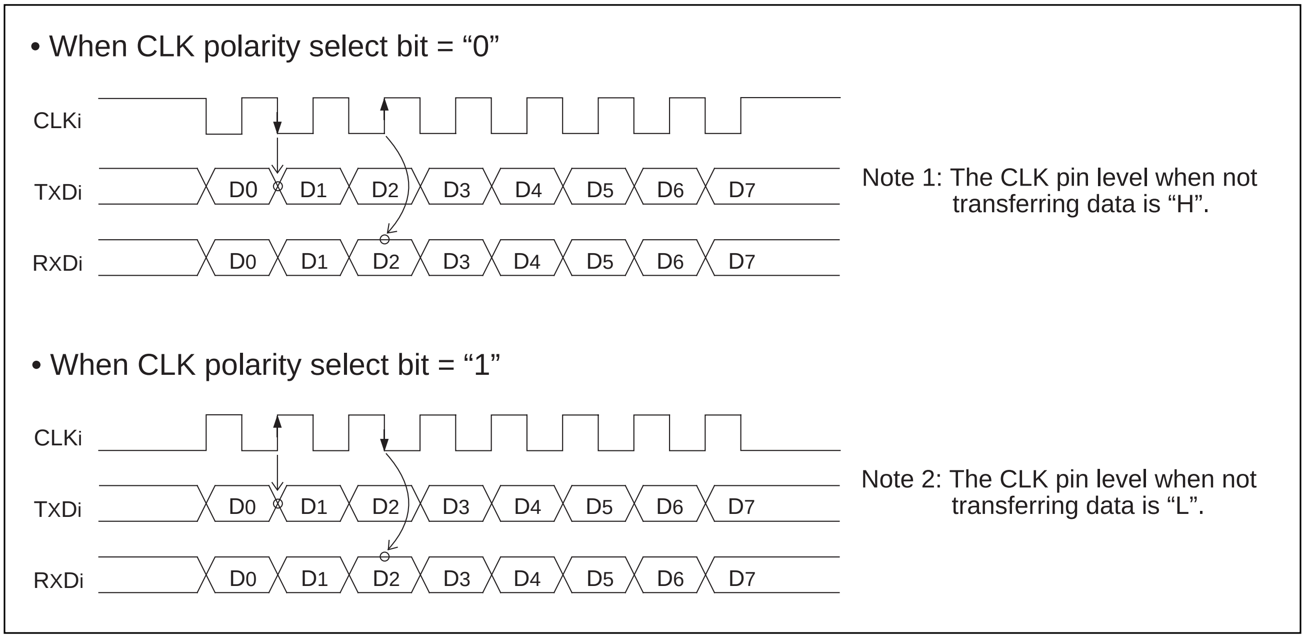
f_{EXT}: frequency of external clock

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 83: Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 84, the CLK polarity select bit (bit 6 at addresses $03A4_{16}$, $03AC_{16}$, $037C_{16}$) allows selection of the polarity of the transfer clock.


• When CLK polarity select bit = “0”

Note 1: The CLK pin level when not transferring data is “H”.

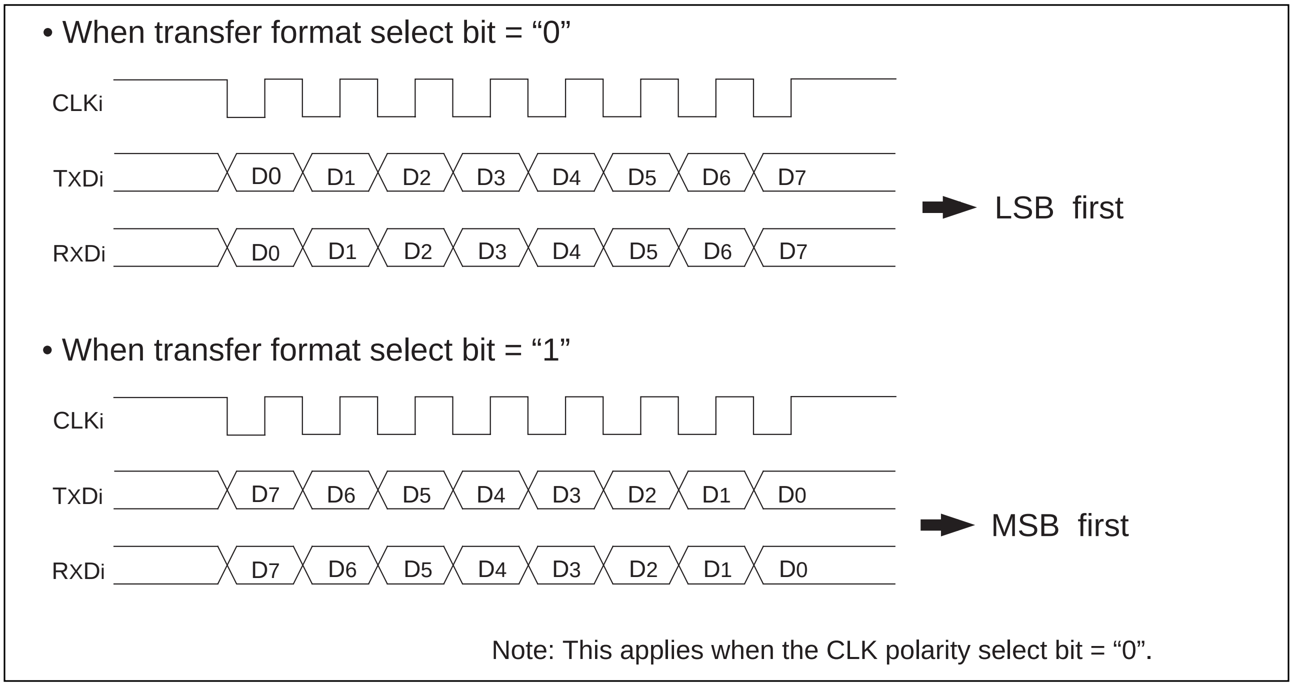
• When CLK polarity select bit = “1”

Note 2: The CLK pin level when not transferring data is “L”.

Figure 84: Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 85, 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”.


• When transfer format select bit = “0”

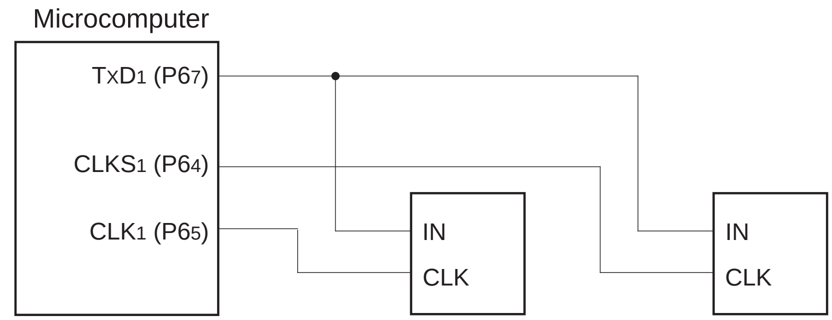
• When transfer format select bit = “1”

Note: This applies when the CLK polarity select bit = “0”.

Figure 85: 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 $03B0_{16}$). (SeeFigure 86.) The multiple pins function is valid only when the internal clock is selected for UART1. Note that when this function is selected, UART1 CTS/RTS function cannot be used.



Note: This applies when the internal clock is selected and transmission
is performed only in clock synchronous serial I/O mode.

Figure 86: 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 unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

(e) Separate CTS/RTS pins function (UART0)

This function works the same way as in the clock asynchronous serial I/O (UART) mode. The method of setting and the input/output pin functions are both the same, so refer to select function in the next section, “(2) Clock asynchronous serial I/O (UART) mode.” Note that this function is invalid if the transfer clock output from the multiple pins function is selected.

(f) 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 87 shows the example of serial data logic switch timing.

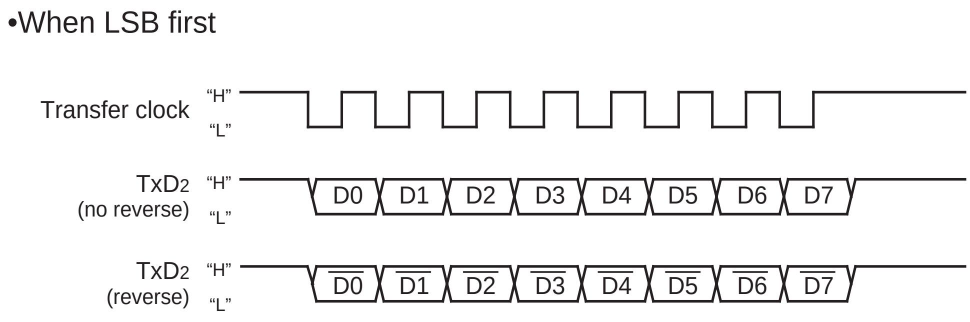


Figure 87: Serial data logic switch timing

(2) Clock asynchronous serial I/O (UART) mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Table 26 and Table 27 list the specifications of the UART mode. Figure 88 shows the UARTi transmit/receive mode register.

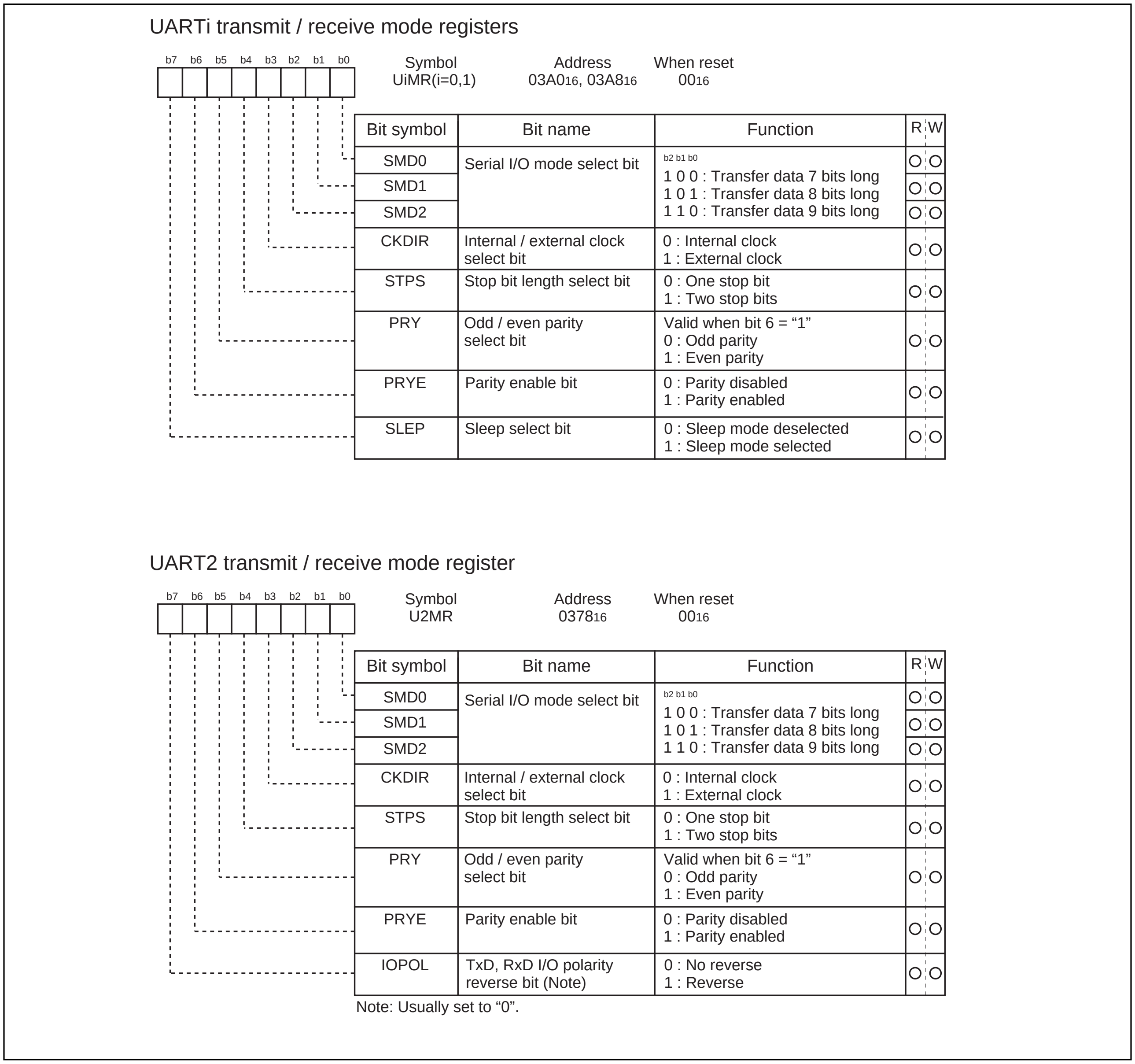
Table 26: Specifications of UART Mode (1)

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 = f1, f8, f32 • When external clock is selected (bit 3 at addresses $03A0_{16}$, 03A816, 0378_{16} =“1”) : fEXT/ 16(n+1)(Note 1) (Note 2)
Transmission/reception control	• CTS function/RTS function/CTS, RTS function chosen to be invalid
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 reques t 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) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out • 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: fEXT 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 is not set to “1”

Table 27: Specifications of UART Mode (2)

Item	Specification
Select function	• Separate CTS/RTS pins (UART0) UART0 CTS and RTS pins each can be assigned to separate pins • Sleep mode selection (UART0, UART1) This mode is used to transfer data to and from one of multiple slave micro-computers •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 This function is reversing TxD port output and RxD port input. All I/O data level is reversed.



UARTi transmit / receive mode registers

Symbol	Address	When reset
UiMR(i=0,1)	03A0$_{16}$, 03A8$_{16}$	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	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd / even parity select bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

UART2 transmit / receive mode register

Symbol	Address	When reset
U2MR	0378$_{16}$	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	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd / even parity select bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note)	0 : No reverse 1 : Reverse	○	○

Note: Usually set to “0”.

Figure 88: UARTi transmit/receive mode register in UART mode

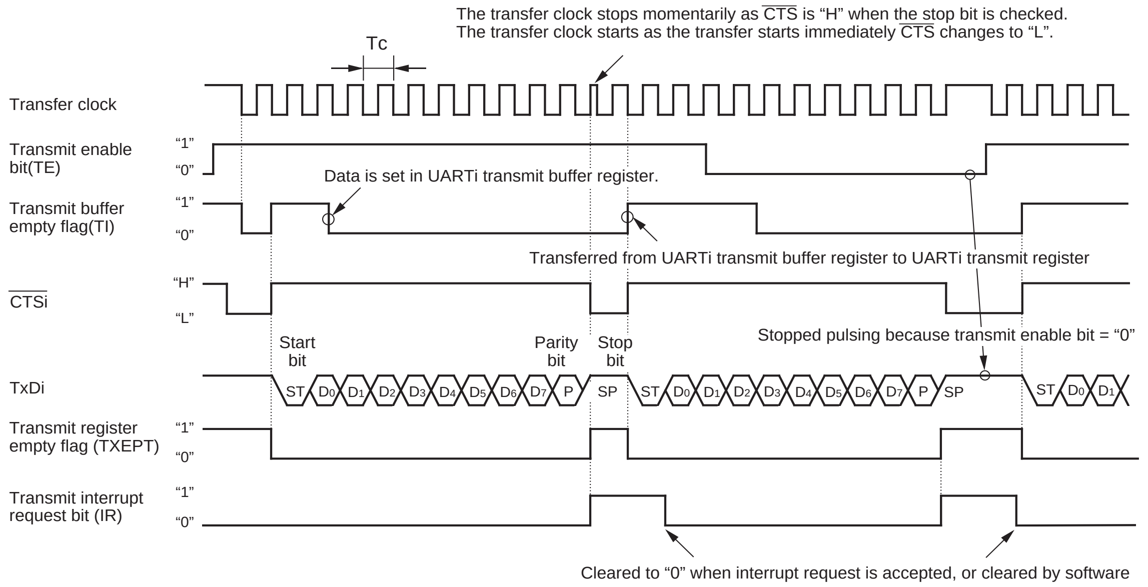
Table 28 lists the functions of the input/output pins during UART mode. This table shows the pin functions when the separate CTS/RTS pins function is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a “H”.

Table 28: Input/output pin functions in UART mode

Pin name	Function	Method of selection
TxDi (P63, P67, P70)	Serial data output	
RxDi (P62, P66, P71)	Serial data input	Port P62, P66, and P71 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 (P61, P65, P72)	Programmable I/O port	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 P61, P65, and P72 direction register (bits 1 and 5 at address 03EE$_{16}$, bit 2 at address 03EF$_{16}$) = “0”
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ (P60,P64,P73)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0” Port P60, P64 and P73 direction register (bits 0 and 4 at address 03EE$_{16}$, bit 3 at address 03EF$_{16}$) = “0”
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”
	Programmable I/O port	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”

Figure 89 and Figure 90 show the typical UART mode transmit and receive timing diagrams.

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

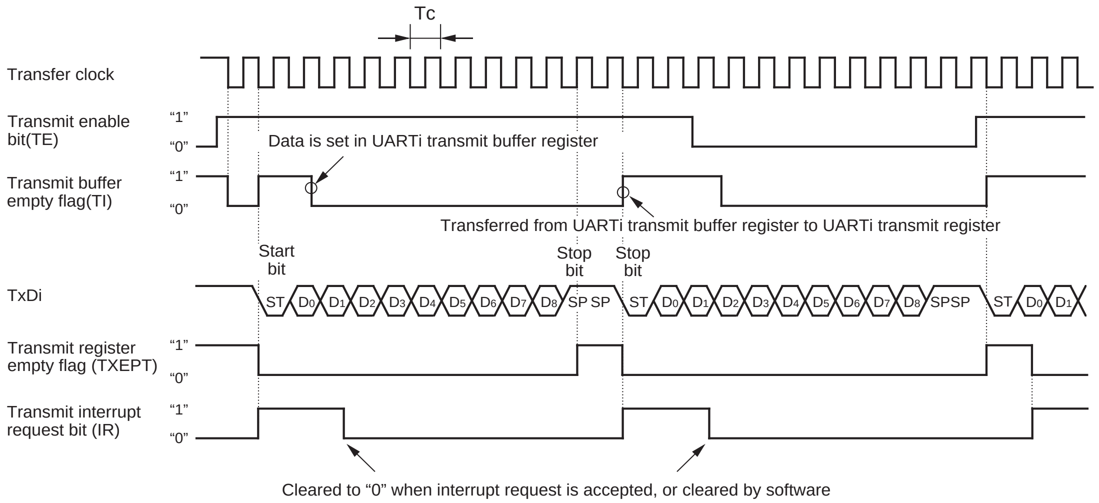

The transfer clock stops momentarily as CTS is "H" when the stop bit is checked.
The transfer clock starts as the transfer starts immediately CTS changes to "L".
Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register.
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
CTSi
"H"
"L"
Start bit
Parity bit
Stop bit
Stopped pulsing because transmit enable bit = "0"
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

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) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

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


Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Stop bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

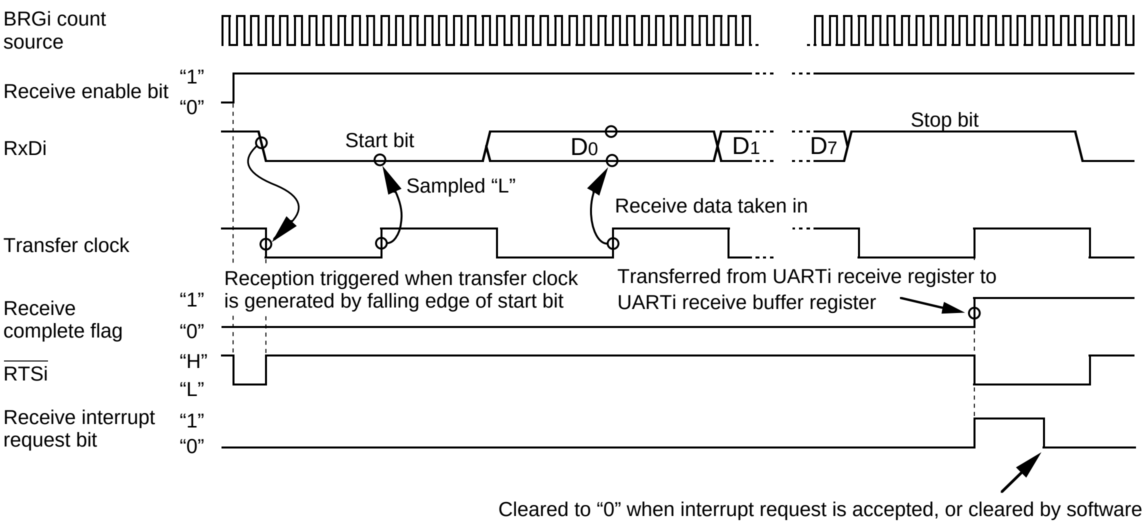
Shown in ( ) are bit symbols.

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) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 89: Typical transmit timings in UART mode

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 90: Typical receive timing in UART mode

(a) Separate CTS/RTS pins function (UART0)

With the separate CTS/RTS bit (bit 6 at address $03B0_{16}$) is set to "1", the unit outputs/inputs the CTS and RTS signals on different pins. (See Figure 91.) This function is valid only for UART0. Note that if this function is selected, the CTS/RTS function for UART1 cannot be used.

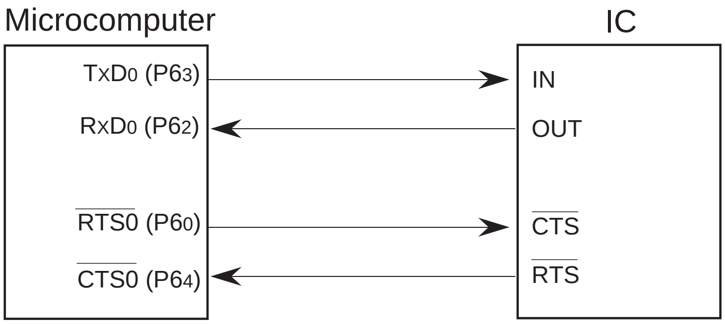


Figure 91: The separate CTS/RTS pins function usage

(b) Sleep mode (UART0, UART1)

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UARTi. The sleep mode is selected when the sleep select bit (bit 7 at addresses $03A0_{16}$, $03A8_{16}$) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

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

When the data logic select bit (bit 6 of address $037D_{16}$) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 92 shows the example of timing for switching serial data logic.

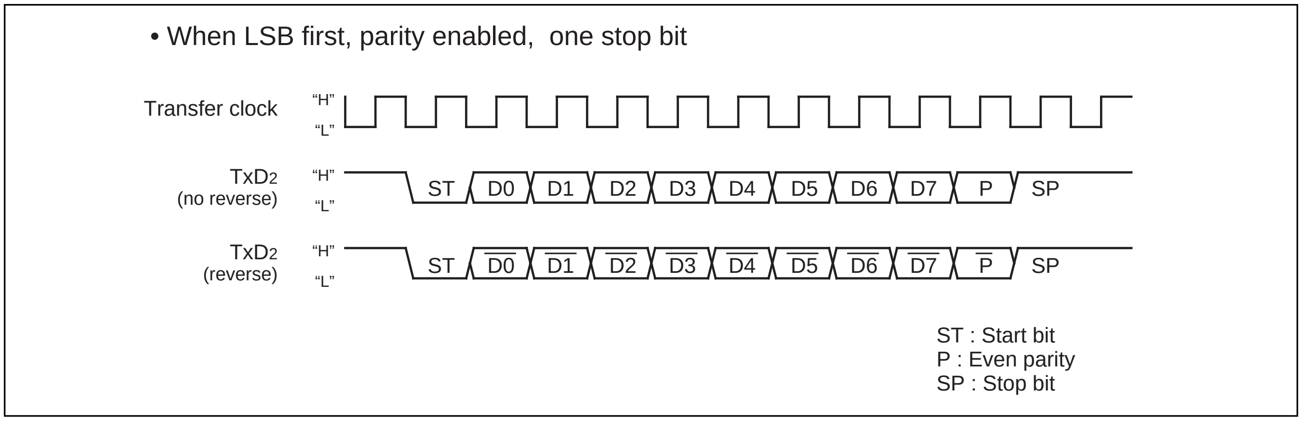


Figure 92: Timing for switching serial data logic

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

This function is to reverse 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 to reverse) for usual use.

(e) Bus collision detection function (UART2)

This function is to sample 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 93 shows the example of detection timing of a buss collision (in UART mode).

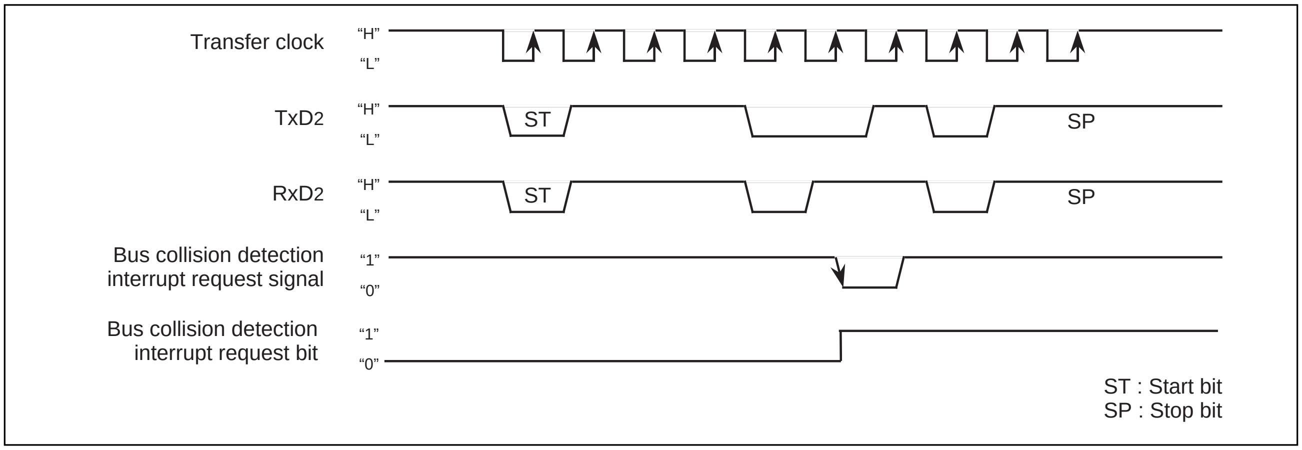

Transfer clock
"H"
"L"
TxD2
"H"
"L"
ST
SP
RxD2
"H"
"L"
ST
SP
Bus collision detection interrupt request signal
"1"
"0"
Bus collision detection interrupt request bit
"1"
"0"
ST : Start bit
SP : Stop bit

Figure 93: Detection timing of a bus collision (in UART mode)

(3) Clock-asynchronous serial I/O mode (compliant with the SIM interface)

The SIM interface is used for connecting the microcomputer with a memory card I/C or the like; adding some extra settings in UART2 clock-asynchronous serial I/O mode allows the user to effect this function. Table 29 shows the specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface). Figure 94 shows the typical transmit/receive timing in UART mode.

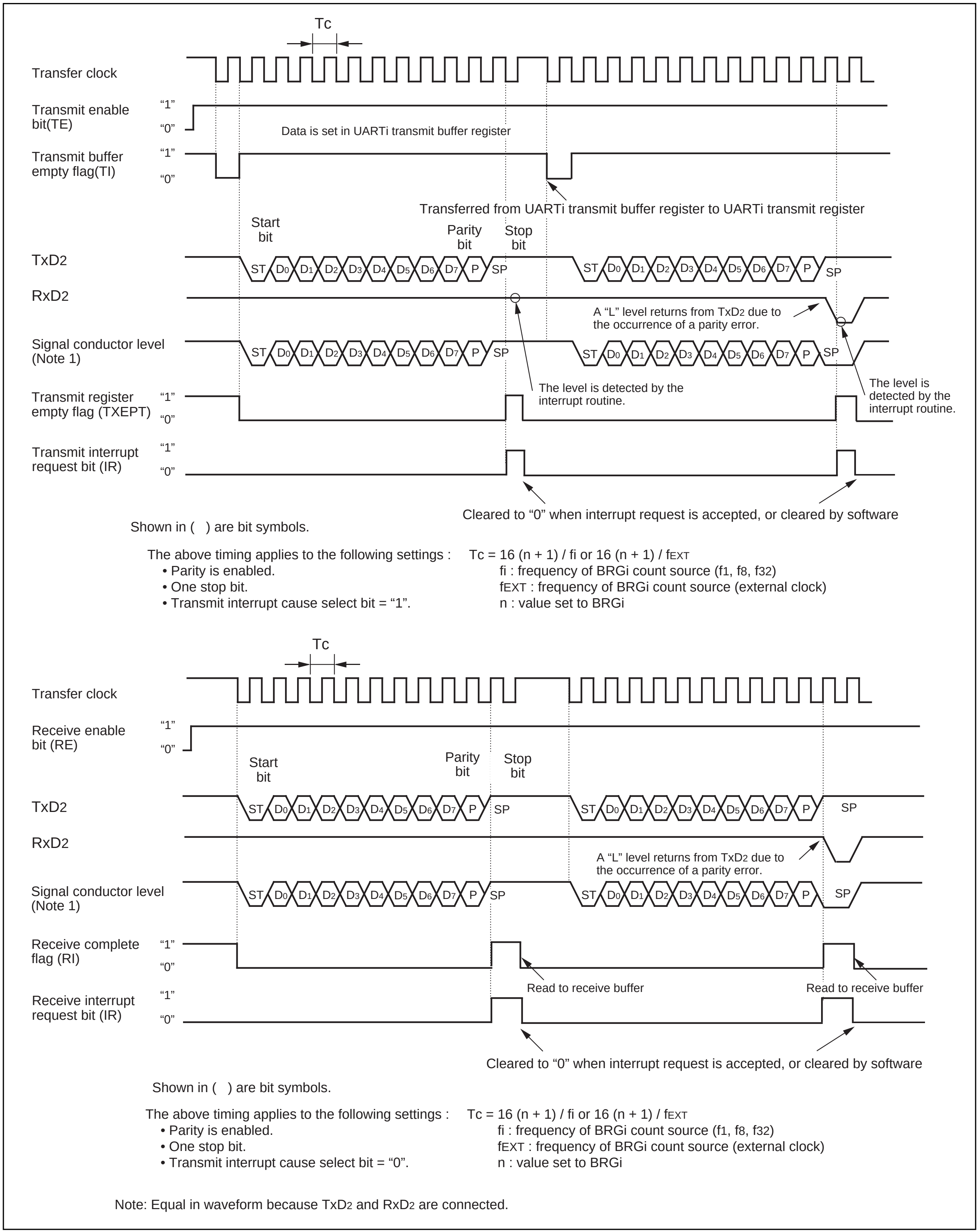
Table 29: Specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface)

Item	Specification
Transfer data format	• Transfer data 8-bit UART mode (bit 2 through bit 0 of address 0378_{16} = “1012”) • One stop bit (bit 4 of address 0378_{16} = “0”) • With the direct format chosen Set parity to “even” (bit 5 and bit 6 of address 0378_{16} = “1” and “1” respectively) 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 and bit 6 of address 0378_{16} = “0” and “1” respectively) 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, f8, f32 • With an external clock chosen (bit 3 of address 0378_{16} = “1”) : fEXT / 16 (n+1) (Note 1) (Note 2)
Transmission / reception control	• Disable the CTS and RTS function (bit 4 of address $037C_{16}$ = “1”)
Other settings	• The sleep mode select function is not available for UART2 • 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 • When transmitting When data transmission from the UART2 transfer 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 3) • 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 UARTi bit rate generator.

Note 2: fEXT is input from the CLK2 pin.

Note 3: If an overrun error occurs, the UART2 receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to “1”.


Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Parity bit
Stop bit
TxD2
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
RxD2
A "L" level returns from TxD2 due to the occurrence of a parity error.
Signal conductor level (Note 1)
Transmit register empty flag (TXEPT)
"1"
"0"
The level is detected by the interrupt routine.
The level is detected by the interrupt routine.
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software
Tc
Transfer clock
Receive enable bit (RE)
"1"
"0"
Start bit
Parity bit
Stop bit
TxD2
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
RxD2
A "L" level returns from TxD2 due to the occurrence of a parity error.
Signal conductor level (Note 1)
Receive complete flag (RI)
"1"
"0"
Read to receive buffer
Read to receive buffer
Receive interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

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 or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "0".

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Note: Equal in waveform because TxD_2 and RxD_2 are connected.

Figure 94: Typical transmit/receive timing in UART mode (compliant with the SIM interface)

(a) Function for outputting a parity error signal

With the error signal output enable bit (bit 7 of address 037D$_{16}$) assigned “1”, you can output an “L” level from the TxD2 pin when a parity error is detected. In step with this function, the generation timing of a transmission completion interrupt changes to the detection timing of a parity error signal. Figure 95 shows the output timing of the parity error signal.

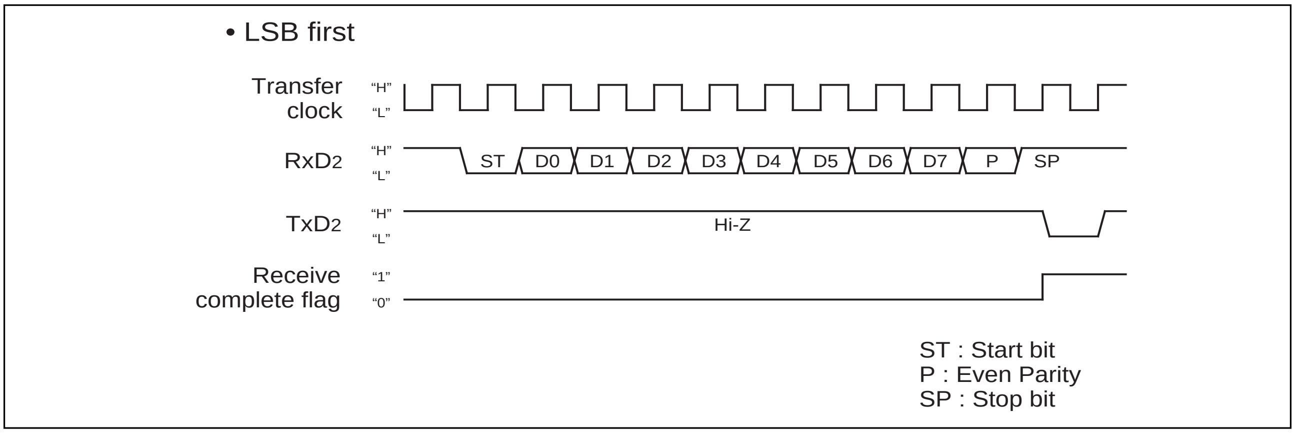


Figure 95: Output timing of the parity error signal

(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 96 shows the SIM interface format.

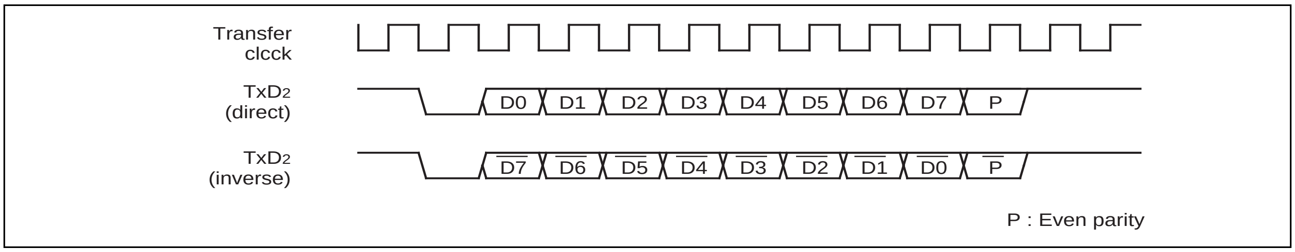


Figure 96: SIM interface format

Figure 97 shows the example of connecting the SIM interface with TxD2 and RxD2 pulled up.

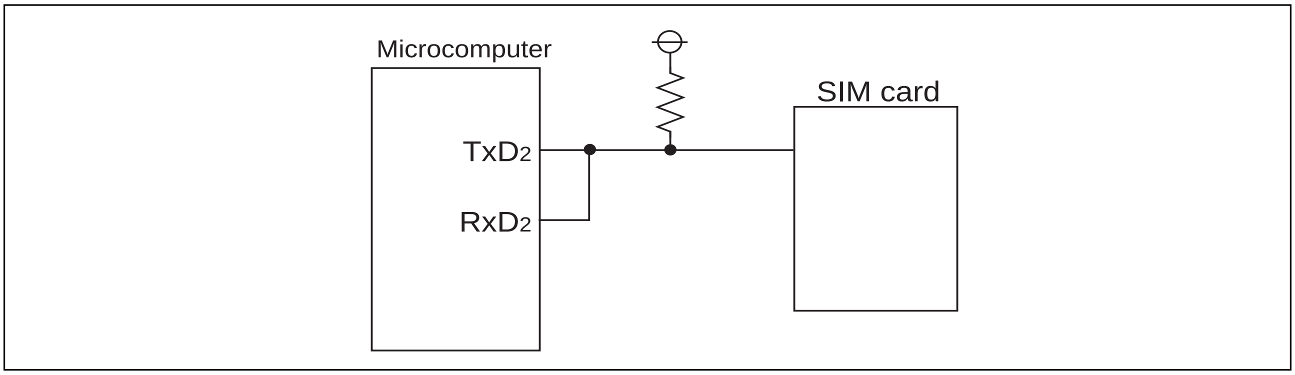


Figure 97: Connecting the SIM interface

2.24 A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P100 to P107 function as the analog signal input pins. 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 (VREF) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of $03D7_{16}$ to connect VREF.

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 30 shows the performance of the A-D converter. Figure 98 shows the block diagram of the A-D converter, and Figure 99 and Figure 100 show the A-D converter-related registers.

r

Table 30: Performance of A-D Converter

Item	Performance	
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)	
Analog input voltage (Note 1)	0V to AVCC (VCC)	
Operating clock fAD (Note 2)	VCC = 5V	fAD/divide-by-2 or fAD/divide-by-4 or fAD, fAD=f(XIN)
Resolution	8-bit or 10-bit (selectable)	
Absolute precision	VCC = 5V	• Without sample and hold function 3LSB • With sample and hold function (8-bit resolution) 2LSB • With sample and hold function (10-bit resolution)AN0 to AN7 input 3LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1	
Analog input pins	8pins (AN0 to AN7)	
A-D conversion start condition	•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 $\overline{\text{AD}}_{\text{TRG}}$/P87 input changes from “H” to “L”	
Conversion speed per pin	•Without sample and hold function 8-bit resolution: 49 ϕAD cycles, 10-bit resolution: 59 ϕAD cycles • With sample and hold function 8-bit resolution: 28 ϕAD cycles, 10-bit resolution: 33 ϕAD cycles	
Note 1 Does not depend on use of sample and hold function. Note 2 Without sample and hold function, set the 0AD frequency to 250 kHz minimum; with sample and hold function, set the 0AD frequency to 1 MHz minimum.		

A-D Converter

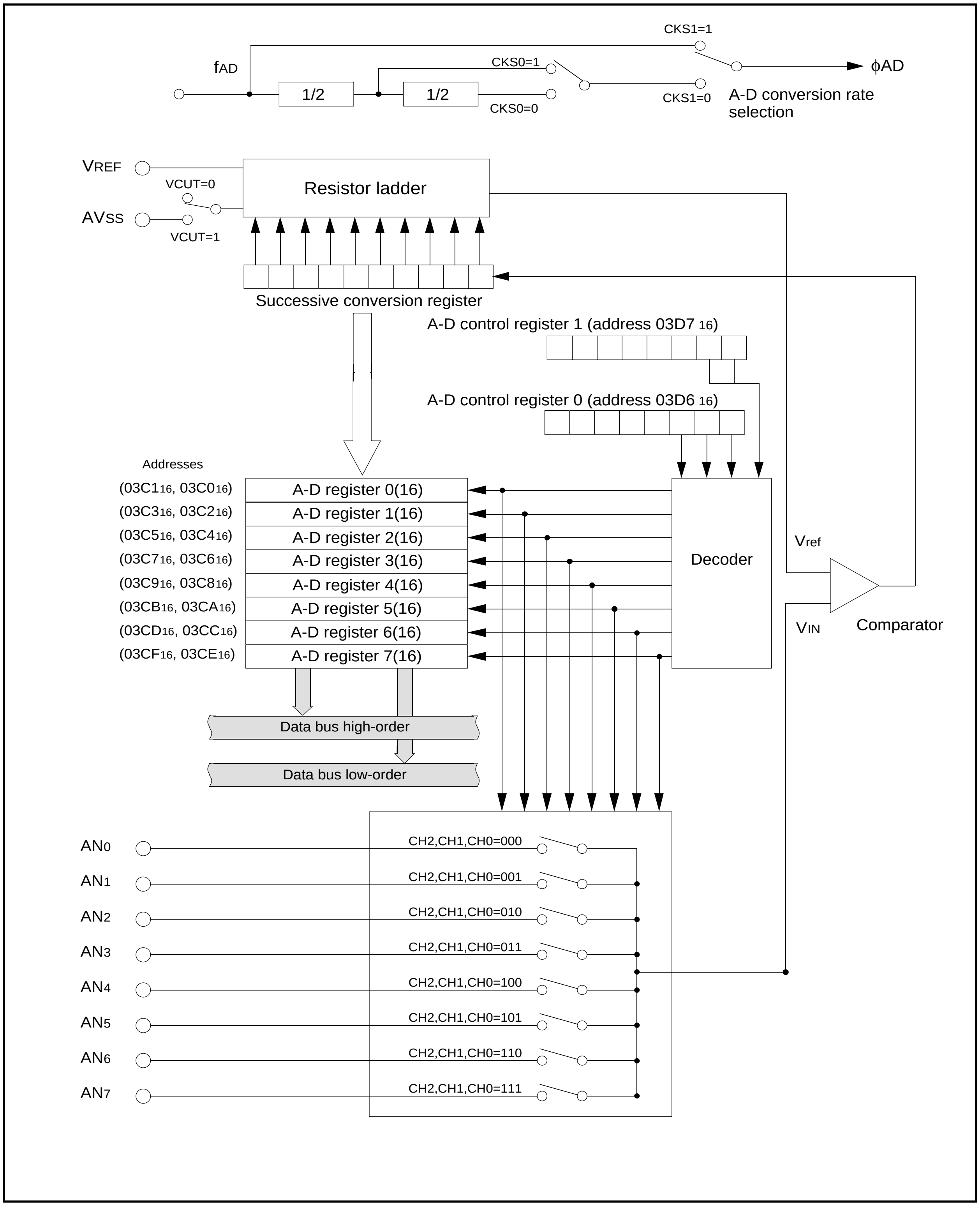


Figure 98: Block diagram of A-D converter

A-D control register 0 (Note 1)

b7 b6 b5 b4 b3 b2 b1 b0

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 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1 (Note 2)	○	○
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	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

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)

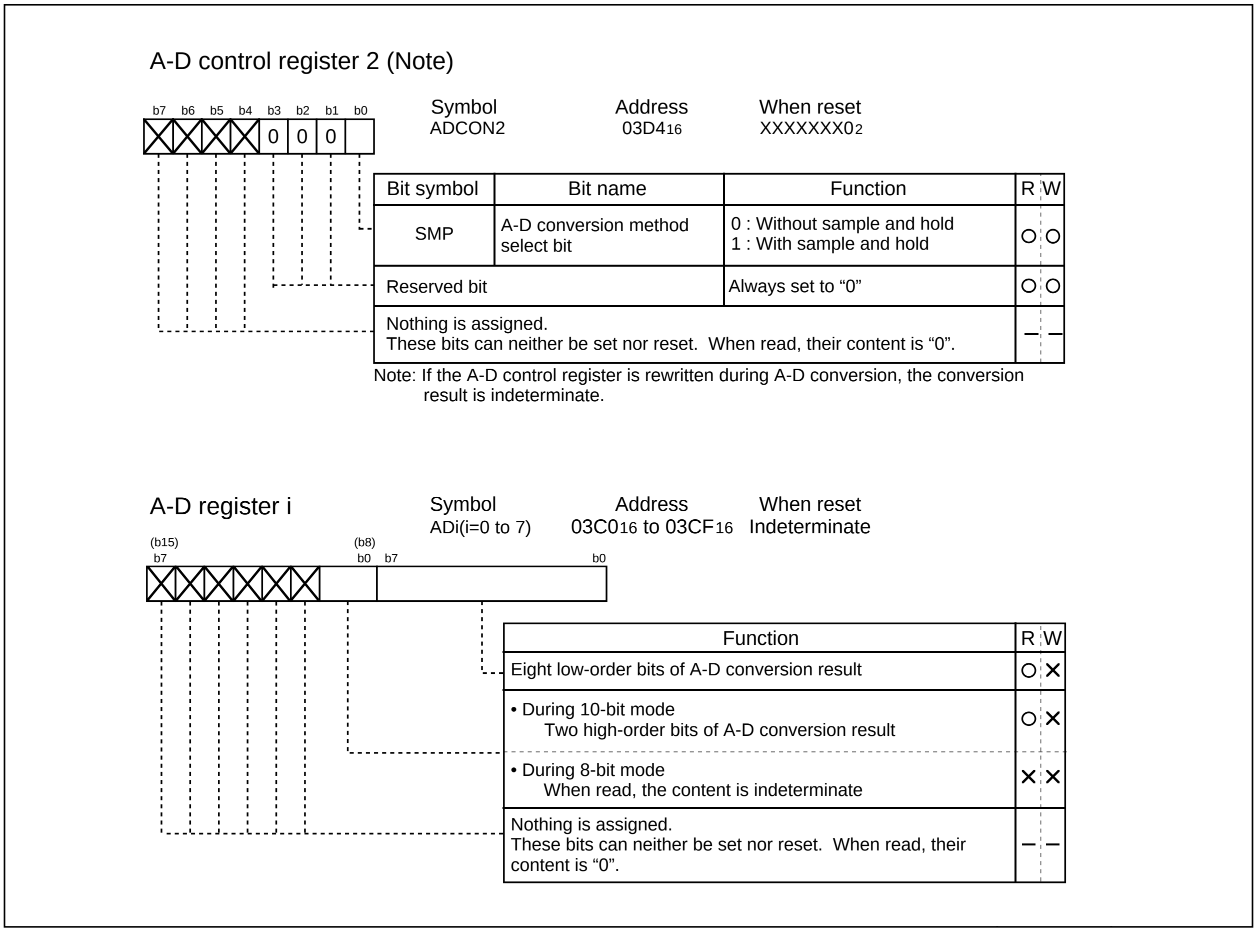
b7 b6 b5 b4 b3 b2 b1 b0

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 and repeat sweep mode 0 are 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		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)	○	○
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 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○

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

Figure 99: A-D converter-related registers (1)



A-D control register 2 (Note)

Symbol	Address	When reset
ADCON2	$03D4_{16}$	$XXXXXXX0_2$

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 bit		Always set to “0”	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their content is “0”.			—	—

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

A-D register i

Symbol	Address	When reset
ADi(i=0 to 7)	$03C0_{16}$ to $03CF_{16}$	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. These bits can neither be set nor reset. When read, their content is “0”.	—	—

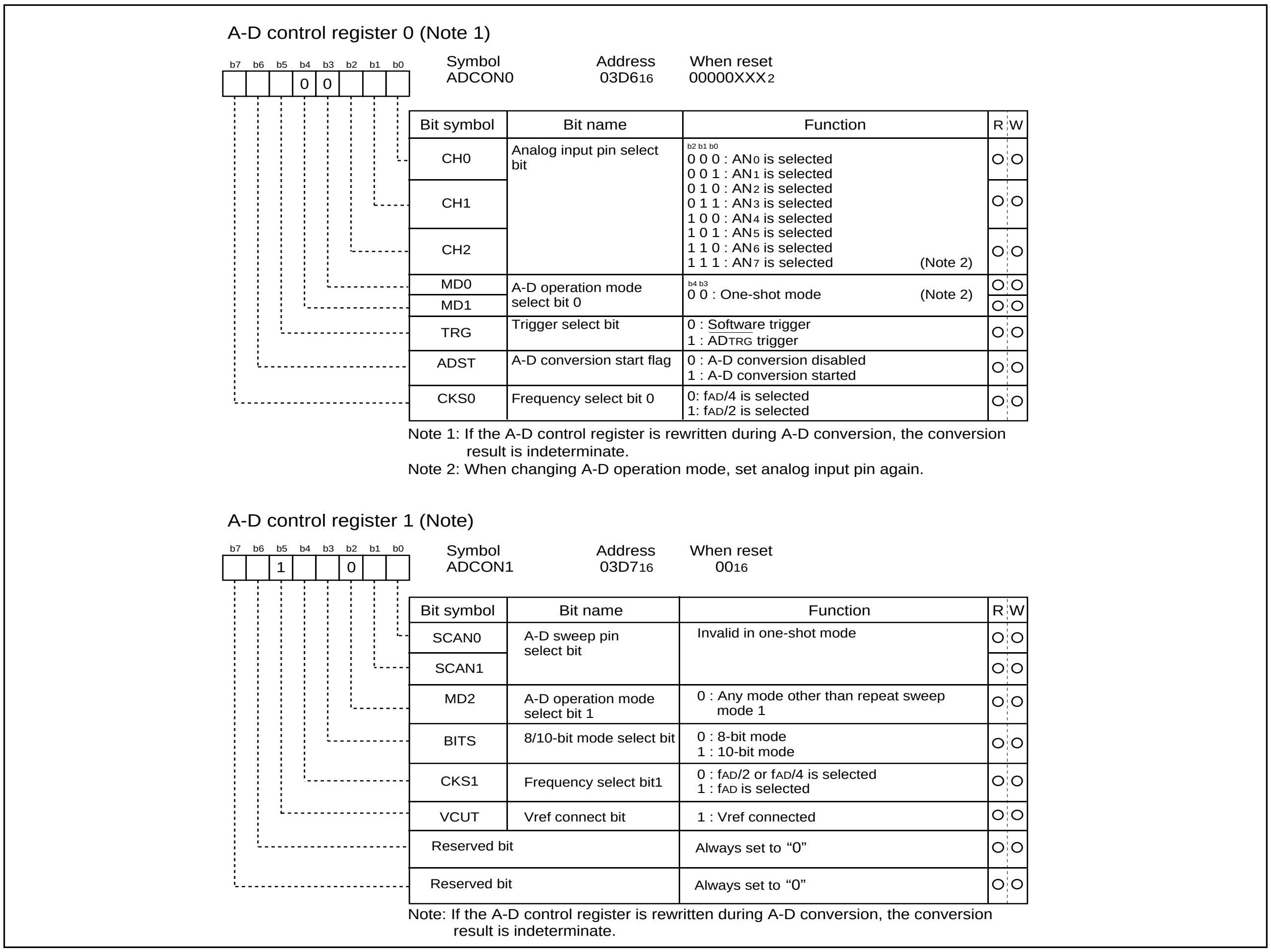
Figure 100: A-D converter-related registers (2)

(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion.Table 31 shows the specifications of one-shot mode. Figure 101 shows the A-D control register in one-shot mode.

Table 31: One-shot mode specification

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing “1” to 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 A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN0 to AN7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected 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 : $\overline{AD_{TRG}}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0: f_{AD}/4 is selected 1: f_{AD}/2 is selected	○	○

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)

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	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	0 : f_{AD}/2 or f_{AD}/4 is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○

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

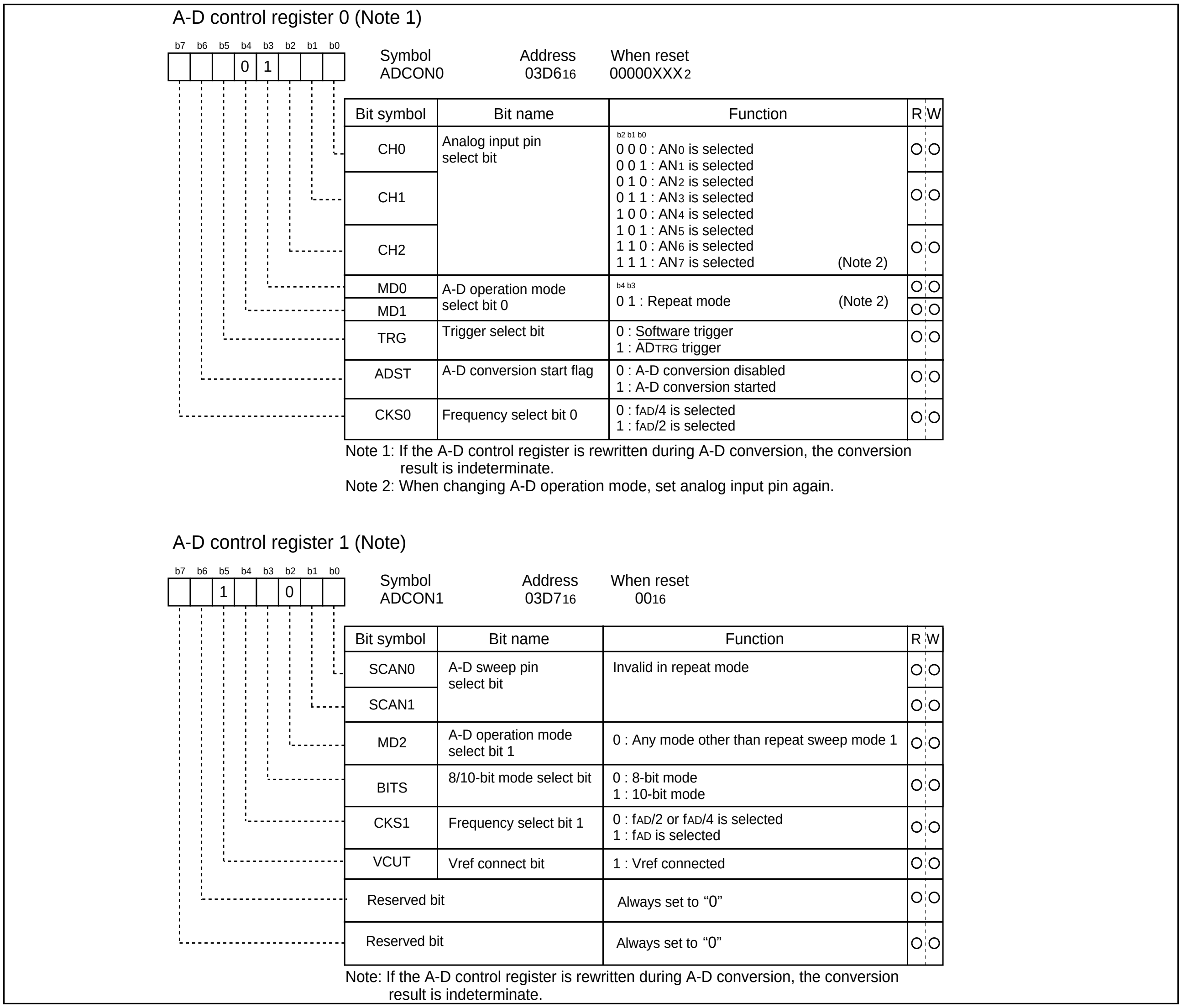
Figure 101: A-D conversion register in one-shot mode

(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. Table 32 shows the specifications of repeat mode. Figure 102 shows the A-D control register in repeat mode.

Table 32: Repeat sweep mode 0 specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Star condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN0 to AN7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin



A-D control register 0 (Note 1)

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 1 : Repeat mode (Note 2)	○	○
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 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

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)

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 repeat mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○

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

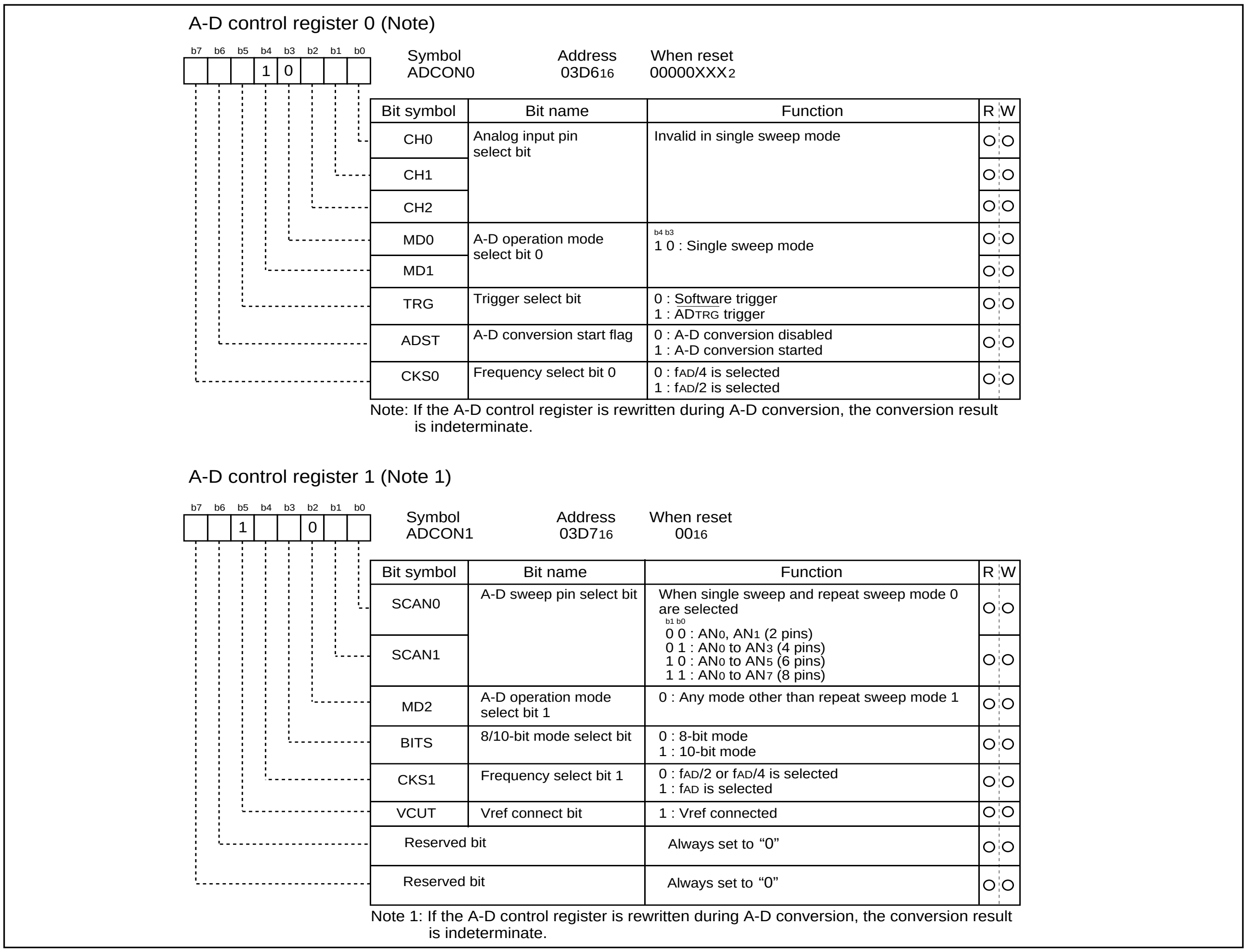
Figure 102: A-D conversion register in repeat mode

(3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bit are used for one-by-one A-D conversion. Table 33 shows the specifications of single sweep mode. Figure 103 shows the A-D control register in single sweep mode.

Table 33: Single sweep mode specification

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for one-by-one A-D conversion
Start condition	Writing “1” to A-D converter 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 A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin



A-D control register 0 (Note)

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

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 : $\overline{AD_{TRG}}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

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

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 : 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	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○

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

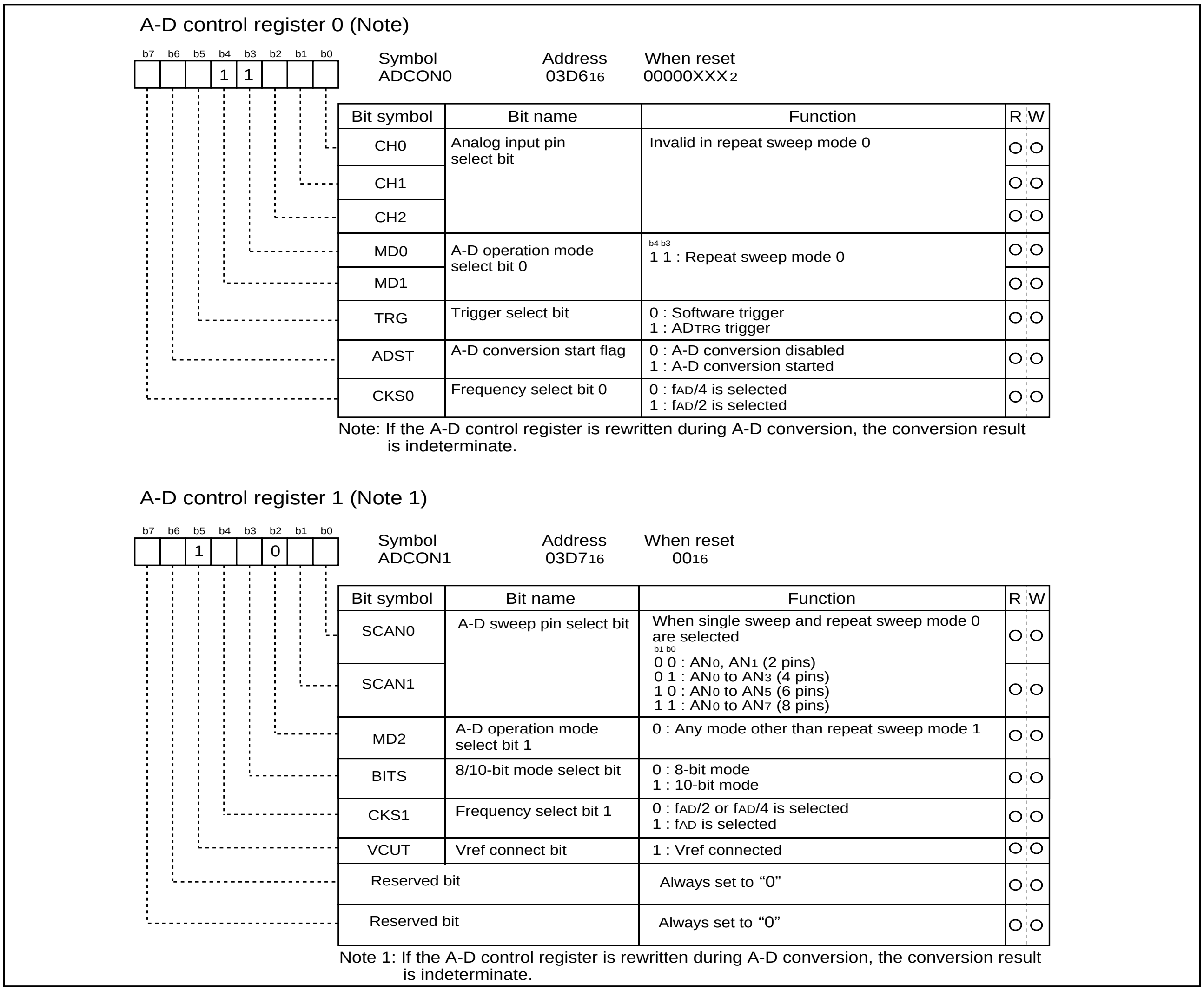
Figure 103: A-D conversion register in single sweep mode

(4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. Table 34 shows the specifications of repeat sweep mode 0. Figure 104 shows the A-D control register in repeat sweep mode 0.

Table 34: Repeat sweep mode 0 specifications

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for repeat sweep A-D conversion
Start condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



A-D control register 0 (Note)

Symbol	Address	When reset
ADCON0	03D6₁₆	00000XXX₂

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 : ADTRG trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

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	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 : 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	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○

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

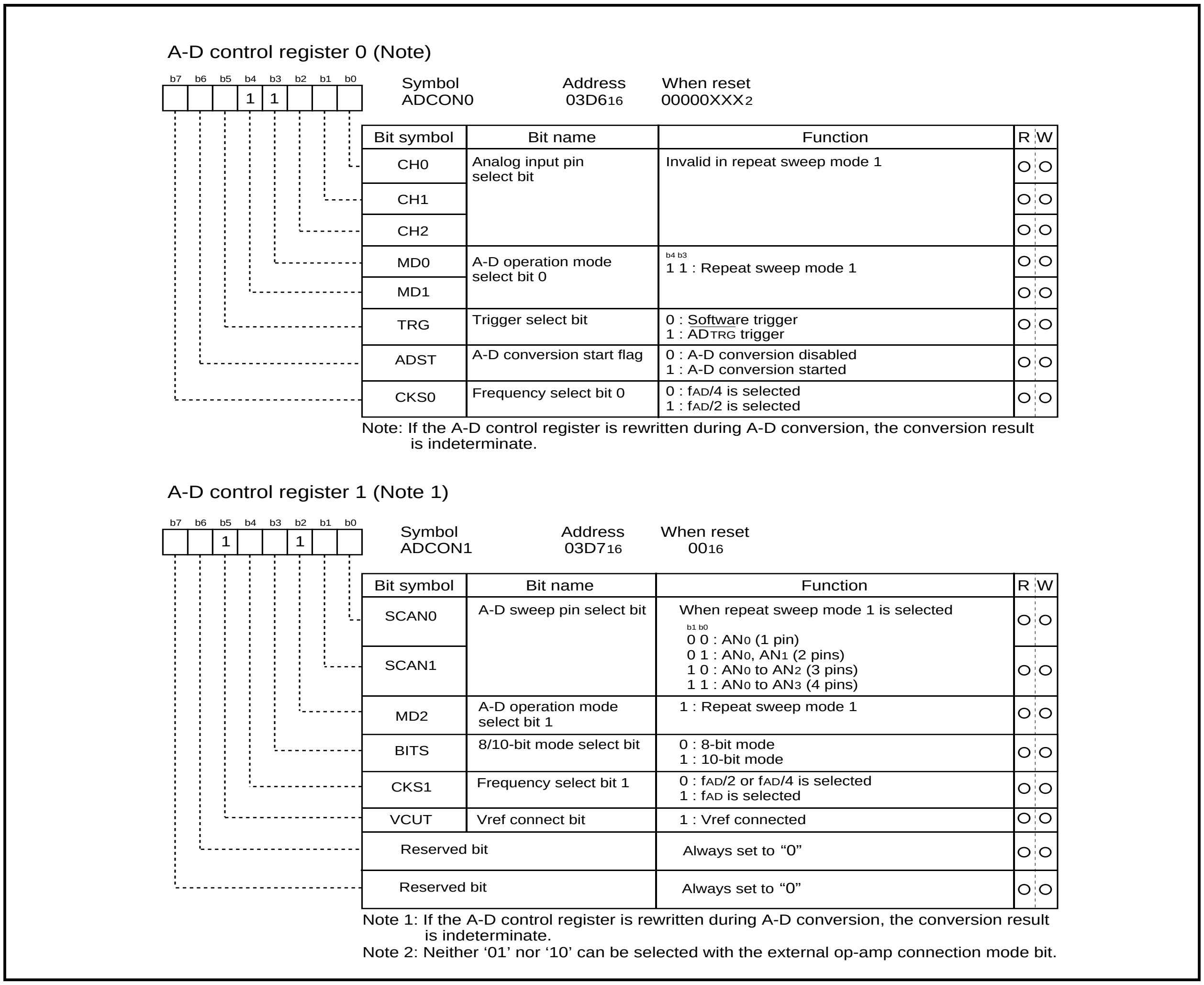
Figure 104: A-D conversion register in repeat sweep mode 0

(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 bit. Table 35 shows the specifications of repeat sweep mode 1. Figure 105 show the A-D control in repeat sweep mode 1.

Table 35: Repeat sweep mode 1 specification

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin select bit Example : AN0 selected AN0 AN1 AN0 AN2 AN0 AN3, etc
Start condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN0 (1 pin), AN0 and AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (4 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



A-D control register 0 (Note)

Symbol ADCON0 Address 03D6

Symbol	Address	When reset
ADCON0	03D616	00000XXX2

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 : ADTRG trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

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	Address	When reset
ADCON1	03D716	0016

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	1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: Neither ‘01’ nor ‘10’ can be selected with the external op-amp connection mode bit.

Figure 105: A-D conversion register in repeat sweep mode 1

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 with 8-bit resolution and 33 ϕ AD with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

2.25 CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) calculation circuit detects an error in data blocks. The microcomputer uses a generator polynomial of CRC_CCITT (X16 + X12 + X5 + 1) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of 8 bits. The CRC code is set in a CRC data register each time one byte of data is transferred to a CRC input register after writing an initial value into the CRC data register. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure 106 shows the block diagram of the CRC circuit. Figure 107 shows the CRC-related registers.

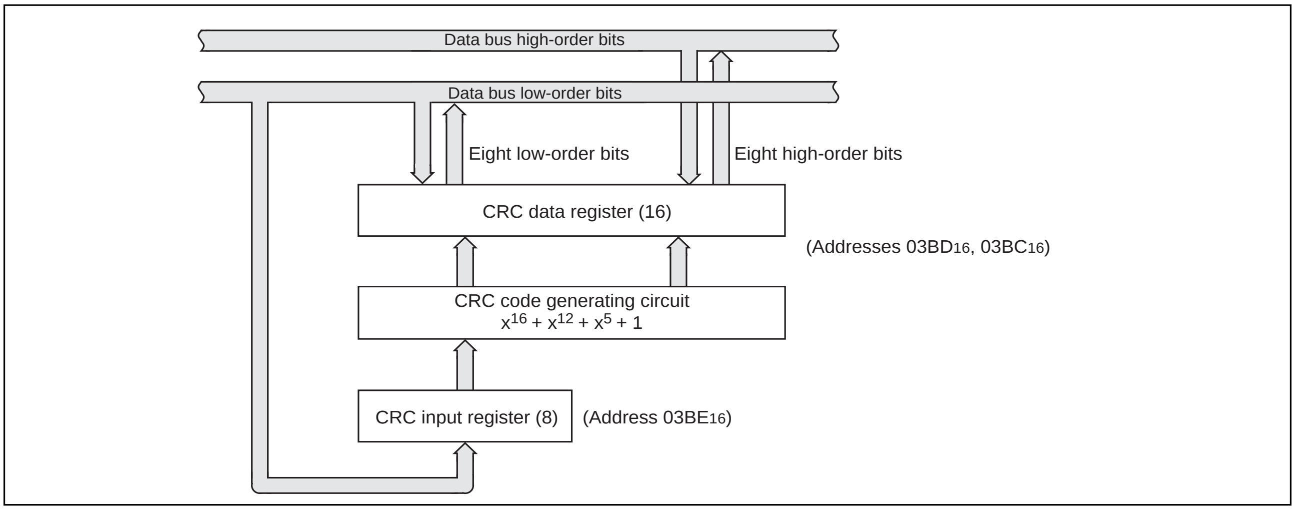
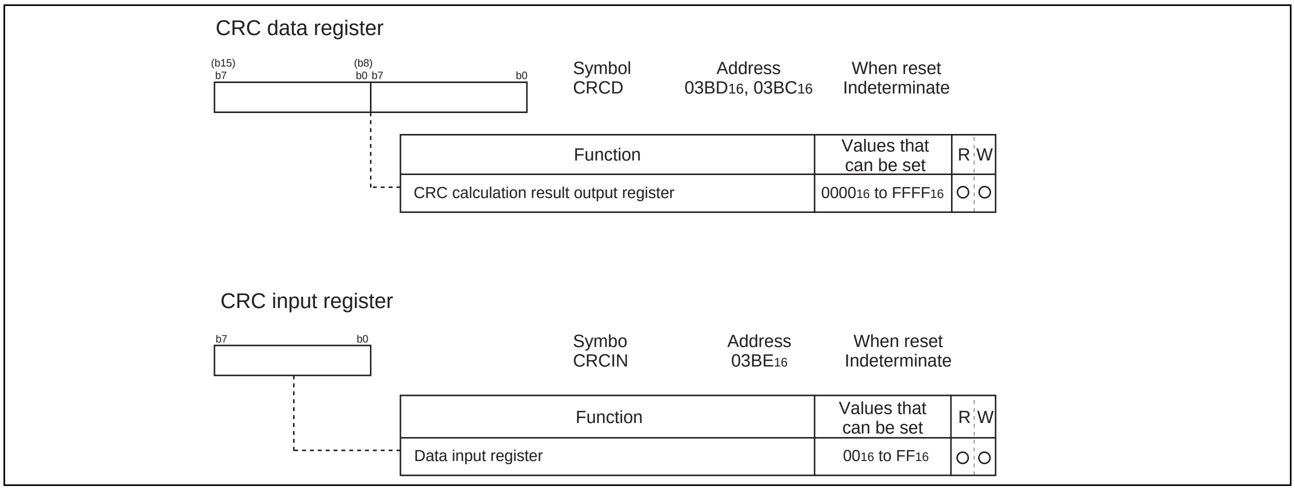


Figure 106: Block diagram of CRC circuit



CRC data register

Symbol	Address	When reset
CRCD	$03BD_{16}$, $03BC_{16}$	Indeterminate

Function	Values that can be set	R	W
CRC calculation result output register	0000_{16} to $FFFF_{16}$	○	○

CRC input register

Symbo	Address	When reset
CRCIN	$03BE_{16}$	Indeterminate

Function	Values that can be set	R	W
Data input register	00_{16} to FF_{16}	○	○

Figure 107: CRC-related registers

2.26 Programmable I/O Ports

There are 63 programmable I/O ports: P0 to P3, P6 to P8 (excluding P85), 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. P85 is an input-only port and has no built-in pull-up resistance.

Figure 108 and Figure 109 show the programmable I/O ports.

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. Unused I/O pins can be terminated as shown in Figure 114 and Table 36 .

(1) Direction registers

Figure 110 shows the 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: There is no direction register bit for P85.

(2) Port registers

Figure 111 shows the port registers.

These registers are used to write and read 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 port registers corresponds one for one to each I/O pin.

(3) Pull-up control registers

Figure 112 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 for input.

(4) High drive capacity registers

Figure 113 shows the Port 2 and PWM drive capacity registers. Port 2 can be configured to drive an LED by increasing the drive strength of the corresponding bit's N-channel transistor. Each PWM output (TA0OUT~TA4OUT) can be configured for high-drive capability by increasing the drive strength of the corresponding bits.

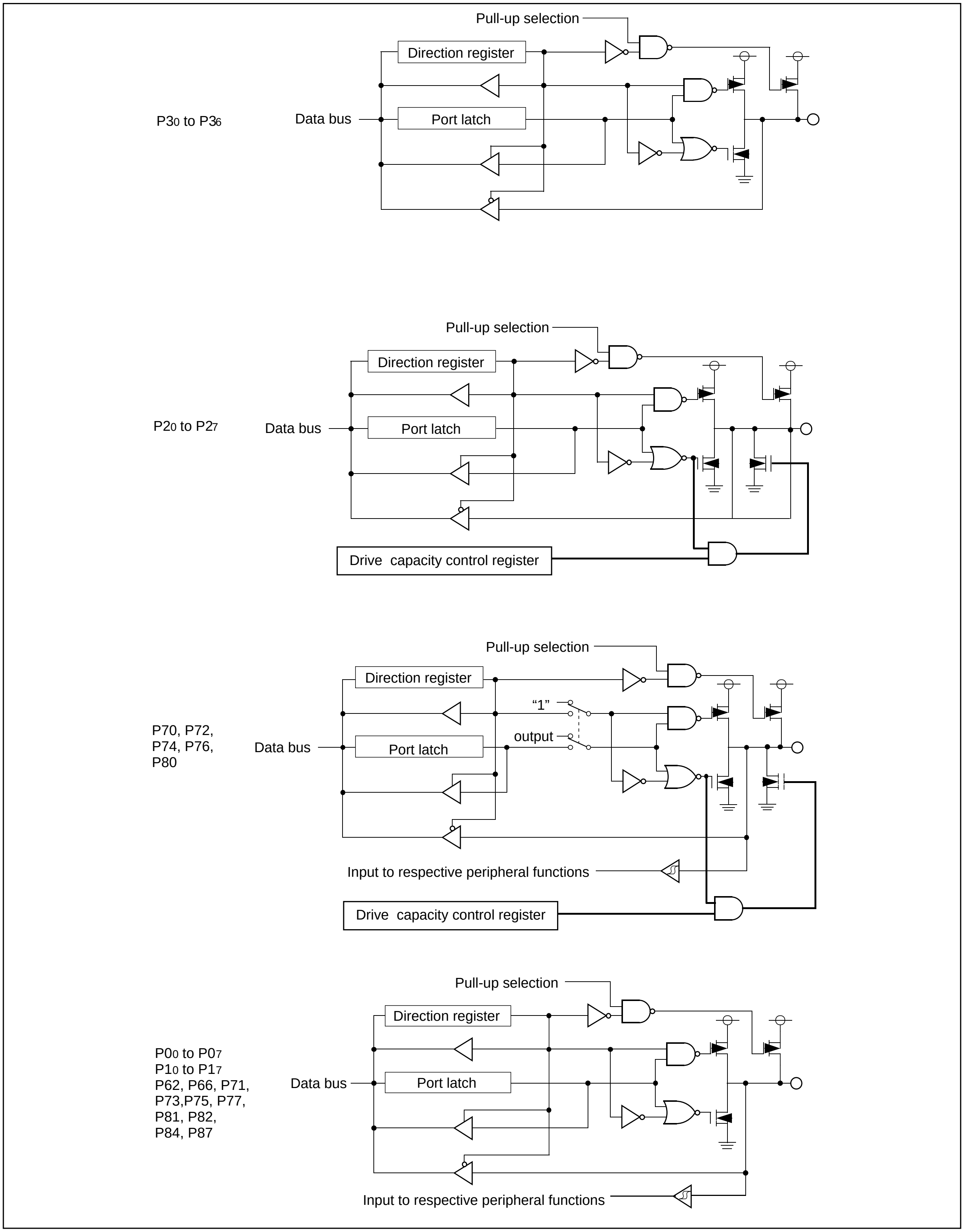


Figure 108: Programmable I/O ports (1)

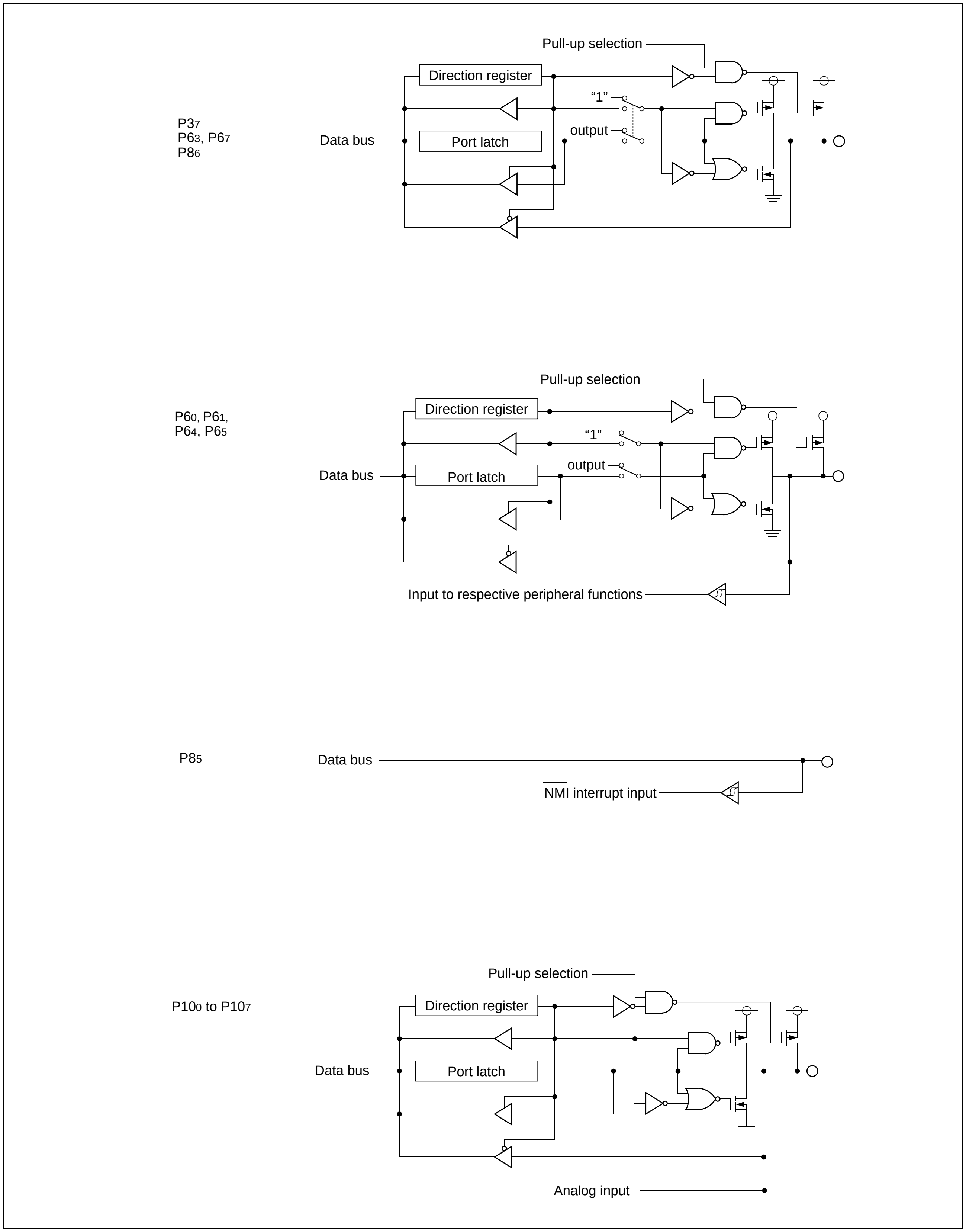
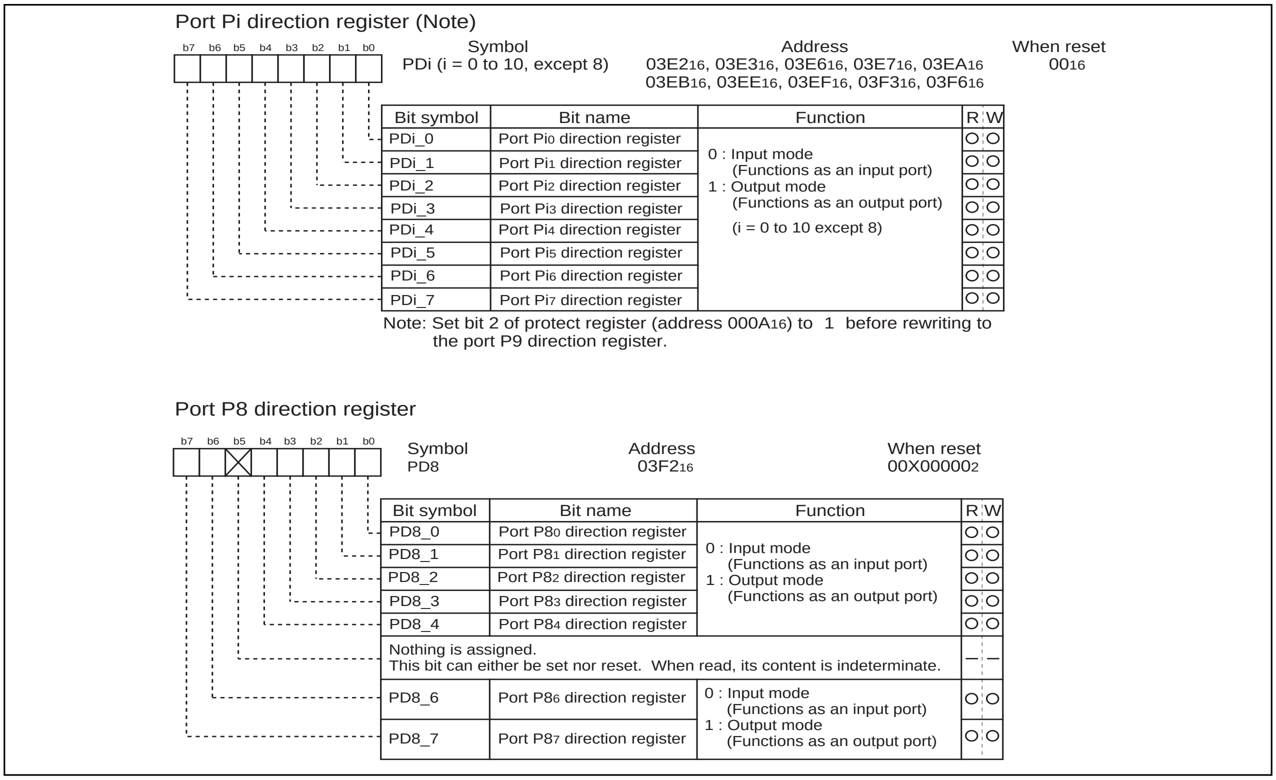


Figure 109: Programmable I/O ports (2)



Port Pi direction register (Note)

Symbol	Address	When reset
PDi (i = 0 to 10, except 8)	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$, 03EA$_{16}$ 03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$, 03F3$_{16}$, 03F6$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi_0 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 10 except 8)	○	○
PDi_1	Port Pi_1 direction register		○	○
PDi_2	Port Pi_2 direction register		○	○
PDi_3	Port Pi_3 direction register		○	○
PDi_4	Port Pi_4 direction register		○	○
PDi_5	Port Pi_5 direction register		○	○
PDi_6	Port Pi_6 direction register		○	○
PDi_7	Port Pi_7 direction register		○	○

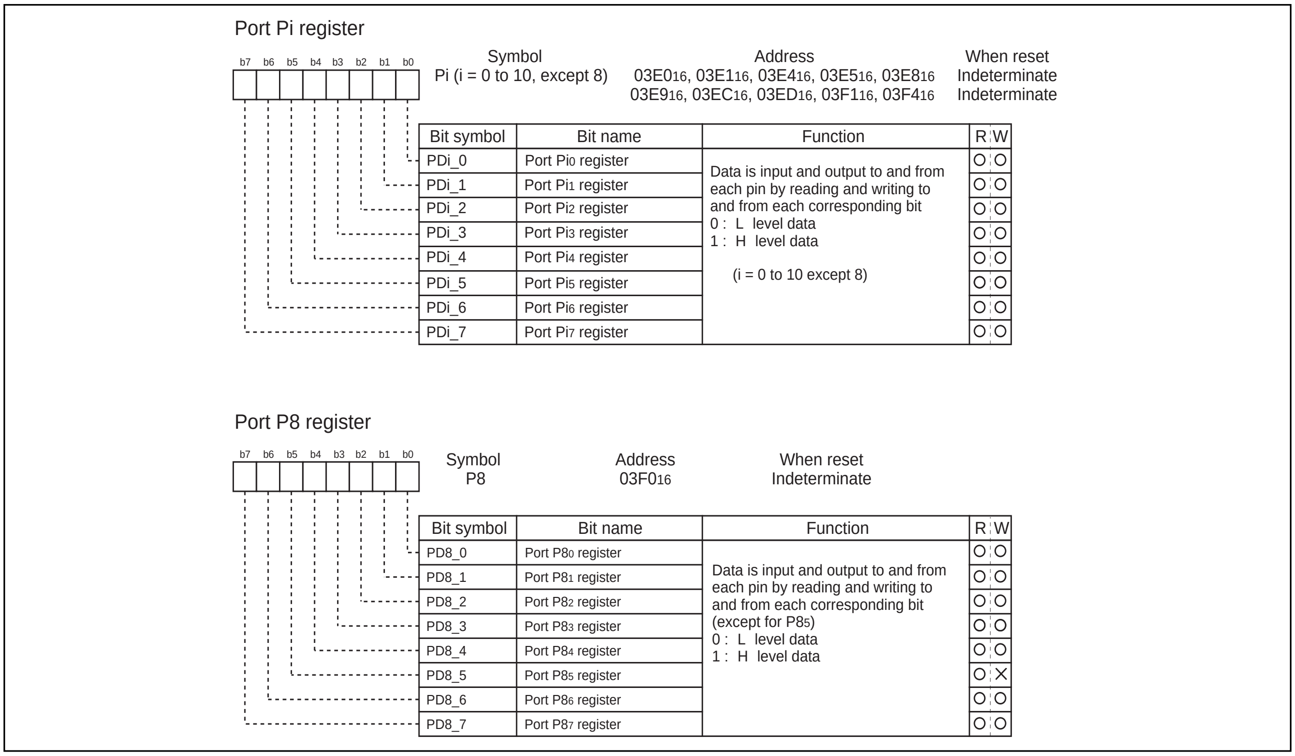
Note: Set bit 2 of protect register (address 000A$_{16}$) to 1 before rewriting to the port P9 direction register.

Port P8 direction register

Symbol	Address	When reset
PD8	03F2$_{16}$	00X00000$_2$

Bit symbol	Bit name	Function	R	W
PD8_0	Port P8$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_1	Port P8$_1$ direction register		○	○
PD8_2	Port P8$_2$ direction register		○	○
PD8_3	Port P8$_3$ direction register		○	○
PD8_4	Port P8$_4$ direction register		○	○
Nothing is assigned. This bit can either be set nor reset. When read, its content is indeterminate.			—	—
PD8_6	Port P8$_6$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_7	Port P8$_7$ direction register		○	○

Figure 110: Direction register



Port Pi register

Symbol	Address	When reset
Pi (i = 0 to 10, except 8)	03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$, 03E8$_{16}$ 03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$, 03F1$_{16}$, 03F4$_{16}$	Indeterminate Indeterminate

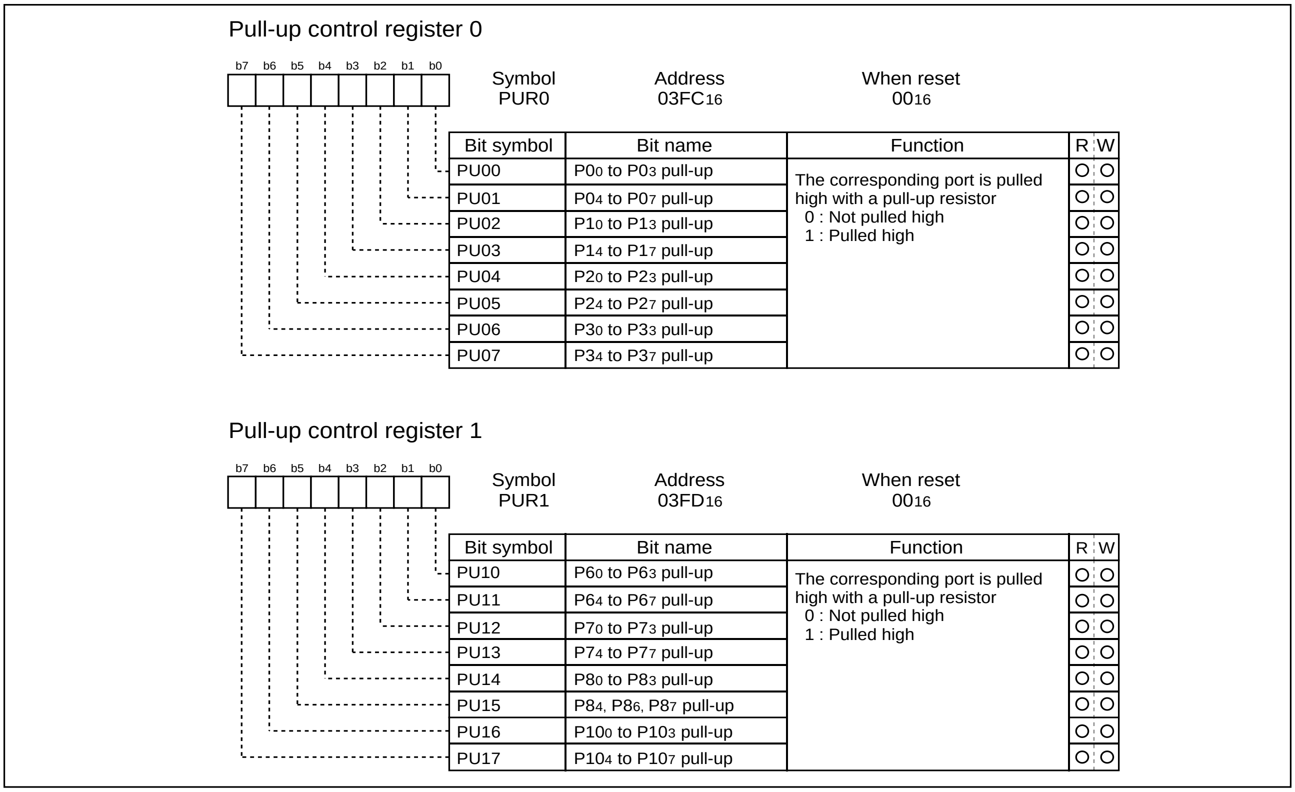
Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi_0 register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : L level data 1 : H level data (i = 0 to 10 except 8)	○	○
PDi_1	Port Pi_1 register		○	○
PDi_2	Port Pi_2 register		○	○
PDi_3	Port Pi_3 register		○	○
PDi_4	Port Pi_4 register		○	○
PDi_5	Port Pi_5 register		○	○
PDi_6	Port Pi_6 register		○	○
PDi_7	Port Pi_7 register		○	○

Port P8 register

Symbol	Address	When reset
P8	03F0$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
PD8_0	Port P8$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for P8$_5$) 0 : L level data 1 : H level data	○	○
PD8_1	Port P8$_1$ register		○	○
PD8_2	Port P8$_2$ register		○	○
PD8_3	Port P8$_3$ register		○	○
PD8_4	Port P8$_4$ register		○	○
PD8_5	Port P8$_5$ register		○	×
PD8_6	Port P8$_6$ register		○	○
PD8_7	Port P8$_7$ register		○	○

Figure 111: Port register



Pull-up control register 0

Symbol	Address	When reset
PUR0	$03FC_{16}$	00_{16}

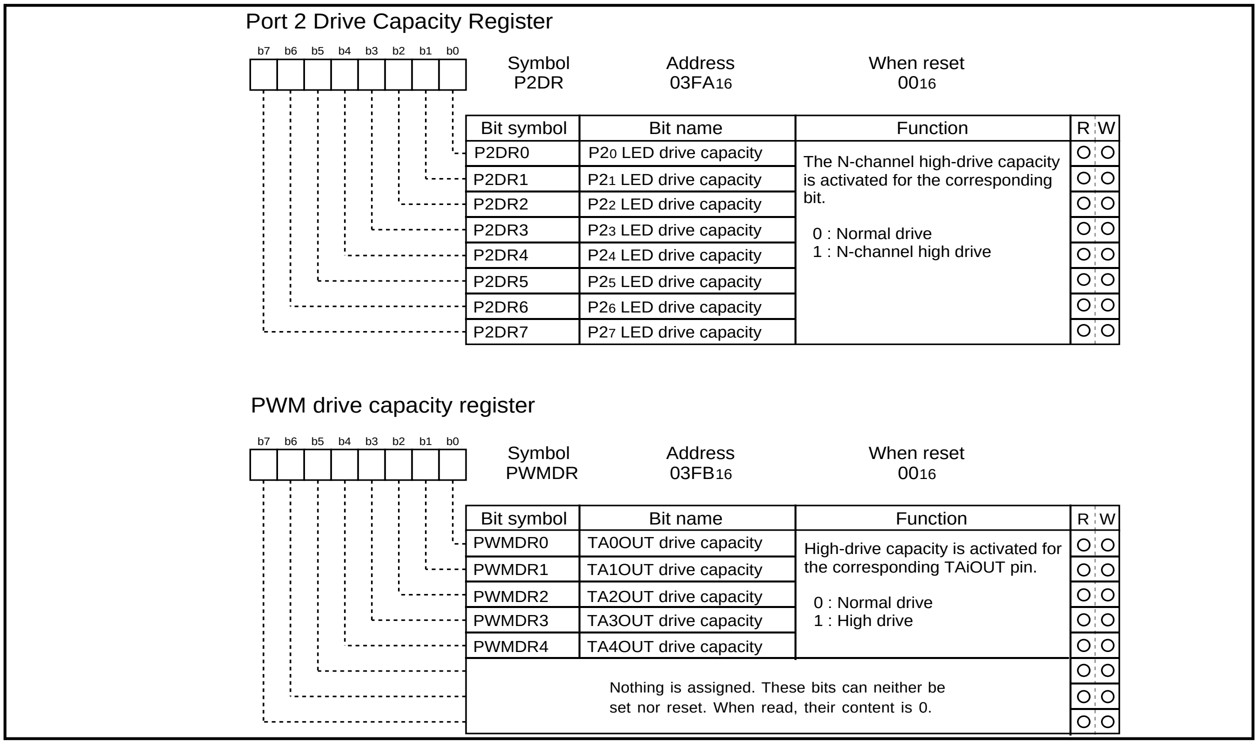
Bit symbol	Bit name	Function	R	W
PU00	$P0_0$ to $P0_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	$P0_4$ to $P0_7$ pull-up		○	○
PU02	$P1_0$ to $P1_3$ pull-up		○	○
PU03	$P1_4$ to $P1_7$ pull-up		○	○
PU04	$P2_0$ to $P2_3$ pull-up		○	○
PU05	$P2_4$ to $P2_7$ pull-up		○	○
PU06	$P3_0$ to $P3_3$ pull-up		○	○
PU07	$P3_4$ to $P3_7$ pull-up		○	○

Pull-up control register 1

Symbol	Address	When reset
PUR1	$03FD_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
PU10	$P6_0$ to $P6_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	$P6_4$ to $P6_7$ pull-up		○	○
PU12	$P7_0$ to $P7_3$ pull-up		○	○
PU13	$P7_4$ to $P7_7$ pull-up		○	○
PU14	$P8_0$ to $P8_3$ pull-up		○	○
PU15	$P8_4$, $P8_6$, $P8_7$ pull-up		○	○
PU16	$P10_0$ to $P10_3$ pull-up		○	○
PU17	$P10_4$ to $P10_7$ pull-up		○	○

Figure 112: Pull-up control register



Port 2 Drive Capacity Register

Symbol	Address	When reset
P2DR	$03FA_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
P2DR0	$P2_0$ LED drive capacity	The N-channel high-drive capacity is activated for the corresponding bit. 0 : Normal drive 1 : N-channel high drive	○	○
P2DR1	$P2_1$ LED drive capacity		○	○
P2DR2	$P2_2$ LED drive capacity		○	○
P2DR3	$P2_3$ LED drive capacity		○	○
P2DR4	$P2_4$ LED drive capacity		○	○
P2DR5	$P2_5$ LED drive capacity		○	○
P2DR6	$P2_6$ LED drive capacity		○	○
P2DR7	$P2_7$ LED drive capacity		○	○

PWM drive capacity register

Symbol	Address	When reset
PWMDR	$03FB_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
PWMDR0	TA0OUT drive capacity	High-drive capacity is activated for the corresponding TAiOUT pin. 0 : Normal drive 1 : High drive	○	○
PWMDR1	TA1OUT drive capacity		○	○
PWMDR2	TA2OUT drive capacity		○	○
PWMDR3	TA3OUT drive capacity		○	○
PWMDR4	TA4OUT drive capacity		○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their content is 0.			○	○
			○	○
			○	○

Figure 113: Port 2 and PWM drive capacity registers

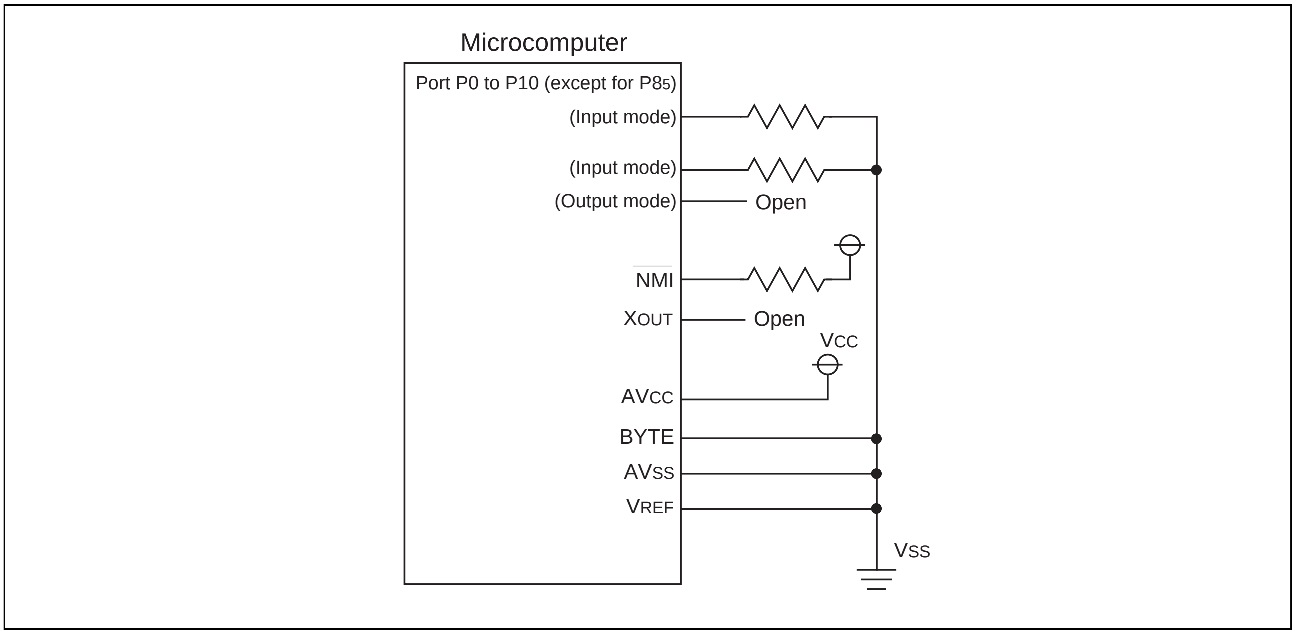


Figure 114: Example connection unused pins

Table 36: Example connection of unused pins in single-chip mode

Pin name	Connection
Ports P0 to P3, P6 to P8, P10 (excluding P85)	After setting for input mode, connect every pin to Vss or Vcc via a resistor; or after setting for output mode, leave these pins open
Xout	Open
$\overline{\text{NMI}}$	Connect via resistor to Vcc (pull-up)
AVcc	Connect to Vcc
Avss, Vref, BYTE	Connect to Vss

3.0 Usage

3.1 Usage Precautions

Timer A (timer mode)

- Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets “FFFF16”. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

Timer A (event counter mode)

- Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets “FFFF16” by underflow or “000016” by overflow. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.
- When stop counting in free run type, set timer again.

Timer A (pulse width modulation mode)

- The timer Ai interrupt request bit becomes “1” if setting operation mode of the timer in compliance with any of the following procedures:
  - Selecting PWM mode after reset.
  - Changing operation mode from timer mode to PWM mode.
  - Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to “0” after the above listed changes have been made.

- Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the TAiOUT pin is outputting an “H” level in this instance, the output level goes to “L”, and the timer Ai interrupt request bit goes to “1”. If the TAiOUT pin is outputting an “L” level in this instance, the level does not change, and the timer Ai interrupt request bit does not becomes “1”.

Timer B (timer mode)

- Reading the timer Bi register while a count is in progress allows reading , with arbitrary timing, the value of the counter. Reading the timer Bi register with the reload timing gets “FFFF16”. Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a proper value.

A-D Converter

- Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).
  In particular, when the Vref connection bit is changed from “0” to “1”, start A-D conversion after an elapse of 1 s or longer.
- When changing A-D operation mode, select analog input pin again.
- Using one-shot mode or single sweep mode
- Read the correspondence A-D register after confirming A-D conversion is finished. (It is known by A-D conversion interrupt request bit.)
- Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1
- Use the undivided main clock as the internal CPU clock.

Stop Mode and Wait Mode

- When returning from stop mode by hardware reset, RESET pin must be set to “L” level until main clock oscillation is stabilized.

Interrupts

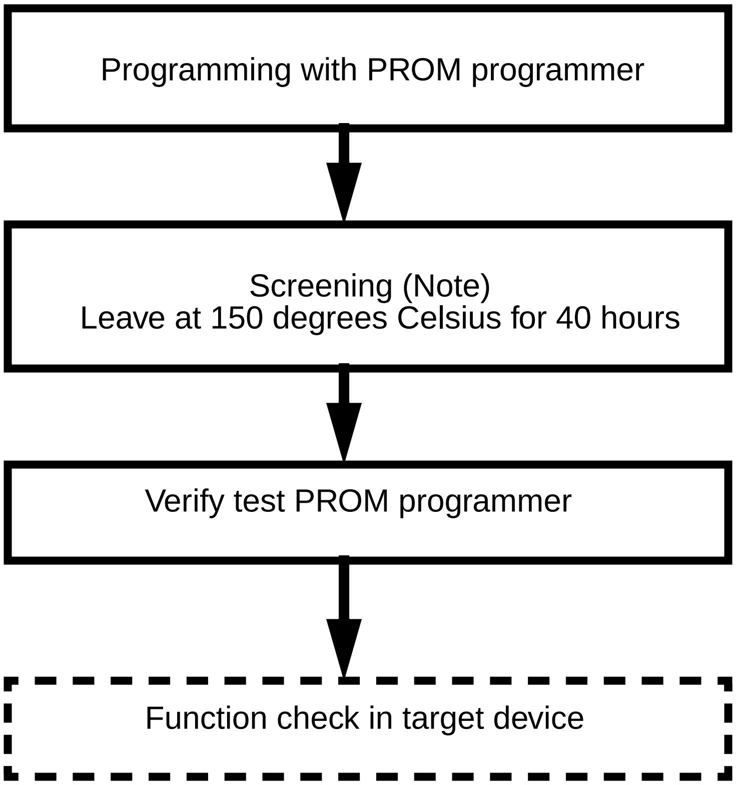
- Reading address 00000_{16}
  - When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.
    The interrupt request bit of the certain interrupt written in address 00000_{16} is then set to “0”.
    Reading address 00000_{16} by software sets enabled highest priority interrupt source request bit to “0”.
    Though the interrupt is generated, the interrupt routine may not be executed.
    Do not read address 00000_{16} by software.
- Setting the stack pointer
  - The value of the stack pointer immediately after reset is initialized to 000016. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.
  - When using the NMI interrupt, initialize the stack point at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupts including the NMI interrupt is prohibited.
- The NMI interrupt
  - As for the NMI interrupt pin, an interrupt cannot be prohibited. Connect it to the VCC pin if unused.
  - Do not get either into stop mode or into wait mode with the NMI pin set to “L”.

Built-in PROM version

- All built-in PROM versions

  High voltage is required to program to the built-in PROM. Be careful not to apply excessive voltage. Be especially careful during power-on.
- One Time PROM version

  One Time PROM versions shipped in blank, of which built-in PROMs are programmed by users, are also provided. For these microcomputers, a programming test and screening are not performed in the assembly process and the following processes. To improve their reliability after programming, we recommend to program and test as flow shown in Figure 115 before use.


Programming with PROM programmer
Screening (Note)
Leave at 150 degrees Celsius for 40 hours
Verify test PROM programmer
Function check in target device

Note: Never expose to 150 degrees Celsius exceeding 100 hours.

Figure 115: Programming and test flow for One-time PROM (OTP) version

- EPROM version
    - Cover the transparent glass window with a shield or others during the read mode because exposing to sun light or fluorescent lamp can cause erasing the information.
    - A shield to cover the transparent window is available from Mitsubishi Electric Corp. Be careful that the shield does not touch the EPROM lead pins.
    - Clean the transparent glass before erasing. Fingers' flat and paste disturb the passage of ultraviolet rays and may affect badly the erasure capability.
    - The EPROM version is a tool only for program development (for evaluation), and do not use it for the mass product run

4.0 Specifications

4.1 Electrical

Table 37: Absolute maximum ratings

Symbol	Parameter		Condition	Rated Value		Unit
V_{CC}	Supply voltage		VCC=AVCC	-0.3 to 7.0		V
AV_{CC}	Analog supply voltage		VCC=AVCC	-0.3 to 7.0		V
V_I	Input voltage	Port0, Port1, Port2, Port3, Port6, Port7, Port8, Port10, RESET, VREF, XIN		-0.3 to Vcc+0.3		V
V_I	Input voltage	CNVSS		-0.3 to 7.0	(Note 1)	V
V_O	Output voltage	Port0, Port1, Port2, Port3, Port6, Port7, Port8 (except P85), Port10, RESET, VREF, XIN				V
P_d	Power dissipation		Ta=25° C	300		mW
T_{opr}	Operating ambient temperature			-20 to 85 / -40 to 85	(Note 2)	°C
T_{stg}	Storage temperature			-65 to 150		°C

Note 1: When writing to EPROM, CNVss rated value is -0.3 to 13 volts
Note 2: Extended temperature version (-40 to 85 °C) must be specified.

Table 38: Recommended operatingconditions (Note 1)

Symbol	Parameter		Standard			Unit
			Min	Typ	Max	
V_{CC}	Supply voltage		4.1	5.0	5.5	V
AV_{CC}	Analog supply voltage			Vcc		V
V_{IH}	High input voltage	Port 0, Port1, Port2, Port3, Port6, Port7, Port8, Port10, RESET, VREF, XIN, CNVSS	0.8Vcc		Vcc	V
V_{IL}	Low input voltage	Port0, Port1, Port2, Port3, Port6, Port7, Port8, Port10, RESET, VREF, XIN, CNVSS	0		0.2Vcc	V
Ioh (peak)	High peak output current	Port0, Port1, Port3, Port6, P71, P73, P75, P77, P81 to P87, Port10			-10	mA
		P20 to P27, P70, P72, P74, P76, P80			-20	mA
Ioh (avg.)	High avg output current	Port0, Port1, Port3, Port6, P71, P73, P75, P77, P81 to P87, Port10			-5	mA
		P20 to P27, P70, P72, P74, P76, P80			-10	mA
ΣIoh(peak)	High peak output current	P2, P3, P6, P7, $P8_0$~$P8_2$			-80	mA
		P0, P1, $P8_3$~$P8_7$, P10			-80	mA
ΣIoh (avg.)	High avg output current	P2, P3, P6, P7, $P8_0$~$P8_2$			-40	mA
		P0, P1, $P8_3$~$P8_7$, P10			-40	mA
Iol (peak)	Low peak output current	Port0, Port1, Port3, Port6, P71, P73, P75, P77, P81 to P87, Port10			10	mA
		P20 to P27, P70, P72, P74, P76, P80			20	mA
Iol (avg.)	Low avg output current	Port0, Port1, Port3, Port6, P71, P73, P75, P77, P81 to P87, Port10			5	mA
		P20 to P27, P70, P72, P74, P76, P80			10	mA
ΣIol (peak)	Low peak output current	P2, P3, P6, P7, $P8_0$~$P8_2$			80	mA
		P0, P1, $P8_3$~$P8_7$, P10			80	mA
ΣIol (avg.	Low avg output current	P2, P3, P6, P7, $P8_0$~$P8_2$			40	mA
		P0, P1, $P8_3$~$P8_7$, P10			40	mA
f(Xin)	Main clock input oscillation frequency		0		12	MHz

Note: The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.

Table 39: Electrical characteristics (Vcc=5V, Vss=0V, Ta=25°C, f(xin) = 12MHz)

Symbol	Parameter			Measuring condition		Standard			Unit
						Min	Typ	Max	
V_{OH}	High output voltage	Port0, Port1, Port2, Port3, Port6, Port71, ,P73,P75,P77,Port8 (except P85), Port10		I_{OH} = -5mA		3.0			V
V_{OH}	High output voltage	Port 70,P72,P74,P76,P80		I_{OH} = -10mA		3.0			V
V_{OH}	High output voltage	Port0, Port1, Port2, Port3, Port6, Port71, ,P73,P75,P77,Port8 (except P85), Port10		I_{OH} = -5mA		3.0			V
V_{OH}	High output voltage	Xout	high power	I_{OH} = -1mA		3.0			V
			low power	I_{OH} = -0.5mA		3.0			V
V_{OL}	Low output voltage	Port0, Port1, Port2, Port3, Port6, Port71, ,P73,P75,P77,Port8 (except P85), Port10		I_{OL} = 5mA				2.0	V
V_{OL}	Low output voltage	High-drive mode Port 2		I_{OL} = 10mA				2.0	V
V_{OL}	Low output voltage	Port 70,P72,P74,P76,P80		I_{OL}= 10mA				2.0	V
V_{OL}	Low output voltage	Port0, Port1, Port2, Port3, Port6, Port71, ,P73,P75,P77,Port8 (except P85), Port10		I_{OL} = 200uA				0.45	V
V_{OL}	Low output voltage	Xout	high power	I_{OH} = 1mA				2.0	V
			low power	I_{OH} = 0.5mA				2.0	V
V_{T+}-V_{T-}	Hysteresis	HOLD, RDY, TA0in to TA4in, INT0 to INT2, ADTRG, CTS0, CTS1, CLK0, CLK1, TA2out to TA4out, NMI, KI0 to KI15				0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis	RESET				0.2		0.8	V
Iih	High input current	Port0, Port1, Port2, Port3, Port6, Port7,Port8, Port10, XIN, RESET, CNVss, BYTE		V_I = 5V				5.0	uA
Iil	Low input current	Port0, Port1, Port2, Port3, Port6, Port7, Port8, Port10, XIN, RESET, CNVss, BYTE		V_I = 0V				-5.0	uA
R_{PULLUP}	Pull-up resistance	Port0, Port1, Port2, Port3, Port6, Port7, Port8, Port10, XIN, RESET, CNVss, BYTE		V_I = 0V		30	50	167	kΩ
R_{XIN}	Feedback resistance, XIN						1.0		MΩ
V_{RAM}	RAM retention voltage			When clock is stopped		2.0			V
Icc	Power supply current(Vcc = 5.5V) 1.Estimated			Output pins open, other pins tied to Vss	f(XIN)=12MHz square wave			83[1]	mA
					Ta=25°C clock stopped			1	uA
					Ta=85°C clock stopped			20	uA

Table 40: A-D conversion characteristics (Vcc,Avcc=5V, Vs,AVss=0V, Ta=25°C, f(xin) = 12MHz)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min	Typ	Max	
-	Resolution		VREF = Vcc			10	Bits
-	Absolute accuracy	Sample and hold function not available	VREF = Vcc = 5V			±3	LSB
		Sample and hold function available (10bit)	VREF = Vcc = 5V			±3	LSB
		Sample and hold function available (8bit)	VREF = Vcc = 5V			±2	LSB
RLADDER	Ladder resistance		VREF = Vcc	10		40	kΩ
tCONV	Conversion time (10bit)			2.75			µs
tCONV	Conversion time (8bit)			2.34			µs
tSAMP	Sampling time			0.25			µs
VREF	Reference voltage			2			V
VIA	Analog input voltage			0		VREF	V

4.2 Timing

Timing requirements referenced to Vcc = 5V, Vss = 0V, Ta = 25°C unless otherwise specified.

Table 41: External clock input

Symbol	Parameter	Standard		Unit
		Min	Max	
tc	External clock input cycle time	83.3		ns
tw(H)	External clock input HIGH pulse width	33		ns
tw(L)	External clock input LOW pulse width	33		ns
tr	External clock rise time		15	ns
tf	External clock fall time		15	ns

Table 42: Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(TA)	TAiIN input cycle time	83		ns
tw(TAH)	TAiIN input HIGH pulse width	33		ns
tw(TAL)	TAiIN input LOW pulse width	33		ns

Table 43: Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(TA)	TAiIN input cycle time	333		ns
tw(TAH)	TAiIN input HIGH pulse width	167		ns
tw(TAL)	TAiIN input LOW pulse width	167		ns

Table 44: Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(TA)	TAiIN input cycle time	167		ns
tw(TAH)	TAiIN input HIGH pulse width	83		ns
tw(TAL)	TAiIN input LOW pulse width	83		ns

Table 45: Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tw(TAH)	TAiIN input HIGH pulse width	83		ns
tw(TAL)	TAiIN input LOW pulse width	83		ns

Table 46: Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(UP)	TAiOUT input cycle time	1667		ns
tw(UPH)	TAiOUT input HIGH pulse width	833		ns
tw(UPL)	TAiOUT input LOW pulse width	833		ns
tsu(UP-TIN)	TAiOUT input setup time	333		ns
th(TIN-UP)	TAiOUT input hold time	333		ns

Table 47: A-D trigger input

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(AD)	$\overline{AD_{TRG}}$ input cycle time (triggerable minimum)	833		ns
tw(ADL)	$\overline{AD_{TRG}}$ input LOW pulse width	105		ns

Table 48: Serial I/O

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(CK)	CLKi input cycle time	167		ns
tw(CKH)	CLKi input HIGH pulse width	83		ns
tw(CKL)	CLKi input LOW pulse width	83		ns
td(C-Q)	TxDi output delay time		80	ns
th(C-Q)	TxDi hold time	0		ns
tsu(D-C)	RxDi input setup time	30		ns
th(C-D)	RxDi input hold time	90		ns

Table 49: External interrupt INTi inputs

Symbol	Parameter	Standard		Unit
		Min	Max	
tw(INH)	$\overline{\text{INTi}}$ input HIGH pulse width	208		ns
tw(INL)	$\overline{\text{INTi}}$ input LOW pulse width	208		ns

4.3 Timing Diagrams- Peripheral/interrupt

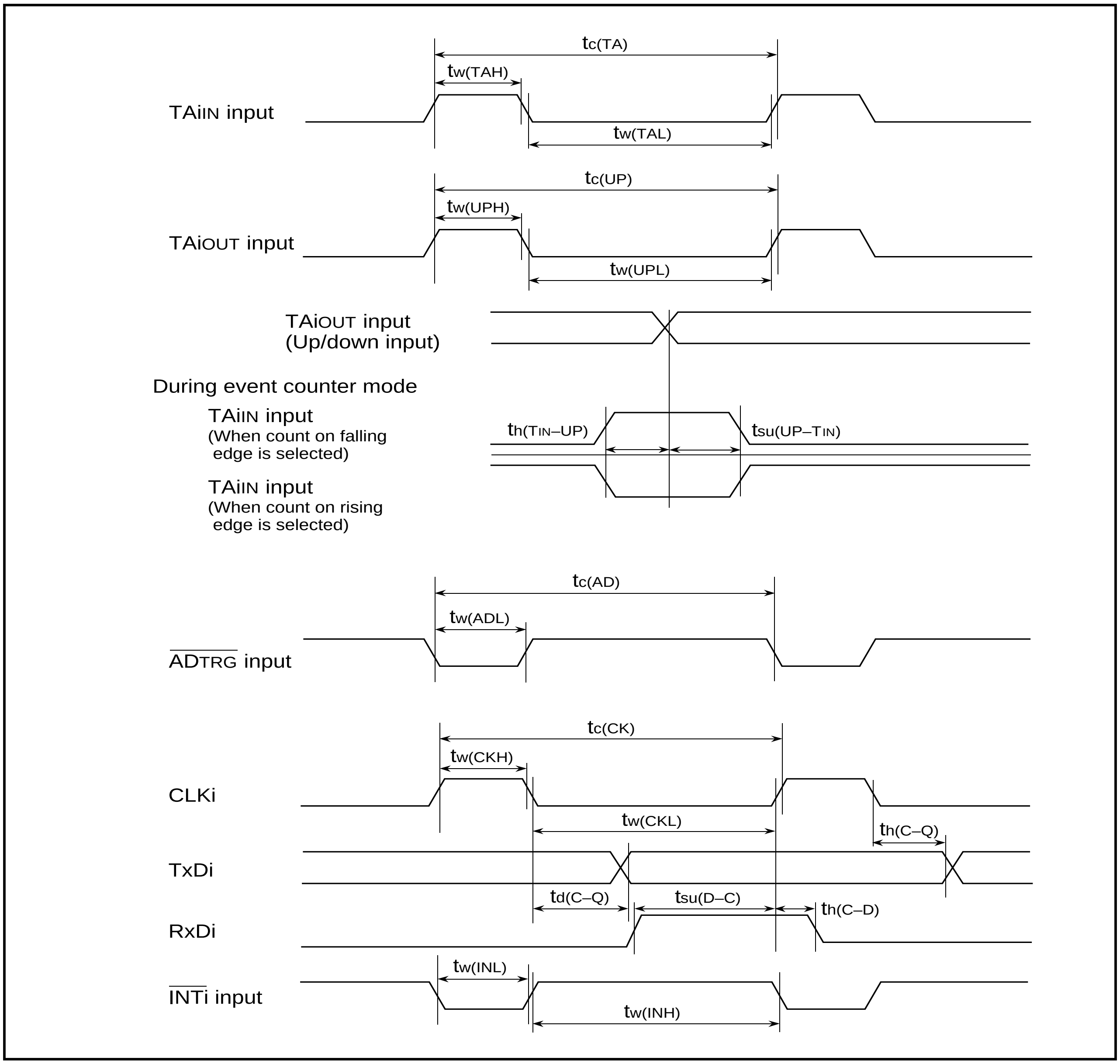


Figure 116: Peripheral / Interrupt timing diagram