1.1 Description

The M30240 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 microcontrollers 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.

1.1.1 Features

- CPU ..... 16-bit (including a hardware multiplier)
- Number of instructions 91
- Shortest instruction execution time ..... 83ns (f(X_{IN})=12MHz)
- USB Features: ..... 9 endpoints
  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 Synthesizer ..... PLL for 48MHz clock
- Memory capacity (mask device):..... ROM (40K, 48K) / RAM (3.0 K)
- Memory capacity (OTP device):..... PROM (128K) / RAM (5K)
- Supply Voltage ..... 4.1 to 5.25V (f(X_{IN})=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 (peak) 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: 19 sources)
- A-D Converter ..... 10 bits X 8 channels
- CRC calculation circuit ..... 1 circuit (industry standard polynomial)
- Watchdog timer ..... 1 line (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-A (0.8 mm pitch)

1.1.2 Applications

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

1.1.3 Table of Contents

1.1 Features . . . 2
Applications . . . 2
Pin Configuration . . . 3
Block Diagram . . . 4
Performance outline . . . 5
Pin Description . . . 7
Overview . . . 9
1.2 Operation of Functional Blocks . . . 10
Central Processing Unit . . . 10
Processor Mode . . . 13
Memory . . . 14
SFR Map . . . 15
Reset . . . 21
Clock-Generating Circuit . . . 22
Stop Mode . . . 25
Wait Mode . . . 25
Status Transition Of Internal Clock Φ . . . 25
Power Control . . . 26
Protection . . . 27
Interrupts . . . 28
NMI Interrupt . . . 34
Key input interrupt . . . 35
Address Match Interrupt . . . 36
Watchdog Timer . . . 36
Frequency Synthesizer Circuit . . . 38
Universal Serial Bus . . . 41
DMAC . . . 60
Timers . . . 65
Timer A . . . 66
Timer B . . . 76
UART0 to UART2 . . . 78
A-D Converter . . . 99
CRC Calculation Circuit . . . 109
Programmable I/O Ports . . . 110
1.3 Usage Precautions . . . 117
Precautions . . . 117
1.4 Specifications . . . 124
Electrical characteristics . . . 124
Timing . . . 128
Timing Diagram . . . 130
1.5 Applications . . . 131
Frequency Synthesizer . . . 131
Attach/Detach Function . . . 135
Low Pass Filter Network . . . 135
USB Transceiver . . . 136
Programming Notes . . . 137

1.1.4 Pin Configuration

Figure 1.1 shows the pin configuration (top view).

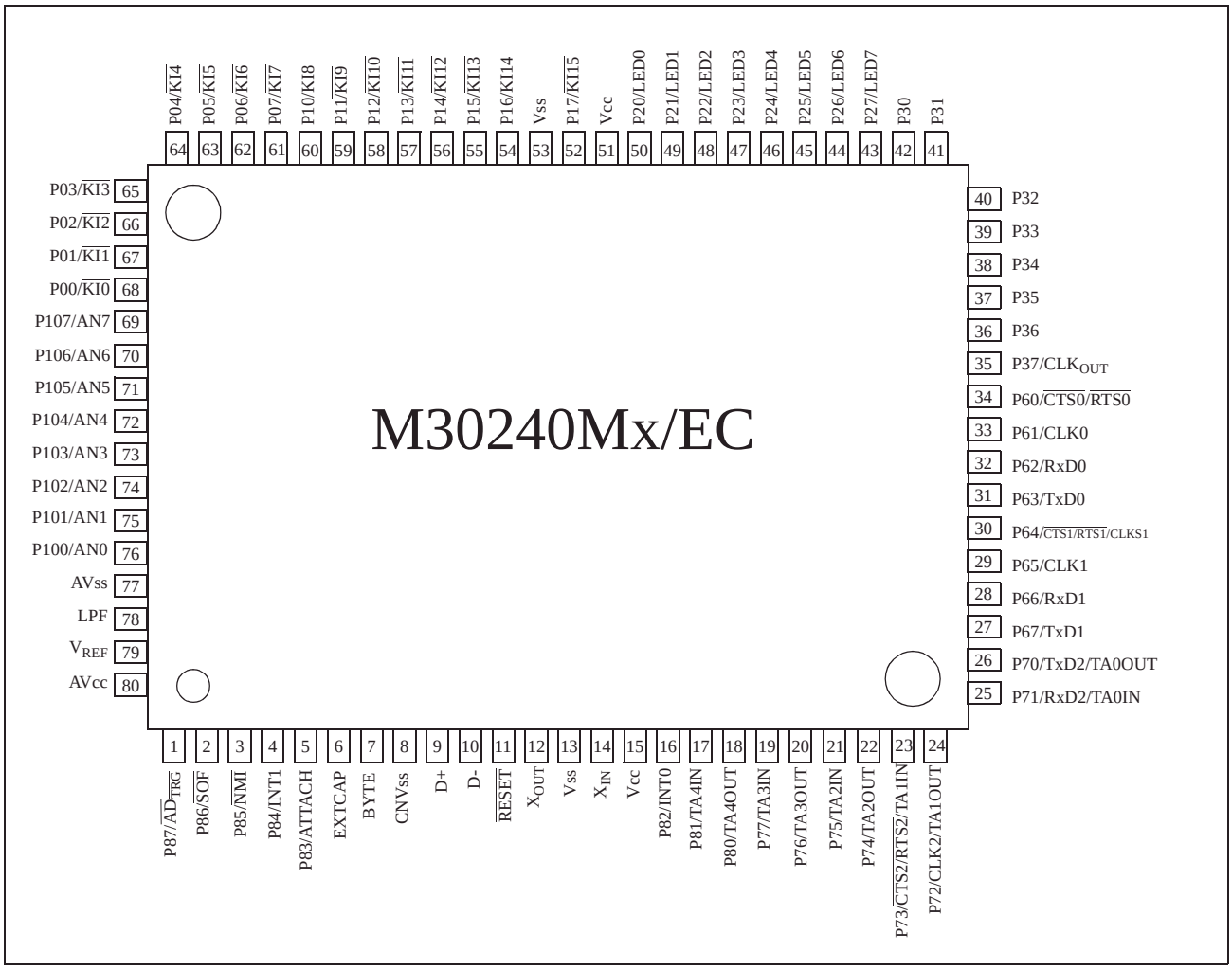


Figure 1.1: Pin Configuration (top view)

1.1.5 Block Diagram

Figure 1.2 is a block diagram of the M30240 group.

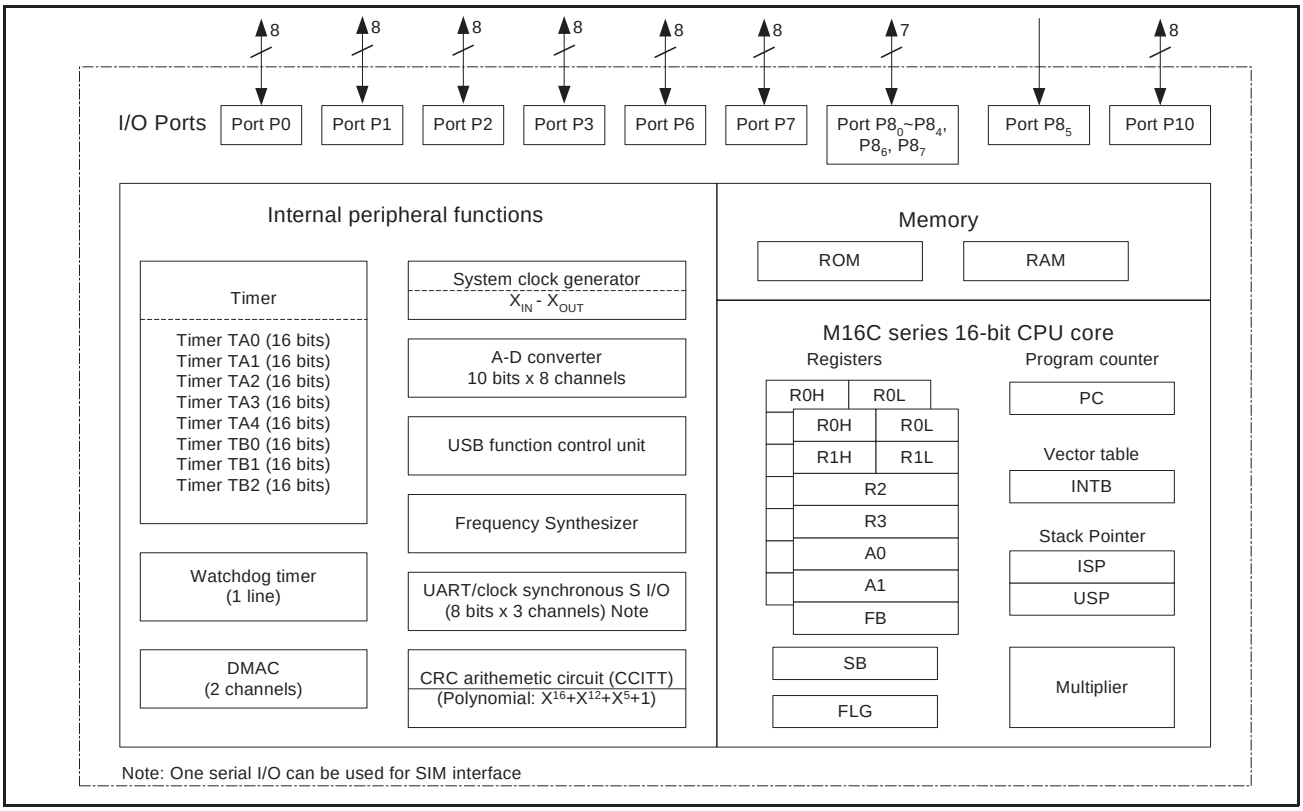


Figure 1.2: Block diagram of M30240 group

1.1.6 Performance outline

Table 1.1 is a performance outline of the M30240 group.

Table 1.1: Performance outline of M30240 group

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		83ns (f(XIN) =12MHz)
Memory capacity	ROM	(See Table 1.2: 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:19 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 (typical)		4.1 to 5.25V, (f(XIN)=12MHz, without software wait)
Power consumption (typical)		250 mW, Vcc=5.0V, 12MHz
I/O characteristics	I/O withstand voltage	5V
	Average output current	5 mA available on ports P0, P1, P3,P6, $P7_1$, $P7_3$, $P7_5$, $P7_7$, $P8_1$~$P8_4$, $P8_6$, $P8_7$, P10 10 mA available on ports P2, $P7_0$, $P7_2$, $P7_4$, $P7_6$, $P8_0$
Operating temperature		0 to 70°C
Device configuration		CMOS high performance silicon gate
Package		80-pin plastic molded QFP

Renesas plans to release the following products in the M30240 group:

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

(2) ROM capacity

(3) Package

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

Figure 1.3 shows the type number, memory size and package for the M30240 group.

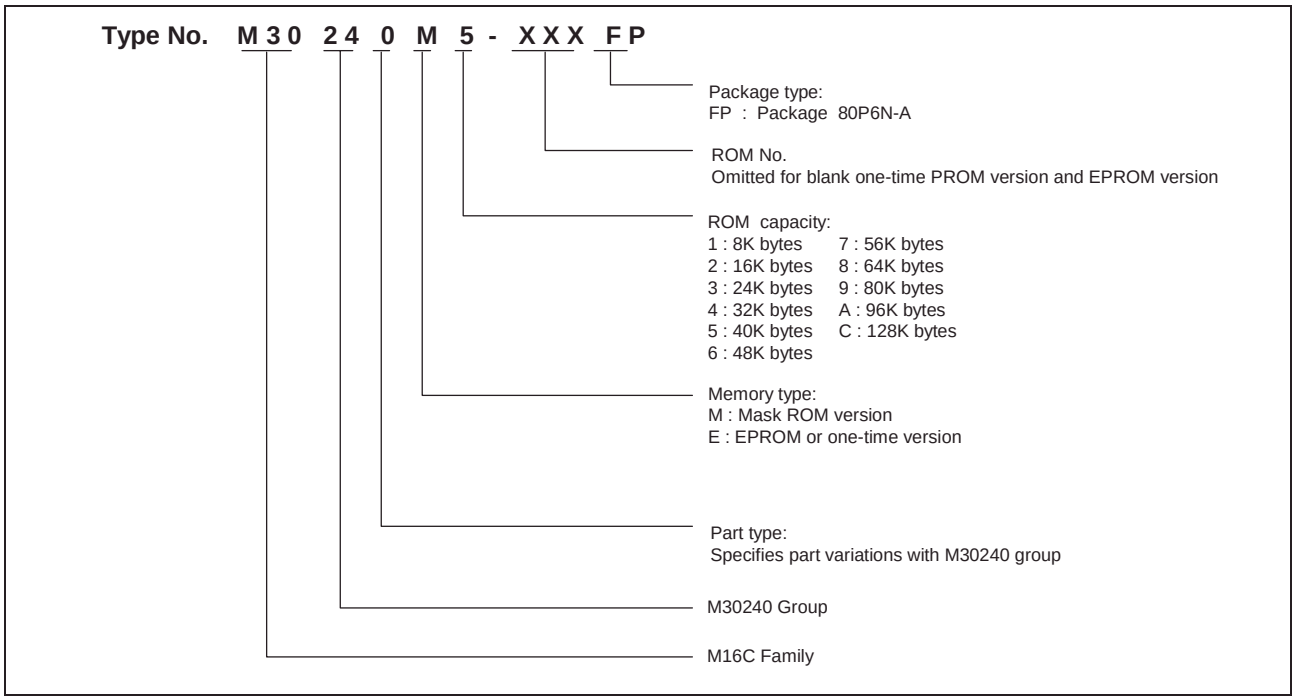


Figure 1.3: Type number, memory size, and package

Table 1.2 shows the Package Number, type, ROM and RAM Capacity for M30240 Group.

Table 1.2: M30240 Group

Type	ROM Capacity	RAM Capacity	Package Type	Remarks
M30240M5-XXXXFP	40K bytes	3K bytes	80P6N-A	Mask ROM Version
M30240M6-XXXXFP	48K bytes	3K bytes	80P6N-A	Mask ROM Version
M30240ECFP	128K bytes	5K bytes	80P6N-A	One-time PROM version

1.1.7 Pin Description

Table 1.3 shows the M30240 pin description.

Table 1.3: 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. $P8_4$ also functions as external interrupt 1 and $P8_3$ is used to enable the stealth detach function for the USB transceiver.
6	EXTCAP	–	An external capacitor (Ext. Cap) pin. When the USB transceiver voltage converter is used, a 2.2 µF and a 0.1 µF capacitor should connect 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 ($000C_{16}$) 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 27 to 33 Ω is connected to this pin.
10	USB D^-	I/O	USB D- voltage line interface, a series resistor of 27 to 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}	-	Ground: V_{SS} = 0V
14	Xin	I	Input and output signals to and from the internal clock generation circuit. Connect a ceramic resonator or quartz crystal between Xin and Xout pins to set the oscillation frequency. If an external clock is used, connect the clock source to the Xin pin and leave the Xout pin open.
15	V_{CC}	-	Power: V_{CC} = 4.1~ 5.25V
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 $\overline{CTS}$, $\overline{RTS}$, 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 $\overline{CTS}$, $\overline{RTS}$, 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 $\overline{CTS}$, $\overline{RTS}$, CLK, RXD, and TXD as selected by software.
35-42	$P3_7$ ~ $P3_0$	I/O	CMOS I/O port.

Pin #	Name	I/O	Description
43-50	$P2_7$/LED7 ~ $P2_0$/LED0	I/O	CMOS I/O port. These pins are capable of driving up to 20mA (peak) for LEDs.
51	V_{cc}	I	Power: V_{cc} = 4.1~ 5.25V
52	$P1_7/\overline{KI}_{15}$	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	$P1_6/\overline{KI}_{14}$~ $P1_0/\overline{KI}_8$	I/O	CMOS I/O port. This port can also function as the key-on wakeup interrupts ($\overline{KI8}$ ~ $\overline{KI14}$).
61-68	$P0_7/\overline{KI}_7$~ $P0_0/\overline{KI}_0$	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	This pin is a power supply input for the AD converter. (Connect to Vss)
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	This pin is a power supply input for the AD converter. (Connect to Vcc)

1.1.8 Overview

The M30240 group 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 1.4.

The USB function control unit of the M30240 group 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.

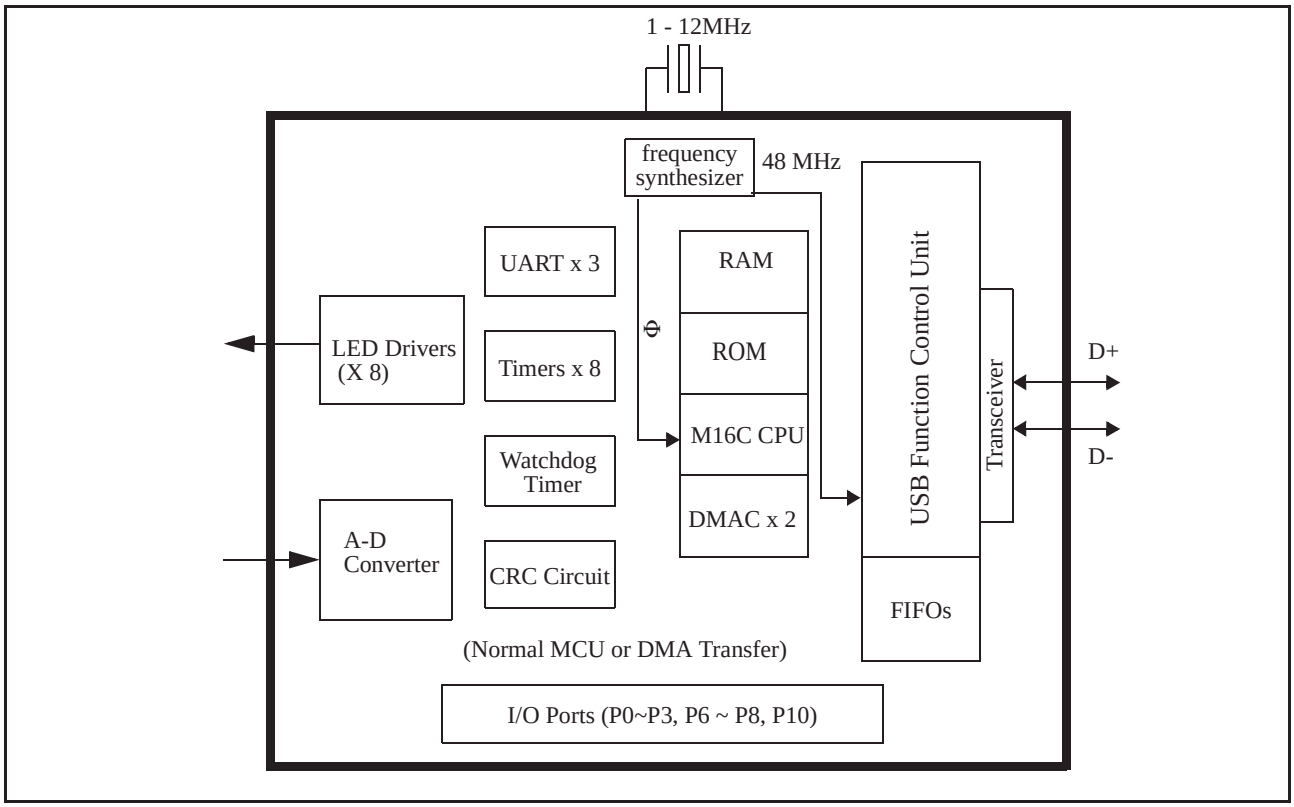


Figure 1.4: M30240 block diagram

1.2 Operation of Functional Blocks

The M30240 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.

1.2.1 Central Processing Unit

The CPU has a total of 13 registers shown in Figure 1.5. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

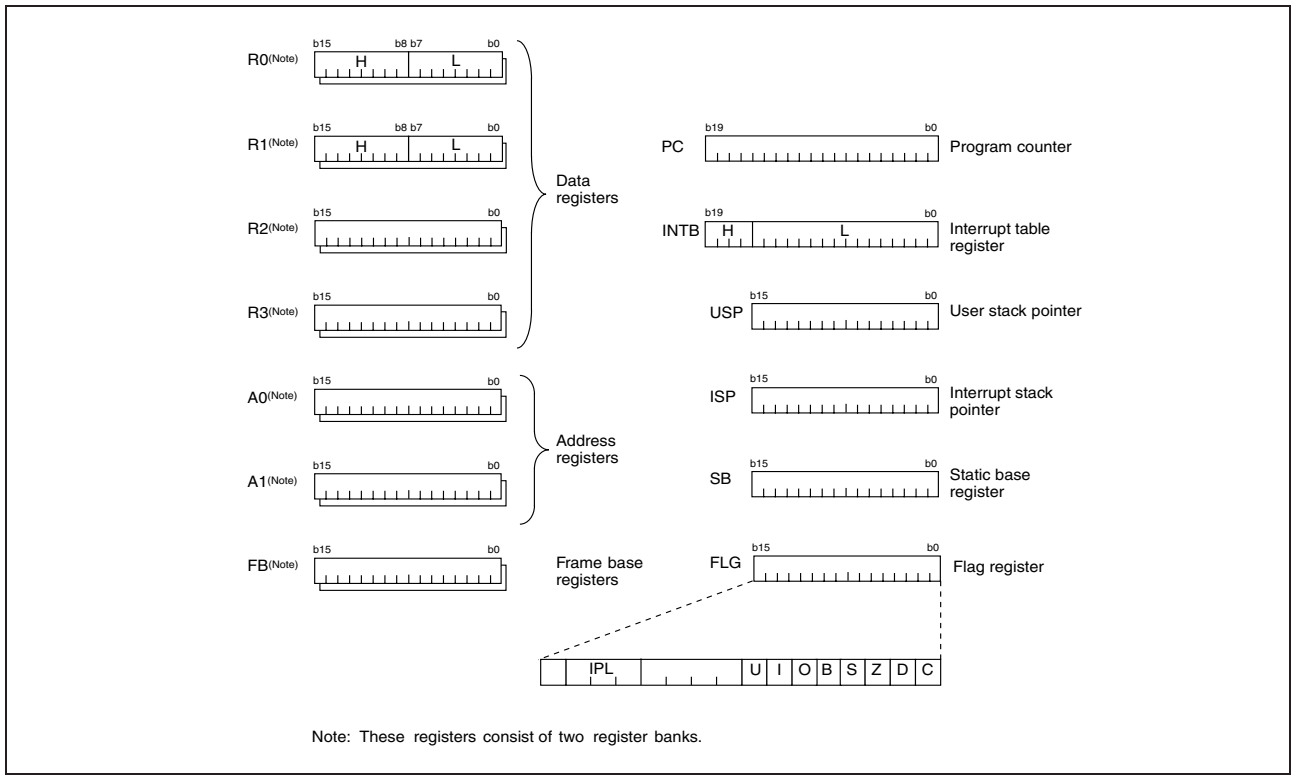


Figure 1.5: Central processing unit register

1.2.1.1 Data registers

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).

1.2.1.2 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).

1.2.1.3 Frame base register (FB)

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

1.2.1.4 Program counter (PC)

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

1.2.1.5 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.

1.2.1.6 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).

1.2.1.7 Static base register (SB)

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

1.2.1.8 Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 1.6 shows the flag register (FLG). The following explains the function of each flag:

1.2.1.8.1 Bit 0: Carry flag (C flag)

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

1.2.1.8.2 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.

1.2.1.8.3 Bit 2: Zero flag (Z flag)

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

1.2.1.8.4 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”.

1.2.1.8.5 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”.

1.2.1.8.6 Bit 5: Overflow flag (O flag)

This flag is set to “1” when an arithmetic operation resulted in overflow; otherwise, cleared to “0”.

1.2.1.8.7 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.

1.2.1.8.8 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.

1.2.1.8.9 Bits 8 to 11: Reserved area

1.2.1.8.10 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.

1.2.1.8.11 Bit 15: Reserved area

The C, Z, S, and O flags are changed when instructions are executed. See the M16C software manual for details.

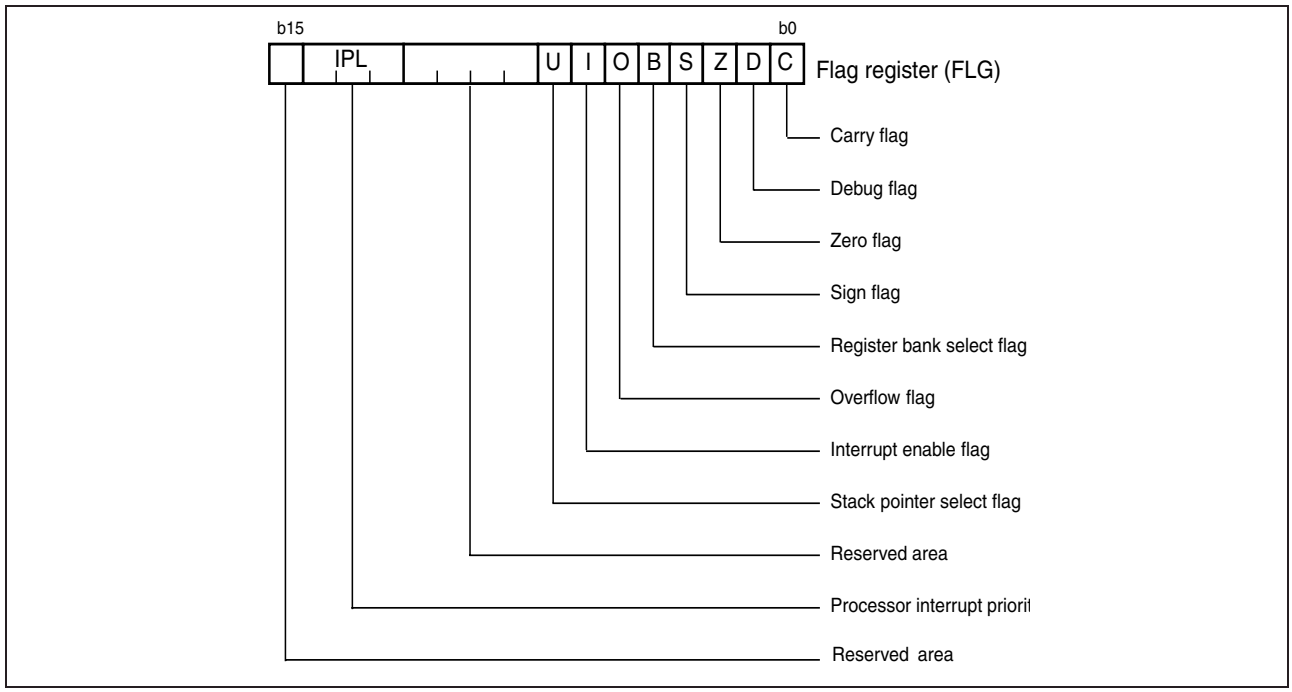


Figure 1.6: Flag register (FLG)

1.2.2 Processor Mode

Figure 1.7 shows the processor mode registers 0 and 1.

Processor mode register 0 (Note)

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

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)

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

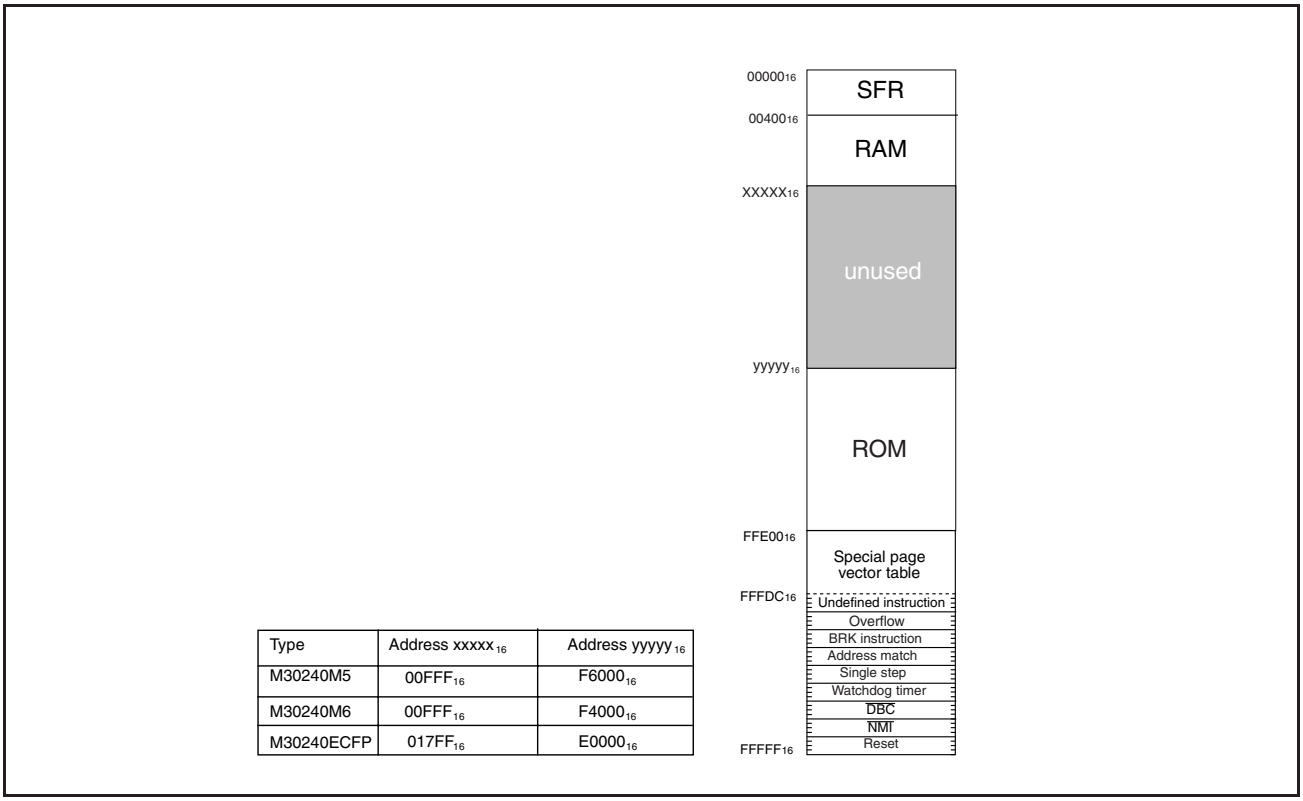
Symbol	Address	When reset
PM1	0005_{16}	$00XXXXX0_2$

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 1.7: Processor mode registers 0 and 1

1.2.3 Memory



Type	Address xxxxx$_{16}$	Address yyyyy$_{16}$
M30240M5	00FFF$_{16}$	F6000$_{16}$
M30240M6	00FFF$_{16}$	F4000$_{16}$
M30240ECFP	017FF$_{16}$	E0000$_{16}$

Figure 1.8: Memory Map

Figure 1.8 is a memory map of the M30240 group. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$. Addresses above $yyyyy_{16}$ are ROM. For example, in the M30240ECFP, 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 $xxxxx_{16}$ are RAM. For example, in M30240ECFP, 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.

1.2.4 SFR Map

The SFR tables show the peripheral control registers, their addresses, acronyms, and values after reset.

Address	Register name	Acronym	Value after reset	Figure number
0000_{16}				
0001_{16}				
0002_{16}				
0003_{16}				
0004_{16}	Processor mode register 0	PM0	00_{16}	Figure 1.7
0005_{16}	Processor mode register 1	PM1	0 0 0	Figure 1.7
0006_{16}	System clock control register 0	CM0	48_{16}	Figure 1.13
0007_{16}	System clock control register 1	CM1	20_{16}	Figure 1.13
0008_{16}				
0009_{16}	Address match interrupt enable register	AIER	0 0	Figure 1.16
$000A_{16}$	Protect register	PRCR	0 0 0	Figure 1.14
$000B_{16}$				
$000C_{16}$	USB control register	USBC	00_{16}	Figure 1.31
$000D_{16}$				
$000E_{16}$	Watchdog timer start register	WDTS		Figure 1.22
$000F_{16}$	Watchdog timer control register	WDC	0 0 0 ? ? ? ? ?	Figure 1.22
0010_{16}	Address match interrupt register 0	RMAD0	00_{16}	Figure 1.16
0011_{16}			00_{16}	
0012_{16}			0 0 0 0	
0013_{16}				
0014_{16}	Address match interrupt register 1	RMAD1	00_{16}	Figure 1.16
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	USBAD	00_{16}	Figure 1.30
0020_{16}	DMA0 source pointer	SAR0		Figure 1.56
0021_{16}				
0022_{16}				
0023_{16}				
0024_{16}	DMA0 destination pointer	DAR0		Figure 1.56
0025_{16}				
0026_{16}				
0027_{16}				
0028_{16}	DMA0 transfer counter	TCR0		Figure 1.56
0029_{16}				
$002A_{16}$				
$002B_{16}$				
$002C_{16}$	DMA0 control register	DM0CON	0 0 0 0 0 ? 0 0	Figure 1.55
$002D_{16}$				
$002E_{16}$				
$002F_{16}$				
0030_{16}	DMA1 source pointer	SAR1		Figure 1.56
0031_{16}				
0032_{16}				
0033_{16}				
0034_{16}	DMA1 destination pointer	DAR1		Figure 1.56
0035_{16}				
0036_{16}				
0037_{16}				

Address	Register name	Acronym	Value after reset	Figure number
0038_{16}	DMA1 transfer counter	TCR1		Figure 1.56
0039_{16}				
$003A_{16}$				
$003B_{16}$				
$003C_{16}$	DMA1 control register	DM1CON	0 0 0 0 0 ? 0 0	Figure 1.55
$003D_{16}$				
$003E_{16}$				
$003F_{16}$				
0040_{16}				
0041_{16}				
0042_{16}				
0043_{16}				
0044_{16}	USB Suspend interrupt control register	SUSPIC	? 0 0 0	Figure 1.16
0045_{16}				
0046_{16}	USB Resume interrupt control register	RSMIC	? 0 0 0	Figure 1.16
0047_{16}	USB SOF interrupt control register	SOFIC	0 0 ? 0 0 0	Figure 1.16
0048_{16}				
0049_{16}				
$004A_{16}$	Bus collision detection interrupt control register	BCNIC	? 0 0 0	Figure 1.16
$004B_{16}$	DMA0 interrupt control register	DM0IC	? 0 0 0	Figure 1.16
$004C_{16}$	DMA1 interrupt control register	DM1IC	? 0 0 0	Figure 1.16
$004D_{16}$	Key input interrupt control register	KUPIC	? 0 0 0	Figure 1.16
$004E_{16}$	A-D conversion interrupt control register	ADIC	? 0 0 0	Figure 1.16
$004F_{16}$	UART2 transmit interrupt control register	S2TIC	? 0 0 0	Figure 1.16
0050_{16}	UART2 receive interrupt control register	S2RIC	? 0 0 0	Figure 1.16
0051_{16}	UART0 transmit interrupt control register	S0TIC	? 0 0 0	Figure 1.16
0052_{16}	UART0 receive interrupt control register	S0RIC	? 0 0 0	Figure 1.16
0053_{16}	UART1 transmit interrupt control register	S1TIC	? 0 0 0	Figure 1.16
0054_{16}	UART1 receive interrupt control register	S1RIC	? 0 0 0	Figure 1.16
0055_{16}	Timer A0 interrupt control register	TA0IC	? 0 0 0	Figure 1.16
0056_{16}	Timer A1 interrupt control register	TA1IC	? 0 0 0	Figure 1.16
0057_{16}	Timer A2 interrupt control register	TA2IC	? 0 0 0	Figure 1.16
0058_{16}	Timer A3 interrupt control register	TA3IC	? 0 0 0	Figure 1.16
0059_{16}	Timer A4 interrupt control register	TA4IC	? 0 0 0	Figure 1.16
$005A_{16}$	Timer B0 interrupt control register	TB0IC	? 0 0 0	Figure 1.16
$005B_{16}$	Timer B1 interrupt control register	TB1IC	? 0 0 0	Figure 1.16
$005C_{16}$	USB Reset interrupt control register	RSTIC	? 0 0 0	Figure 1.16
$005D_{16}$	INT0 interrupt control register	INT0IC	0 0 ? 0 0 0	Figure 1.16
$005E_{16}$	INT1 interrupt control register	INT1IC	0 0 ? 0 0 0	Figure 1.16
$005F_{16}$	USB function interrupt control register	USBFIC	? 0 0 0	Figure 1.16

0300_{16}	USB function address register	USBA	00_{16}	Figure 1.32
0301_{16}	USB power management register	USBPM	00_{16}	Figure 1.33
0302_{16}	USB interrupt status register 1	USBIS1	00_{16}	Figure 1.34
0303_{16}	USB interrupt status register 2	USBIS2	00_{16}	Figure 1.35
0304_{16}	USB interrupt enable register 1	USBIE1	FF_{16}	Figure 1.36
0305_{16}	USB interrupt enable register 2	USBIE2	33_{16}	Figure 1.37
0306_{16}	USB frame number low register	USBSOFL	00_{16}	Figure 1.38
0307_{16}	USB frame number high register	USBSOFH	00_{16}	Figure 1.39
0308_{16}	USB ISO control register	USBISOC	00_{16}	Figure 1.40
0309_{16}	USB DMA0 Request register	USBSAR0	00_{16}	Figure 1.41
$030A_{16}$	USB DMA1 Request register	USBSAR1	00_{16}	Figure 1.42
$030B_{16}$	USB Endpoint enable	USBEPEN	FF_{16}	Figure 1.43
$030C_{16}$				
$030D_{16}$				
$030E_{16}$				
$030F_{16}$				
0310_{16}	USB reserved			
0311_{16}	USB Endpoint 0 control/status register	EP0CS	00_{16}	Figure 1.44
0312_{16}	USB reserved			
0313_{16}	USB Endpoint 0 max packet size register	EP0MP	08_{16}	Figure 1.45

Address	Register name	Acronym	Value after reset	Figure number
0314_{16}	USB reserved			
0315_{16}	USB Endpoint 0 OUT write count	EP0WC	00_{16}	Figure 1.46
0316_{16}	USB reserved			
0317_{16}	USB reserved			
0318_{16}	USB reserved			
0319_{16}	USB Endpoint 1 IN control/status register	EP1ICS	00_{16}	Figure 1.47
$031A_{16}$	USB Endpoint 1 OUT control/status register	EP1OCS	00_{16}	Figure 1.48
$031B_{16}$	USB Endpoint 1 IN max packet size register	EP1IMP	00_{16}	Figure 1.49
$031C_{16}$	USB Endpoint 1 OUT max packet size register	EP1OMP	00_{16}	Figure 1.50
$031D_{16}$	USB Endpoint 1 OUT write count	EP1WC	00_{16}	Figure 1.51
$031E_{16}$	USB reserved			
$031F_{16}$	USB reserved			
0320_{16}	USB reserved			
0321_{16}	USB Endpoint 2 IN control/status register	EP2ICS	00_{16}	Figure 1.47
0322_{16}	USB Endpoint 2 OUT control/status register	EP2OCS	00_{16}	Figure 1.48
0323_{16}	USB Endpoint 2 IN max packet size register	EP2IMP	00_{16}	Figure 1.49
0324_{16}	USB Endpoint 2 OUT max packet size register	EP2OMP	00_{16}	Figure 1.50
0325_{16}	USB Endpoint 2 OUT write count	EP2WC	00_{16}	Figure 1.51
0326_{16}	USB reserved			
0327_{16}	USB reserved			
0328_{16}	USB reserved			
0329_{16}	USB Endpoint 3 IN control/status register	EP3ICS	00_{16}	Figure 1.47
$032A_{16}$	USB Endpoint 3 OUT control/status register	EP3OCS	00_{16}	Figure 1.48
$032B_{16}$	USB Endpoint 3 IN max packet size register	EP3IMP	00_{16}	Figure 1.49
$032C_{16}$	USB Endpoint 3 OUT max packet size register	EP3OMP	00_{16}	Figure 1.50
$032D_{16}$	USB Endpoint 3 OUT write count	EP3WC	00_{16}	Figure 1.51
$032E_{16}$	USB reserved		00_{16}	
$032F_{16}$	USB reserved			
0330_{16}	USB reserved			
0331_{16}	USB Endpoint 4 IN control/status register	EP4ICS	00_{16}	Figure 1.47
0332_{16}	USB Endpoint 4 OUT control/status register	EP4OCS	00_{16}	Figure 1.48
0333_{16}	USB Endpoint 4 IN max packet size register	EP4IMP	00_{16}	Figure 1.49
0334_{16}	USB Endpoint 4 OUT max packet size register	EP4OMP	00_{16}	Figure 1.50
0335_{16}	USB Endpoint 4 OUT write count	EP4WC	00_{16}	Figure 1.51
0336_{16}	USB reserved			
0337_{16}	USB reserved			
0338_{16}	USB Endpoint 0 FIFO	EP0		Figure 1.52
0339_{16}	USB Endpoint 1 FIFO	EP1		Figure 1.52
$033A_{16}$	USB Endpoint 2 FIFO	EP2		Figure 1.52
$033B_{16}$	USB Endpoint 3 FIFO	EP3		Figure 1.52
$033C_{16}$	USB Endpoint 4 FIFO	EP4		Figure 1.52
$033D_{16}$	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}				

Address	Register name	Acronym	Value after reset	Figure number
0351_{16}				
0352_{16}				
0353_{16}				
0354_{16}				
0355_{16}				
0356_{16}				
0357_{16}				
0358_{16}				
0359_{16}				
$035A_{16}$				
$035B_{16}$				
$035C_{16}$				
$035D_{16}$				
$035E_{16}$				
$035F_{16}$				
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}	Figure 1.77
0379_{16}	UART2 bit rate generator	U2BRG		Figure 1.76
$037A_{16}$	UART2 transmit buffer register	U2TB		Figure 1.76
$037B_{16}$				
$037C_{16}$	UART2 transmit / receive control register 0	U2C0	08_{16}	Figure 1.78
$037D_{16}$	UART2 transmit / receive control register 1	U2C1	02_{16}	Figure 1.79
$037E_{16}$	UART2 receive buffer register	U2RB		Figure 1.76
$037F_{16}$				
0380_{16}		TABSR	00_{16}	Figure 1.61
0381_{16}	Count start flag			
0382_{16}	Reserved	ONSF	0 0 0 0 0 0 0 0	Figure 1.62
0383_{16}	One-shot start flag	TRGSR	00_{16}	Figure 1.62
0384_{16}	Trigger select register	UDF	00_{16}	Figure 1.61
0385_{16}	Up/down flag			
0386_{16}	Timer A0	TA0		Figure 1.61
0387_{16}				
0388_{16}	Timer A1	TA1		Figure 1.61
0389_{16}				
$038A_{16}$	Timer A2	TA2		Figure 1.61
$038B_{16}$				
$038C_{16}$	Timer A3	TA3		Figure 1.61
$038D_{16}$				
$038E_{16}$	Timer A4	TA4		Figure 1.61
$038F_{16}$				
0390_{16}	Timer B0	TB0		Figure 1.72
0391_{16}				
0392_{16}	Timer B1	TB1		Figure 1.72
0393_{16}				
0394_{16}	Timer B2	TB2		Figure 1.72
0395_{16}				
0396_{16}	Timer A0 mode register	TA0MR	00_{16}	Figure 1.63 Figure 1.64 Figure 1.66 Figure 1.68
0397_{16}	Timer A1 mode register	TA1MR	00_{16}	Figure 1.63 Figure 1.64 Figure 1.66 Figure 1.68

Address	Register name	Acronym	Value after reset	Figure number
0398_{16}	Timer A2 mode register	TA2MR	00_{16}	Figure 1.63 Figure 1.64 Figure 1.65 Figure 1.66 Figure 1.68
0399_{16}	Timer A3 mode register	TA3MR	00_{16}	Figure 1.63 Figure 1.66 Figure 1.65 Figure 1.68
$039A_{16}$	Timer A4 mode register	TA4MR	00_{16}	Figure 1.63 Figure 1.65 Figure 1.66 Figure 1.68
$039B_{16}$	Timer B0 mode register	TB0MR	0 0 ? (0) 0 0 0 0	Figure 1.71
$039C_{16}$	Timer B1 mode register	TB1MR	0 0 ? (0) 0 0 0 0	Figure 1.71
$039D_{16}$	Timer B2 mode register	TB2MR	0 0 ? (0) 0 0 0 0	Figure 1.71
$039E_{16}$				
$039F_{16}$				
$03A0_{16}$	UART0 transmit / receive mode register	U0MR	00_{16}	Figure 1.77
$03A1_{16}$	UART0 bit rate generator	U0BRG		Figure 1.76
$03A2_{16}$ $03A3_{16}$	UART0 transmit buffer register	U0TB		Figure 1.76
$03A4_{16}$	UART0 transmit / receive control register 0	U0C0	08_{16}	Figure 1.78
$03A5_{16}$	UART0 transmit / receive control register 1	U0C1	02_{16}	Figure 1.79
$03A6_{16}$ $03A7_{16}$	UART0 receive buffer register	U0RB		Figure 1.76
$03A8_{16}$	UART1 transmit / receive mode register	U1MR	00_{16}	Figure 1.77
$03A9_{16}$	UART1 bit rate generator	U1BRG		Figure 1.76
$03AA_{16}$ $03AB_{16}$	UART1 transmit buffer register	U1TB		Figure 1.76
$03AC_{16}$	UART1 transmit / receive control register 0	U1C0	08_{16}	Figure 1.77
$03AD_{16}$	UART1 transmit / receive control register 1	U1C1	02_{16}	Figure 1.79
$03AE_{16}$ $03AF_{16}$	UART1 receive buffer register	U1RB		Figure 1.76
$03B0_{16}$	UART transmit / receive control register 2	UCON	(0) 0 0 0 0 0 0 0	Figure 1.80
$03B1_{16}$				
$03B2_{16}$				
$03B3_{16}$				
$03B4_{16}$				
$03B5_{16}$				
$03B6_{16}$				
$03B7_{16}$				
$03B8_{16}$	DMA0 Request cause select register	DM0SL	00_{16}	Figure 1.54
$03B9_{16}$				
$03BA_{16}$	DMA1 Request cause select register	DM1SL	00_{16}	Figure 1.54
$03BB_{16}$				
$03BC_{16}$ $03BD_{16}$	CRC data register	CRCD		Figure 1.105
$03BE_{16}$	CRC input register	CRCIN		Figure 1.105
$03BF_{16}$				
$03C0_{16}$ $03C1_{16}$	A-D register 0	AD0		Figure 1.98
$03C2_{16}$ $03C3_{16}$	A-D register 1	AD1		Figure 1.98
$03C4_{16}$ $03C5_{16}$	A-D register 2	AD2		Figure 1.98
$03C6_{16}$ $03C7_{16}$	A-D register 3	AD3		Figure 1.98
$03C8_{16}$ $03C9_{16}$	A-D register 4	AD4		Figure 1.98
$03CA_{16}$ $03CB_{16}$	A-D register 5	AD5		Figure 1.98

Address	Register name	Acronym	Value after reset	Figure number
$03CC_{16}$	A-D register 6	AD6		Figure 1.98
$03CD_{16}$				
$03CE_{16}$	A-D register 7	AD7		Figure 1.98
$03CF_{16}$				
$03D0_{16}$				
$03D1_{16}$				
$03D2_{16}$				
$03D3_{16}$				
$03D4_{16}$	A-D control register 2	ADCON2	0	Figure 1.98
$03D5_{16}$				
$03D6_{16}$	A-D control register 0	ADCON0	0 0 0 0 0 ? ? ?	Figure 1.97
$03D7_{16}$	A-D control register 1	ADCON1	00_{16}	Figure 1.97
$03D8_{16}$				
$03D9_{16}$				
$03DA_{16}$				
$03DB_{16}$	Frequency synthesizer clock control	FSCCR	00_{16}	Figure 1.28
$03DC_{16}$	Frequency synthesizer control	FSC	60_{16}	Figure 1.27
$03DD_{16}$	Frequency synthesizer multiplier control	FSM	FF_{16}	Figure 1.25
$03DE_{16}$	Frequency synthesizer prescaler control	FSP	FF_{16}	Figure 1.24
$03DF_{16}$	Frequency synthesizer divider	FSD	FF_{16}	Figure 1.26
$03E0_{16}$	Port P0	P0		Figure 1.110
$03E1_{16}$	Port P1	P1		Figure 1.110
$03E2_{16}$	Port P0 direction register	PD0	00_{16}	Figure 1.109
$03E3_{16}$	Port P1 direction register	PD1	00_{16}	Figure 1.109
$03E4_{16}$	Port P2	P2		Figure 1.110
$03E5_{16}$	Port P3	P3		Figure 1.110
$03E6_{16}$	Port P2 direction register	PD2	00_{16}	Figure 1.109
$03E7_{16}$	Port P3 direction register	PD3	00_{16}	Figure 1.109
$03E8_{16}$				
$03E9_{16}$				
$03EA_{16}$				
$03EB_{16}$				
$03EC_{16}$	Port P6	P6		Figure 1.110
$03ED_{16}$	Port P7	P7		Figure 1.110
$03EE_{16}$	Port P6 direction register	PD6	00_{16}	Figure 1.109
$03EF_{16}$	Port P7 direction register	PD7	00_{16}	Figure 1.109
$03F0_{16}$	Port P8	P8		Figure 1.110
$03F1_{16}$				
$03F2_{16}$	Port P8 direction register	PD8	0 0 0 0 0 0 0	Figure 1.109
$03F3_{16}$				
$03F4_{16}$	Port P10	P10		Figure 1.110
$03F5_{16}$				
$03F6_{16}$	Port P10 direction register	PD10	00_{16}	Figure 1.109
$03F7_{16}$				
$03F8_{16}$				
$03F9_{16}$				
$03FA_{16}$	P2 drive capacity	P2DR	00_{16}	Figure 1.112
$03FB_{16}$	Timer A Output Drive Capacity	TADR	00_{16}	Figure 1.112
$03FC_{16}$	Pull-up control register 0	PUR0	00_{16}	Figure 1.111
$03FD_{16}$	Pull-up control register 1	PUR1	00_{16}	Figure 1.111
$03FE_{16}$				
$03FF_{16}$				

1.2.5 Reset

There are two types of resets: hardware and software. In both cases, operation is the same after the reset.

1.2.5.1 Hardware reset

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 f(X_{IN}) 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 1.9 shows an example of a reset circuit. Figure 1.10 shows the reset sequence.

.

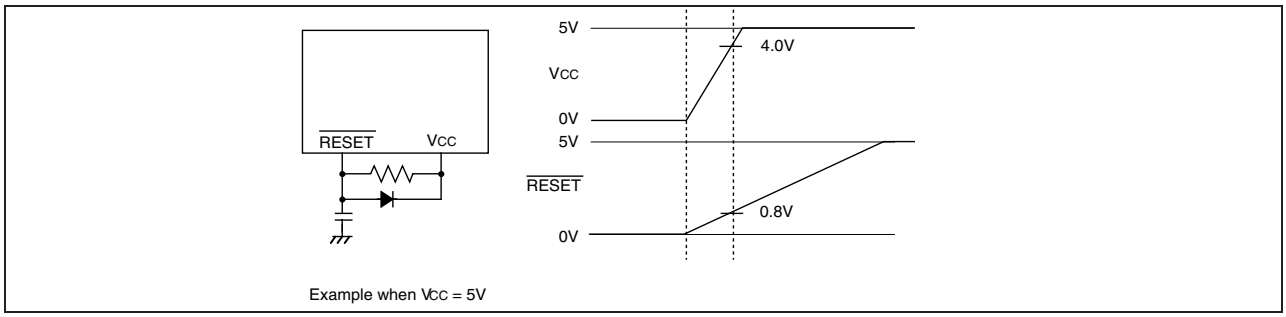


Figure 1.9: Reset circuit

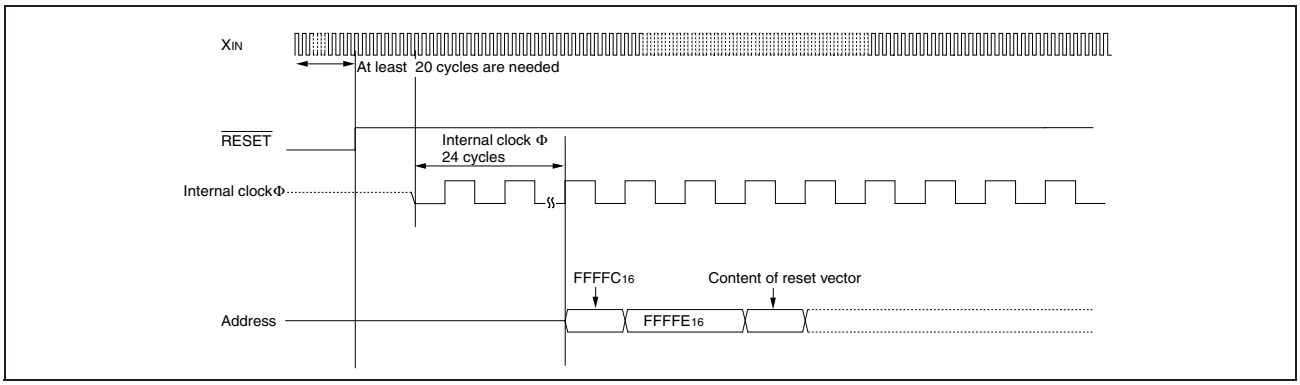


Figure 1.10: Reset sequence

When the RESET pin level = “L”, all ports change to input mode (floating.) Table 1.4 shows the status of the other pins while the RESET pin level is “L”.

Table 1.4: Pin status when RESET pin level is “L”

Pin name	Status
P0	Input port (floating)
P1	Input port (floating)
P2, P3	Input port (floating)
P6, P7, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, P10	Input port (floating)

1.2.5.2 Software Reset

Writing a “1” to bit 3 of the 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 with the following exceptions:

The contents of internal RAM are preserved

USB registers (0300_{16} to $033C_{16}$), USB attach/detach register ($001F_{16}$), USB control register ($000C_{16}$), and Frequency synthesizer related registers ($03DB_{16}$-$03DF_{16}$) values are preserved.

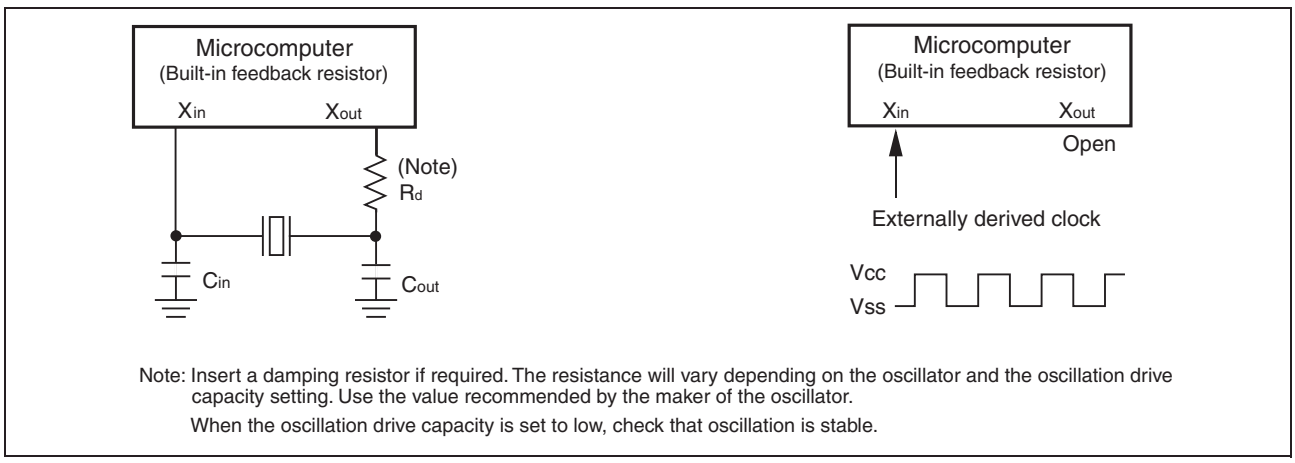
1.2.6 Clock-Generating Circuit

The clock-generating circuit contains one oscillator circuit that supplies the operating clock sources to the CPU and internal peripheral units. Table 1.5 shows the main clock generating circuits.

Table 1.5: 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

Figure 1.11 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 1.11 vary with each oscillator used. Use circuit constant values recommended by the oscillator manufacturer.



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 1.11: Examples of clock source

1.2.6.1 Clock Control

Figure 1.12 shows the block diagram of the clock-generating circuit.

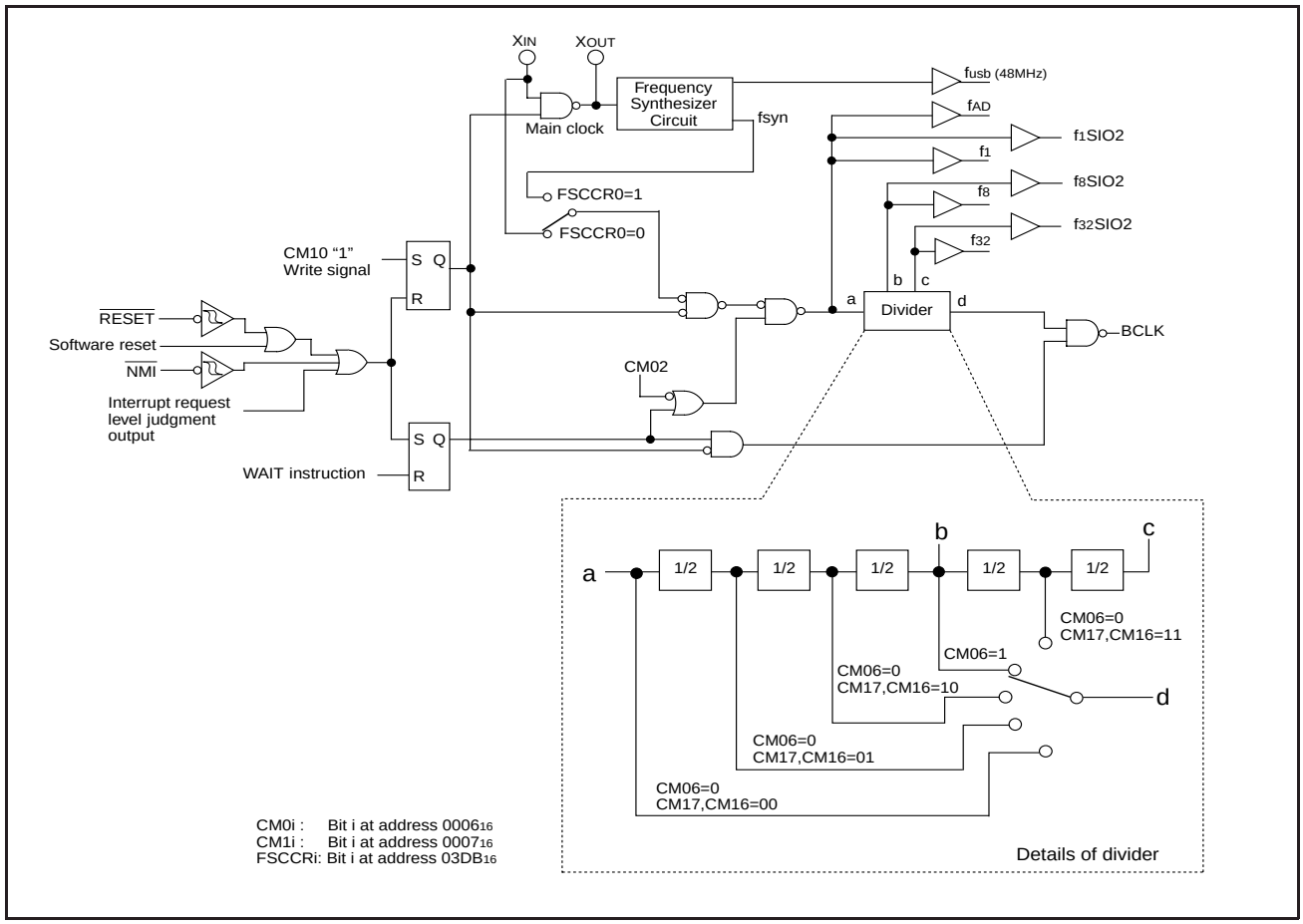


Figure 1.12: Clock-generating circuit

1.2.6.2 Clocks generated by the clock-generating circuit.

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.

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

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".

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.

• fUSB

This is the 48mHz clock that is used for USB operation. This clock is generated from the main clock by the frequency synthesizer circuit.

1.2.6.3 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/CLK_{OUT} 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 1.13 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol CM0 — Address 000616 — When reset 4816

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit	b1 b0 0 0 : I/O port P37 0 1 : Invalid 1 0 : f8 output 1 1 : f32 output	O	O
CM01			O	O
CM02	WAIT peripheral function clock stop bit	0 : Do not stop f1, f8, f32 in wait mode 1 : Stop f1, f8, f32 in wait mode	O	O
Reserved bit		Always set to "1"	O	O
Reserved bit		Always set to "0"	O	O
Reserved bit		Always set to "0"	O	O
CM06	Main clock division select bit 0 (Note 2)	0 : CM16 and CM17 valid 1 : Division by 8 mode	O	O
Reserved bit		Always set to "0"	O	O

Note 1: Set bit 0 of the Protect register (address 000A16) to "1" before writing to this register.
Note 2: Changes to "1" when shifting to stop mode.

System clock control register 1 (Note 1)

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

Symbol CM1 — Address 000716 — When reset 2016

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bit		Always set to “0”	O	O
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	O	O
CM17				

Note 1: Set bit 0 of the Protect register (address 000A16) to "1" before writing to this register.
Note 2: Changes to "1" when shifting to stop mode.
Note 3: Can be selected when bit 6 of the System clock control register 0 (address 000616) is "0".
If "1", division mode is fiixed at "8".

Figure 1.13: System clock control registers 0 and 1

1.2.7 Stop Mode

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

Because the oscillation of internal clock Φ, f_1 to f_{32}, 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 1.6 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. The I flag must also be set prior to stopping for an interrupt to cancel it. After coming out of stop mode, it is recommended that four “NOP” instructions be executed to clear the instruction queue.

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

Table 1.6: Port status during stop mode

Pin	Single-chip mode
Port	Retains status before stop mode
CLK_{OUT}	Retains status before stop mode

1.2.8 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 1.7 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 1.7: Port status during wait mode

Pin	Single-chip mode
Port	Retains status before stop mode
CLK_{OUT}	Does not stop when the WAIT peripheral function clock stop bit is “0”. When the WAIT peripheral function clock stop bit is “1”, the status immediately prior to entering WAIT mode is maintained.

1.2.9 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 1.8 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

• Division by 2 mode

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

• Division by 4 mode

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

• 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.

• Division by 16 mode

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

• No-division mode

The main clock is used as internal clock.

Table 1.8: Operating modes dictated by settings of system clock control registers 0 and 1

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

1.2.10 Power Control

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

1.2.10.1 Normal Operation Mode

• High-speed mode

Divide-by-1 frequency of the main clock become the internal clock Φ. The CPU operates with the internal clock selected. Each peripheral function operates according to its assigned clock.

• Medium-speed mode

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

1.2.10.2 Wait mode

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

1.2.10.3 Stop Mode

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

1.2.11 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 1.14 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

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

Symbol PRCR — Address 000A16 — When reset XXXXX0002

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 000616 and 000716) and frequency synthesizer registers (addresses 03DB16 to 03DF16)	0 : Write-inhibited 1 : Write-enabled	O	O
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004 16 and 000516)	0 : Write-inhibited 1 : Write-enabled	O	O
Reserved bit		Must always be set to "0"	O	O
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

Figure 1.14: Protect register

1.2.12 Interrupts

Table 1.9 and Table 1.10 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 1.9 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 1.10 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). Sixty four 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 1.15 shows the format for specifying the address.

Table 1.9: Interrupt vectors with fixed 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 $FFF3_{16}$	
$\overline{DBC}$ (Note)	$FFFF4_{16}$ to $FFFF7_{16}$	Do not use
$\overline{NMI}$	$FFFF8_{16}$ to $FFFFB_{16}$	External interrupt by $\overline{NMI}$ pin
Reset	$FFFFC_{16}$ to $FFFFF_{16}$	

Note: Interrupts used for debugging purposes only

Figure 1.15: Format for specifying interrupt vector addresses

Table 1.10: Interrupt vectors with variable addresses)

Software interrupt number	Vector table addresses Address(L) to Address(H) (Note 1)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3	BRK instruction	Cannot be masked by I flag
Software interrupt number 4	+16 to +19	USB Suspend	
Software interrupt number 6	+24 to +27	USB Resume	
Software interrupt number 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	+128 to +131 +252 to +255	Software interrupt	Cannot be masked by I flag

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

1.2.12.1 Interrupt control registers

Peripheral I/O interrupts have their own interrupt control registers. Table 1.11 shows the addresses of the interrupt control registers. Figure 1.16 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”.)

$\overline{INT}0$ and $\overline{INT}1$ are triggered by the edges of external inputs. The edge polarity is selected using the polarity select bit. (Other interrupts are described elsewhere.)

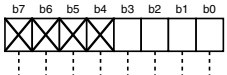
An interrupt must first be enabled before it can be used to cancel stop mode.

Table 1.11: Addresses in interrupt control register

Interrupt control register	Symbol name	Address	Interrupt control register	Symbol name	Address
USB Suspend Interrupt	SUSPIC	0044_{16}	UART1 receive	S1RIC	0054_{16}
USB 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	$004F_{16}$	USB Reset	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}	$XXXXX000_2$
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 1)	$005A_{16}$ 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 1: This bit can only be reset (= 0), but cannot be set (= 1).
Note 2: When rewriting the interrupt control register, do so at a point that does not generate an interrupt requst for that register. For details, see the interrupt precaution section.

Symbol	Address	When reset
INTiIC (i= 0, 1)	$005D_{16}$, $005E_{16}$	$XX00X000_2$
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		Must always be 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.
Note 3: When rewriting the interrupt control register, do so at a point that does not generate an interrupt requst for that register. For details, see the interrupt precaution section.

Figure 1.16: Interrupt control registers

1.2.12.2 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 > $\overline{\text{NMI}}$ > $\overline{\text{DBC}}$ > Watchdog 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 1.17 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, NMI, DBC, watchdog timer, single-step, address-match, BRK instruction, overflow, and undefined instruction interrupts are accepted regardless of the interrupt enable flag (I flag).

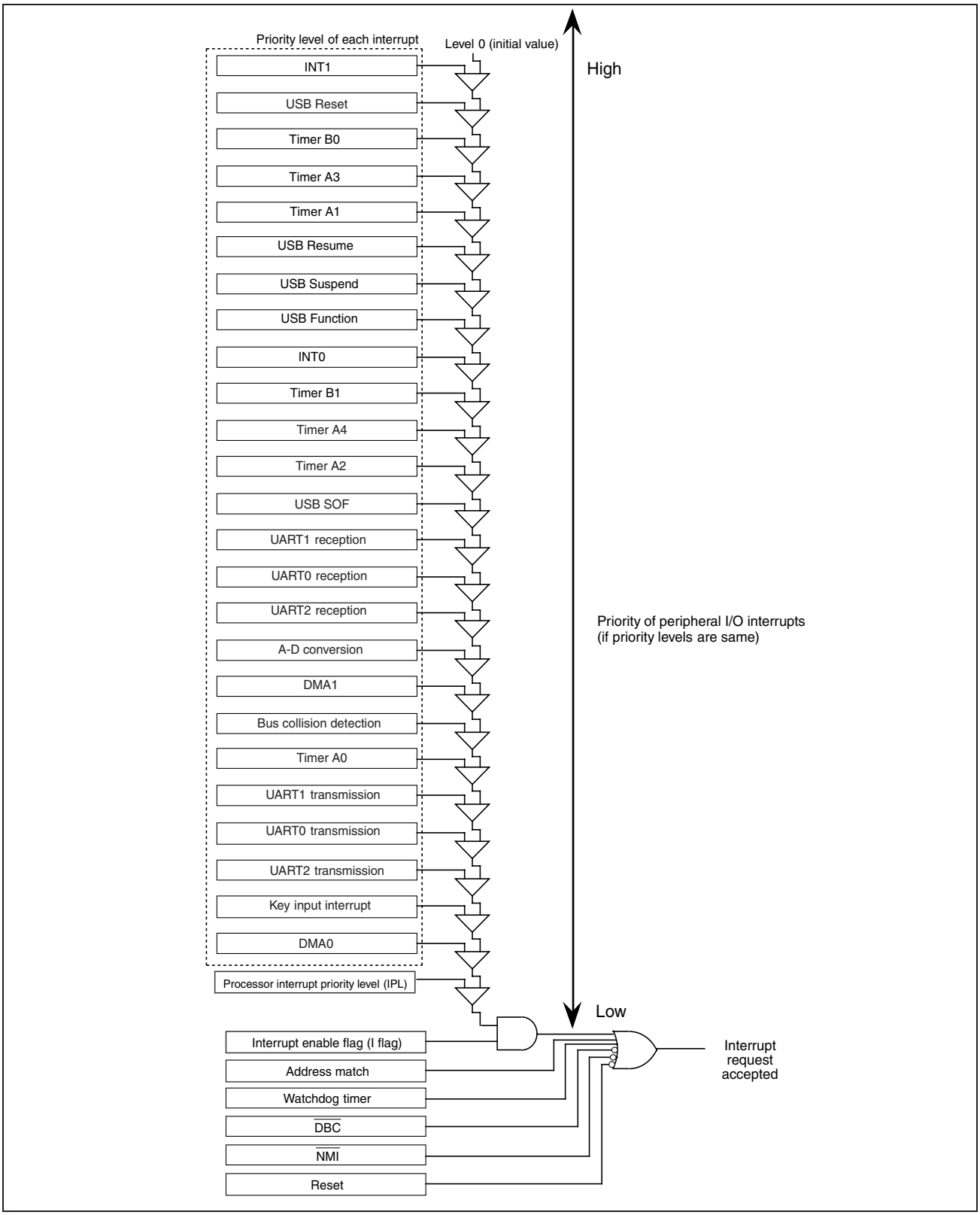


Figure 1.17: Interrupt resolution circuit

1.2.12.3 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, $\overline{\text{NMI}}$, $\overline{\text{DBC}}$, watchdog timer, single-step, address-match, BRK instruction, overflow, and undefined instruction interrupts. Table 1.12 shows how the IPL changes when interrupt requests are received.

Table 1.12: Change of IPL state when interrupt request are accepted

Interrupt	Change of IPL
Reset	Level 0 (000_2), is set
$\overline{\text{NMI}}$	Level 7 (111_2), is set
$\overline{\text{DBC}}$	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

1.2.13 $\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when the input to the $P8_5/\overline{\text{NMI}}$ pin changes from “H” to “L”. The $\overline{\text{NMI}}$ interrupt is a non-maskable external interrupt. The pin level can be checked in the Port $P8_5$ 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 $\overline{\text{NMI}}$ function, be sure to connect the $\overline{\text{NMI}}$ pin to VCC. Because the $\overline{\text{NMI}}$ interrupt is non-maskable, it cannot be disabled.

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

1.2.14 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 1.18 shows the block diagram of the key-input interrupt.

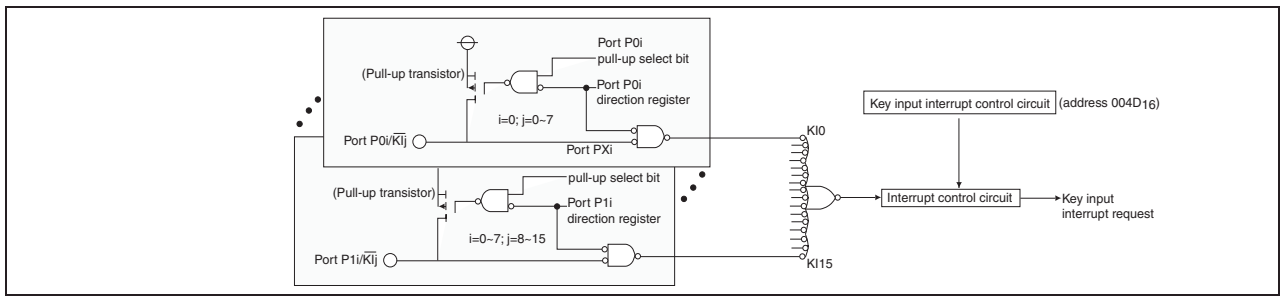


Figure 1.18: Block diagram of key input interrupt

1.2.14.1 Enable/Disable

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).

1.2.14.2 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 ($\overline{KI_0}$ through $\overline{KI_{15}}$). 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”.

1.2.14.3 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.

1.2.14.4 Related registers

Figure 1.19 shows the memory map of key-input interrupt-related registers

Address	Register name	Acronym
$004E_{16}$		
$004D_{16}$	Key input interrupt register	KUPIC
≈	≈	≈
$03E0_{16}$	Port 0	P0
$03E1_{16}$	Port 1	P1
$03E2_{16}$	Port 0 direction register	PD0
$03E3_{16}$	Port 1 direction register	PD1
$03E4_{16}$		
≈	≈	≈
$03FA_{16}$		
$03FB_{16}$		
$03FC_{16}$	Pull-up control register 0	PUR0
$03FD_{16}$	Pull-up control register 1	PUR1
$03FE_{16}$		
$03FF_{16}$		

Figure 1.19: Memory Map of key input interrupt related registers

1.2.15 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 1.20 shows the address match interrupt-related registers.

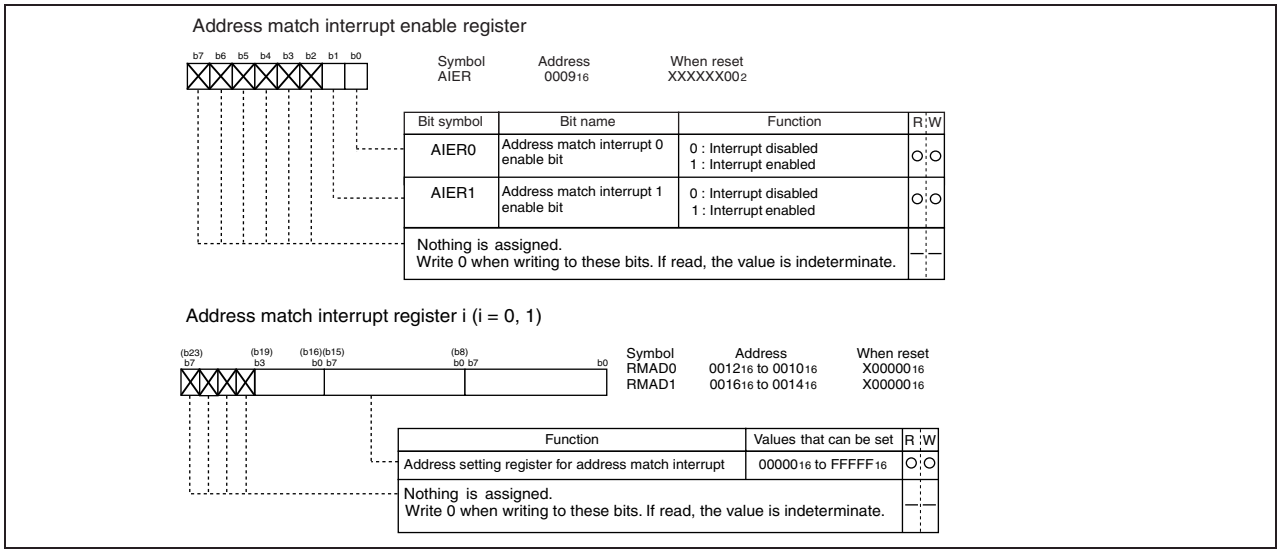


Figure 1.20: Address match interrupt-related registers

1.2.16 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 1.13 shows the periodic table for the watchdog timer.

Table 1.13: Watchdog timer periodic table (f(XIN)=12MHz)

CM06	CM17	CM16	Internal clock Φ	WDC7	Period (Note)
0	0	0	12MHz	0	Approx. 43.7ms
				1	Approx. 349.5ms
0	0	1	6MHz	0	Approx. 87.4ms
				1	Approx. 699.1ms
0	1	0	3MHz	0	Approx. 174.8ms
				1	Approx. 1.40s
0	1	1	0.75MHz	0	Approx. 699.1ms
				1	Approx. 5.59s
1	Invalid	Invalid	1.5MHz	0	Approx. 349.5ms
				1	Approx. 2.80s

Note: The watchdog timer's period is subject to some error due to 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, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address $000E_{16}$).

Figure 1.21 shows the block diagram of the watchdog timer. Figure 1.22 shows the watchdog timer-related registers.

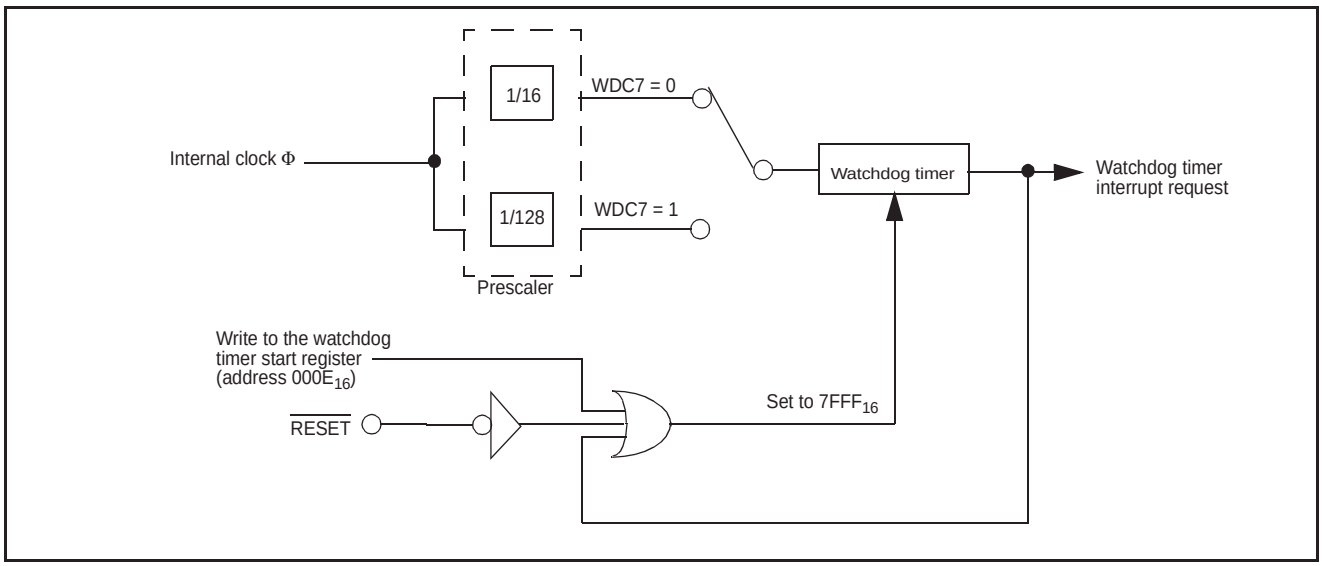


Figure 1.21: Block diagram of watchdog timer

Watchdog timer control register (Note)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

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

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

Note: Set the desired prescale value before initializing the Watchdog timer.

Watchdog timer start register

b7	b0

Symbol: WDTS | Address: $000E_{16}$ | When reset: Indeterminate

Function	R	W
The Watchdog timer is initialized and starts counting after the first write instruction to this register after reset. Writing any value to this register resets the counter to $7FFF_{16}$.	×	O

Figure 1.22: Watchdog timer control and start registers

1.2.17 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 $f(X_{IN})$. A block diagram of the circuit is shown in Figure 1.23.

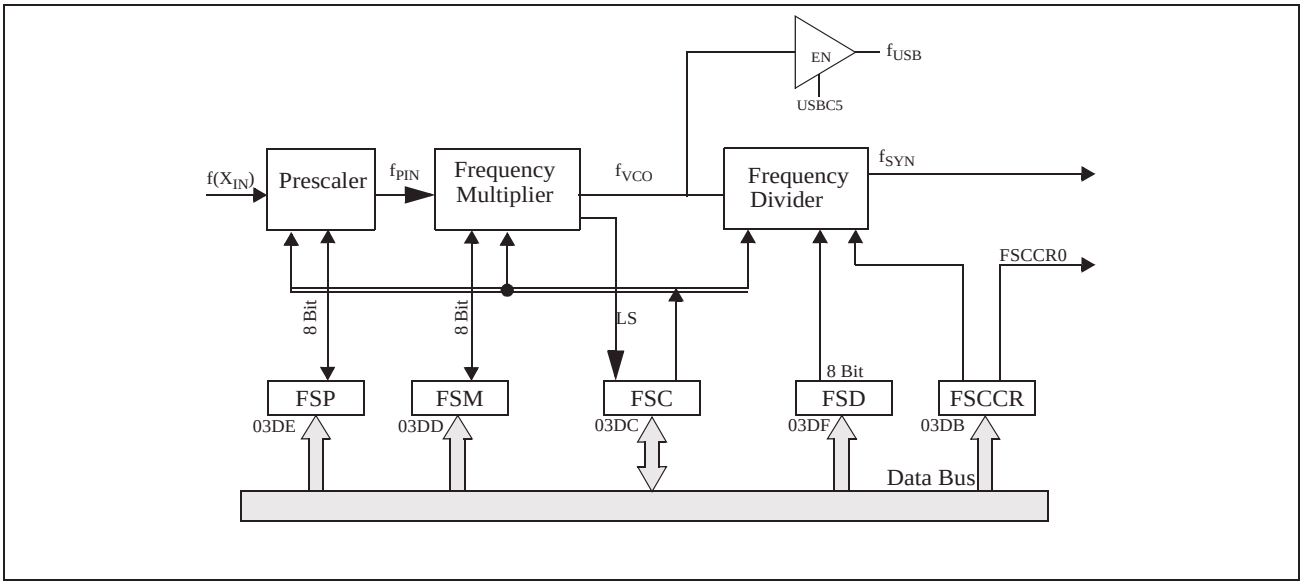


Figure 1.23: Frequency Synthesizer Circuit

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 $f(X_{IN})$ 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.

1.2.17.1 Prescaler

Clock f_{PIN} is a divided down version of clock $f(X_{IN})$ (see Figure 1.24). The relationship between f_{PIN} and the clock input to the prescaler $f(X_{IN})$ is as follows:

- $f_{PIN} = f(X_{IN}) / 2(n+1)$ where n is a decimal number between 0 and 254.
  Setting FSP to 255 disables the prescaler and $f_{PIN} = f(X_{IN})$.
- Note: $f(X_{IN})$ frequency below 1 MHz is not recommended.

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

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

f_{PIN}	FSP		f(Xin)
	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

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

Figure 1.24: Frequency Synthesizer Prescaler Register (FSP)

1.2.17.2 Multiplier

Clock f_{VCO} is a multiplied up version of clock f_{PIN} (See Figure 1.25). 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 1: *n* must be chosen such that f_{VCO} equals 48 MHz.

  Note 2: Minimum f_{PIN} is 1 MHz.

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

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

f_{PIN}	FSM		f_{VCO}
	Dec(n)	Hex(n)	
1 MHz	23	17	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 1.25: Frequency Synthesizer Multiply Register (FSM)

1.2.17.3 Divider

Clock f_{SYN} is a divided down version of clock f_{VCO} (See Figure 1.26). 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	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	LSB 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 1.26: 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 1.27.

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 MCU clock source is selected via the Frequency Synthesizer Clock Control register (FSCCR). See Figure 1.28.

Note: None of the registers must be written to once the frequency synthesizer is enabled and used as the system clock source (FSCCR register, address $03DB_{16}$, bit “0” set to “1”) because it will cause the output of the PLL to freeze. Switch system back to f(X_{IN}) and disable before modifying PLL registers.

Frequency Synthesizer Control Register

b7	b6	b5	b4	b3	b2	b1	b0
			0	0			

Symbol: FSC — Address: $03DC_{16}$ — When reset: 60_{16}

Bit symbol	Bit name	Function	R	W
FSE	Frequency Synthesizer Enable	0 : Disable 1 : Enabled	O	O
VCO0	VCO Gain Control	Bit 2 Bit 1 0 0: Lowest Gain (Note) 0 1: Low Gain 1 0: High Gain 1 1: Highest Gain	O	O
VCO1				
Reserved bit		Must always be set to "0"	O	O
CHG0	LPF Current Control	Bit 6 Bit 5 0 0: Disabled 0 1: Low Current 1 0: Intermediate Current (Note) 1 1: High Current	O	O
CHG1				
LS	Frequency Synthesizer Lock Status	0: Unlocked 1: Locked	O	O

Note : Recommended

Figure 1.27: Frequency Synthesizer Control Register (FSC)

Frequency Synthesizer Clock Control Register

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

Symbol: FSCCR — Address: $03DB_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
FSCCR0	Clock source selection	0 : X_{IN} 1 : fsyn	O	O
Reserved		Must always be set to "0"	O	O

Figure 1.28: Frequency Synthesizer Clock Control Register (FSCCR)

1.2.18 Universal Serial Bus

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

- Complete USB Specification (version 1.1) Compatibility
- Error-handling capabilities
- FIFOs:
  - Endpoint 0:IN/OUT 32-byte
  - Endpoint 1:IN 128-byteOUT 128-byte
  - Endpoint 2:IN 32-byteOUT 32-byte
  - Endpoint 3:IN 32-byteOUT 32-byte
  - Endpoint 4:IN 32-byteOUT 32-byte
- Nine endpoints - control endpoint (Endpoint 0 - bi-directional) plus four IN and four OUT endpoints
- Complete device configuration
- Support of all device commands
- Supports of 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

1.2.18.1 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 1.29). 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

1.2.18.1.1 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 items pertaining to the USB protocol such as handling inter-packet time-outs and packet ID (PID) decoding.

1.2.18.1.2 Generic Function Interface

The GFI handles 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-transmission the current data set; write pointer reversal for reception of the last data set again and data toggle synchronization.

1.2.18.1.3 Serial Engine Interface Unit

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

1.2.18.1.4 Microcontroller Interface

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

1.2.18.1.5 USB Transceiver

The USB transceiver, designed to interface with the physical layer of the USB, is compliant with the USB Specification (version 1.1) for full-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 1.5.4 “USB Transceiver” for more detailed information.

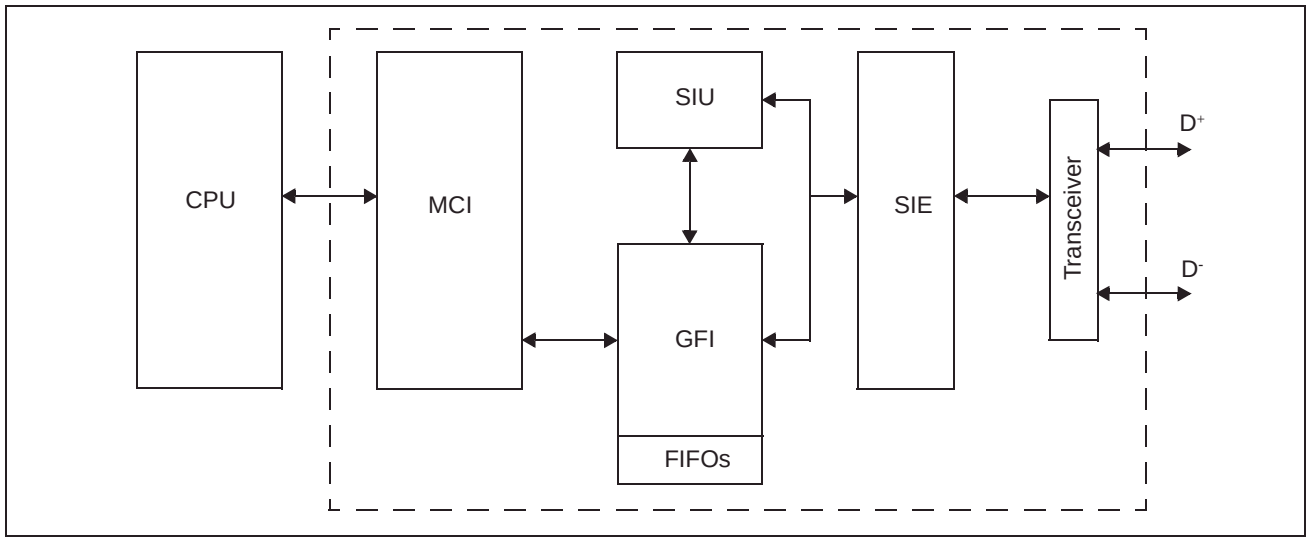


Figure 1.29: USB Function Control Unit Block Diagram

1.2.18.2 USB Interrupts

There are five USB interrupts in this device:

- USB Function interrupt
- USB Reset interrupt
- USB Suspend interrupt
- USB Resume interrupt
- USB Start-of-Frame (SOF) interrupt.

The first four interrupts are used to control the data flow and USB power. The SOF interrupt is used to monitor the transfer of isochronous (ISO) data. Each of the five USB interrupts is enabled by setting the corresponding bit in the Interrupt Control Register of the Interrupt Control Unit. Because the USB Function Interrupt has multiple interrupt sources, another level of enabling is within the USB Interrupt Registers 1 & 2.

1.2.18.2.1 USB Function Interrupt

The USB Function Interrupt can be triggered by 10 sources; many of these may be cause by several different events. Interrupt status flags associated with each source are contained in USBIS1 and USBIS2.

Endpoints 1-4 have two interrupt status flags associated with it to control data transfer or to report a STALL/ UNDER_RUN/OVER RUN condition.

The USB Endpoint x Out Interrupt Status Flag is set when

- USB FCU successfully receives a packet of data OR
- USB FCU sets the FORCE_STALL flag or OVER_RUN flag of the Endpoint x OUT CSR.

The USB Endpoint x In Interrupt Status Flag is set when

- USB FCU successfully sends a packet of data OR
- USB FCU sets the UNDER_RUN flag of the Endpoint x IN CSR.

The USB Endpoint 0 (control endpoint) has one interrupt status bit associated with it to control data transfer or report a STALL condition.

The USB Endpoint 0 Interrupt Status Flag is set when

- USB FCU successfully receives/sends a packet of data
- Sets the SETUP_END flag or the FORCE_STALL flag, OR clears the DATA_END bit in the Endpoint 0 IN CSR.

The Overrun/Underrun Interrupt Status Flag is set when (applicable to endpoints used for isochronous data transfer)

- Overrun condition occurs in a endpoint (CPU is too slow to unload the data from the FIFO), OR
- Underrun condition occurs in an endpoint (CPU is too slow to load the data to the FIFO).

Each endpoint interrupt and overrun/underrun interrupt is enabled by setting the corresponding bit in the USB Interrupt Enable Register 1 and 2.

1.2.18.2.2 USB Reset Interrupt

The USB Reset Interrupt Status Flag is set when the USB FCU sees a SE0 present on D+/D- for at least 2.5µs. When this bit is set, all USB internal registers except INTST13 (bit5 of USBIS2) are reset to their default values. INTST13, the USB reset Interrupt Status Flag, is set to a “1” when the USB Reset is detected.

When the CPU recognizes a USB Reset Interrupt, it needs to re initialize the USB FCU so that the USB operation can behave properly. It must also clear INTST13 by writing a “1” to this bit to allow a USB Reset Interrupt request to occur the next time a USB Reset is detected.

Register RSTIC contains the USB Reset Interrupt’s request bit and its interrupt priority select bits which are used to enable the interrupt and set its software priority level.

1.2.18.2.3 USB Suspend and Resume Interrupts

The USB Suspend Interrupt is set when the USB FCU does not detect any bus activity on D+/D- (in J-state) for at least 3ms.

The USB Suspend Signaling Interrupt Status Flag (INTST15, bit 7 of USBIS2) is set to a “1” when the USB Suspend is detected. The CPU must clear INTST15 by writing a “1” to this bit to allow a USB Suspend Interrupt request to occur the next time a USB Suspend is detected.

The USB Resume Signaling Interrupt Status Flag is set when a USB FCU is in the suspend state and detects non-idle signaling on the D+/D-.

Register SUSPIC contains the USB Suspend Interrupt’s request bit and its interrupt priority select bits which are used to enable the interrupt and set its software priority level.

The USB Resume Interrupt request is set when the USB FCU is in the suspend state and detects non-idle signaling on D+/D-.

The USB Signaling Interrupt Status Flag (INTST14, bit 6 of USBIS2) is set to a “1” when the USB Resume is detected. The CPU must clear INTST14 by writing a “1” to this bit to allow a USB Resume Interrupt request to occur the next time a USB Resume is detected.

Register RSMIC contains the USB Resume Interrupt’s request bit and its interrupt priority select bits, which are used to enable the interrupt an set its software priority level.

1.2.18.2.4 USB SOF Interrupt

The USB SOF (Start-Of-Frame) Interrupt is used to control the transfer of isochronous data. The USB FCU generates a USB SOF Interrupt request when a start-of-frame packet is received.

Register SOFIC contains the USB SOF Interrupt's request bit and its interrupt priority select bits, which are used to enable the interrupt and set its software priority level.

1.2.18.3 USB Endpoint FIFOs

The USB FCU has an IN (transmit) FIFO and an OUT (receive) FIFO for each endpoint. Each endpoint (except endpoint 0) can be configured to support either single packet mode (in which only a single data packet is allowed to reside in the endpoint's FIFO) or dual packet mode (in which up to two data packets are allowed to reside in the endpoint's FIFO). Dual packet mode provides support for back-to-back transmission or back-to-back reception. The mode is automatically determined by the MAXP value. When MAXP > 1/2 of the endpoint's FIFO size, single packet mode is set. When MAXP <= 1/2 of the endpoint's FIFO size, dual packet mode is set.

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

Throughout this specification, the terms "IN FIFO" and "OUT FIFO" usually refer to the FIFOs associated with a specific endpoint.

1.2.18.3.1 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. The CPU/DMA should only write data to the IN FIFO when the IN_PKT_RDY bit of the associated IN CSR is a "0".

- **Endpoint 0 IN FIFO Operation:**

The CPU writes a "1" to the IN_PKT_RDY bit of Endpoint 0 CSR after it finishes writing a packet of data to the IN FIFO. The USB FCU clears the IN_PKT_RDY bit after the packet has been successfully transmitted to the host (i.e., ACK is received from the host) or the SETUP_END flag of the Endpoint O CSR is set to a "1".

- **Endpoint 1-4 IN FIFO Operation when AUTO_SET (bit 7 of Endpoint x IN CSR) = "0" (disabled):**

MAXP > 1/2 of the IN FIFO size: The CPU writes a "1" to the IN_PKT_RDY bit of the associated IN CSR after the CPU/DMAC finishes writing a packet of data to the IN FIFO. The USB FCU clears the IN_PKT_RDY bit after the packet has been successfully transmitted to the host (which is assumed for isochronous transfers and is concluded when an ACK is received from the host for non-isochronous transfers).

MAXP <= 1/2 of the IN FIFO size: The CPU writes a "1" to the IN_PKT_RDY bit of the associated IN CSR after the CPU/DMAC finishes writing a packet of data to the IN FIFO. The USB FCU clears the IN_PKT_RDY bit as soon as the IN FIFO is ready to accept another data packet. (The FIFO can hold up to two data packets at the same time in this configuration for back-to-back transmission.)

- **Endpoint 1-4 IN FIFO Operation when AUTO_SET (bit 7 of Endpoint x IN CSR) = "1" (enabled):**

MAXP > 1/2 of the IN FIFO size: When the number of bytes of data equal to the MAXP (maximum packet size) has been written to the IN FIFO by the CPU/DMAC, the USB FCU sets the IN_PKT_RDY bit of the associated IN CSR to a "1" automatically. The USB FCU clears the IN_PKT_RDY bit after the packet has been successfully transmitted to the host (which is assumed for isochronous transfers and is concluded when an ACK is received from the host for non-isochronous transfers).

MAXP <= 1/2 of the IN FIFO size: When the number of bytes of data equal to the MAXP (maximum packet size) has been written to the IN FIFO by the CPU/DMAC, the USB FCU sets the IN_PKT_RDY bit to a "1" automatically. The USB FCU clears the IN_PKT_RDY bit as soon as the IN FIFO is ready to accept another data packet. (The FIFO can hold up to two data packets at the same time in this configuration for back-to-back transmission.)

A software or a hardware flush causes the USB FCU to act as if a packet has been successfully transmitted out to the host. When there is one packet in the IN FIFO, a flush causes the IN FIFO to be empty. When there are two packets in the IN FIFO, a flush causes the older packet to be flushed out from the IN FIFO. A flush also updates the IN FIFO status bits IN_PKT_RDY and TX_NOT_EPT of the associated IN CSR.

The status of endpoint 1-4 IN FIFOs for both of the above cases can be obtained from the IN CSR of the corresponding IN FIFO as shown in Table 1.14 .

Table 1.14: TA FIFO Status

IN_PKT_RDY	TX_NOT_EPT	IN FIFO Status
0	0	No data packet in IN FIFO
0	1	One data packet in IN FIFO if MAXP <= 1/2 of the FIFO size./ Invalid when MAXP>1/2 of the FIFO size
1	0	Invalid
1	1	Two data packets in IN FIFO when MAXP <=1/2 of the FIFO size One data packet in IN FIFO when MAXP > 1/2 of the FIFO size

1.2.18.3.2 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 of the corresponding OUT CSR to a "1". The CPU/DMAC should only read data from the OUT FIFO when 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 sets bit SERVICED_OUT_PKT_RDY to a "1" to clear the OUT_PKT_RDY bit after the packet of data has been unloaded from the OUT FIFO by the CPU.

- **Endpoint 1-4 OUT FIFO Operation when AUTO_CLR (bit 7 of Endpoint x OUT CSR) = "0" (disabled):**

MAXP > 1/2 of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit of the associated IN CSR 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 has been unloaded from the OUT FIFO by the CPU/DMAC.

MAXP <= 1/2 of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit of the associated IN CSR 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 has been unloaded from the OUT FIFO by the CPU/DMAC. If another packet is in the OUT FIFO, the OUT_PKT_RDY bit will be set to a "1" again almost immediately (such that it may appear that the OUT_PKT_RDY bit remains a "1"). In this configuration, the FIFO can store up to two data packets at the same time for back-to-back reception.

- **Endpoint 1-4 OUT FIFO Operation when AUTO_CLR (bit 7 of Endpoint x OUT CSR) = "1" (enabled):**

MAXP > 1/2 of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit of the associated IN CSR 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) has been unloaded from the OUT FIFO by the CPU/DMAC.

MAXP <= 1/2 of the OUT FIFO size: The USB FCU sets the OUT_PKT_RDY bit of the associated IN CSR 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) has been unloaded from the OUT FIFO by the CPU/DMAC. If another packet is in the OUT FIFO, the OUT_PKT_RDY bit will be set to a "1" again almost immediately (such that it may appear that the OUT_PKT_RDY bit remains a "1"). In this configuration, the FIFO can store up to two data packets at the same time for back-to-back reception.

A software flush causes the USB FCU to act as if a packet has been unloaded from the OUT FIFO. If there is one packet in the OUT FIFO, a flush will cause the OUT FIFO to be empty. If there are two packets in the OUT FIFO, a flush will cause the older packet to be flushed out from the OUT FIFO.

1.2.18.3.3 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.

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.

1.2.18.4 USB Special Function Registers

The MCU controls USB operation through the use of special function registers (SFR). This section describes 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.
- Writing a "1" to a register that can only be set to a "0" by the CPU has not effect on that register bit.

Each figure and description of the special function registers details this operation.

All USB Special Function Registers, with the exception of USB Attach/Detach ($001F_{16}$) and USB control ($000C_{16}$) must use byte access. Work access is prohibited for USB internal registers (0300_{16} - $033C_{16}$).

The contents of all USB Special Functions Registers, including USB Attach/Detach and USB Control, are preserved on a software reset.

1.2.18.4.1 USB Attach/Detach Register

The USB Attach / Detach Register is shown in Figure 1.30. The register is used to attach and detach the USB function from a USB host without physically disconnecting the USB cable. This functionality is enabled by setting $P8_3$_Second to a "1". Doing this forces $P8_3$ to operate as a pull-up for D+ (through an external 1.5k ohm resistor). The port driver is tri-stated and a "1" is always read from the port bit in this mode. When the ATTACH/DETACH bit is a "1" (and $P8_3$_Second is a "1"), $P8_3$ is driven with the voltage on EXTCAP, causing D+ to be pulled up and the host to detect an attach. When the ATTACH/DETACH bit is a "0" (and $P8_3$_Second is a "1"), $P8_3$ is tri-stated, causing D+ to be pulled down (through the cable and 15k ohm resistor on the host/hub side) and a detach to be registered by the host. A 1.5k ohm pull-up resistor must be connected externally from $P8_3$ to D+ when this functionality is used. When it is not used, the 1.5k ohm resistor should be placed between EXTCAP and D+.

USB Attach/Detach Register

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

Symbol: USBAD — Address: $001F_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
$P8_3$_2nd	Port 8_3-Second	0 : Normal mode for Port $P8_3$ 1 : Forces Port $P8_3$ to operate as pull up for D+.	○	○
Attach/ Detach	Attach/Detach	0 : Tri-states, $P8_3$ causing the host to detect a detach 1 : Drives $P8_3$ with voltage on EXTCAP, causing the host to detect an attach	○	○
Reserved		Must always be set to "0"	○	○

Figure 1.30: USB Attach/Detach Register

1.2.18.4.2 USB Control Register

The USB Control Register, shown in Figure 1.31**,** is used to control the USB FCU. 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.

USB Control Register

b7	b6	b5	b4	b3	b2	b1	b0
					0	0	0

Symbol: USBC | Address: 000C16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
Reserved		Must always be set to "0"	○	○
USBC3	Tranceiver voltage converter High/Low current mode selection	0: High current mode (Note 1) 1: Low current mode (Note 2)	○	○
USBC4	USB tranceiver voltage converter enable bit	0: Disabled 1: Enabled	○	○
USBC5	USB clock enable bit	0: Disabled 1: Enabled	○	○
USBC6	USB $\overline{SOF}$ port select bit	0: Disabled (Note 3) 1: Enabled	○	○
USBC7	USB enable bit	0: Disabled (Note 4) 1: Enabled	○	○

Note 1: For USB normal operation
Note 2: For USB suspend operation
Note 3: $P8_6$ is used as GPIO pin
Note 4: All USB internal registers are held at their default values.

Figure 1.31: USB Control Register

1.2.18.4.3 USB Function Address Register

The USB Function Address Register, shown in Figure 1.32, 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}. For the procedures on how to update this register, refer to Application Notes USB Consecutive Set Address.

USB Function Address Register

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol: USBA | Address: 030016 | When reset: 0016

Bit symbol	Bit name	Function	R	W
FUNAD0-6	Function Address	7-bit programmable Function Address	○	○
Reserved		Must always be set to "0"	○	○

Figure 1.32: USB Function Address Register

1.2.18.4.4 USB Power Management Register

The USB Power Management Register, shown in Figure 1.33, is used for power management in the USB FCU.

- SUSPEND Detection Flag:

When the USB FCU does not detect any bus activity on D+/D- for at least 3ms (and D+/D- are in the J-state), it sets the Suspend Detection Flag and generates an interrupt. This bit is cleared when signaling from the host is detected on D+/D- (which sets the Resume Detection Flag and generates an interrupt), or the Remote Wake-up Bit is set and then cleared by the CPU. If the USB clock was disabled during the suspend state, the SUSPEND Detection Flag is not cleared until after the USB clock is re-enabled.

- RESUME Detection Flag:

When the USB FCU is in the suspend state and detects activity on D+/D- from the host, it sets the Resume Detection Flag and generates an interrupt. The CPU writes a “1” to INTST14 (bit 6 of USB Interrupt Status Register 2) to clear this flag.

- WAKEUP Control Bit:

The CPU writes a “1” to the WAKEUP Control Bit for remote wake-up. While this bit is set and the USB FCU is in suspend mode, resume signaling is sent 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.

USB Power Management Register

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

Symbol: USBPM — Address: 030116 — When reset: 0016

Bit symbol	Bit name	Function	R	W
SUSPEND	USB Suspend Detection Flag	0 : No USB suspend signal detected 1 : USB suspend signal detected	○	○ Note 1
RESUME	USB Resume Detection Flag	0 : No USB resume signal detected 1 : USB resume signal detected	○	○ Note 1
WAKEUP	USB Remote Wakeup Bit	0 : End remote resume signaling 1 : Remote resume signaling (Note 2)	○	○
Reserved		Must always be set to "0"	○	○

Note 1: Write "0" only or Read
Note 2: If SUSPEND = "1"

Figure 1.33: USB Power Management Register

1.2.18.4.5 USB Interrupt Status Registers 1 and 2

USB Interrupt Status Registers 1 and 2**,** shown in Figure 1.34 and Figure 1.35, are used to indicate the condition that caused a USB function interrupt and USB Reset, Suspend and Resume Interrupts to the CPU. A "1" indicates the corresponding condition caused an interrupt. The USB Interrupt Status Register bits can be cleared by writing a "1" to the corresponding bit.

INTST0 is set to a "1" by the USB FCU when (in Endpoint 0 CSR):

- A packet of data is successfully received (EP0CSR0 - OUT_PKT_RDY is set by the USB FCU)
- A packet of data is successfully sent (EP0CSR - IN_PKT_RDY is cleared by the USB FCU)
- EP0CSR3 (DATA_END) bit is cleared by the USB FCU
- EP0CSR4 (FORCE_STALL) bit is set by the USB FCU
- EP0CSR5 (SETUP_END) bit is set by the USB FCU

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

- A packet of data is successfully sent (INXCSR0 - IN_PKT_RDY is cleared by the USB FCU)
- INXCSR1 (UNDER_RUN) bit is set by the USB FCU

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

- A packet of data is successfully received (OUTXCSR0 - OUT_PKT_RDY is set by the USB FCU)
- OUTXCSR1 (OVER_RUN) bit is set by the USB FCU
- OUTXCSR4 (FORCE_STALL) bit is set by the USB FCU

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

INTST13 is set to a "1" by the USB FCU when a USB reset signaling from the host is received. All internal register bits except this bit are reset to their default values when the USB reset is received.

INTST14 is set to a "1" by the USB FCU when the USB FCU is in the suspend state and non-idle signaling is received from D+/D-.

INTST15 is set to a "1" by the USB FCU when D+/D- are in the idle state for more than 3ms.

USB Interrupt Status Register 1

b7	b6	b5	b4	b3	b2	b1	b0
						0	

Symbol: USBIS1 — Address: 030216 — When reset: 0016

Bit symbol	Bit name	Function	R	W
INTST0	USB Endpoint 0 Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
Reserved		Must always be set to "0"	○	○
INTST2	USB Endpoint 1 IN Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST3	USB Endpoint 1 OUT Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST4	USB Endpoint 2 IN Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST5	USB Endpoint 2 OUT Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST6	USB Endpoint 3 IN Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST7	USB Endpoint 3 OUT Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○

Figure 1.34: USB Interrupt Status Register 1

USB Interrupt Status Register 2

b7	b6	b5	b4	b3	b2	b1	b0
				0	0		

Symbol	Address	When reset
USBIS2	030316	0016

Bit symbol	Bit name	Function	R	W
INTST8	USB Endpoint 4 IN Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST9	USB Endpoint 4 OUT Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
Reserved		Must always be set to "0"	○	○
INTST12	USB Overrun/Underrun Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST13	USB Reset Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST14	USB Resume Signaling Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○
INTST15	USB Suspend Signaling Interrupt Status Flag	0 : No interrupt request issued 1 : Interrupt request issued	○	○

Figure 1.35: USB Interrupt Status Register 2

1.5.0.0.1 Clearing USB Interrupt Status Registers

The USB Interrupt Status Register 1 and 2 are used to indicate pending interrupts for a given source. The USB FCU sets the interrupt status bits. The CPU writes a “1” to each status bit to clear it.

Because the USB Function Interrupt has multiple sources that can generate an interrupt, it is recommended that the user first read the two status registers and store them in variables then write back the same value for clearing all the existing interrupts that were pending when the status registers were read. This procedure prevents any interrupt that occurs after the status registers are read from being cleared by the ‘write-back’ operation. The CPU must read, then write both status registers, writing to status register 1 first and status register 2 second to guarantee proper operation. The upper three bits of the value written back to USBIS2 should always be “000” to prevent any of the USB Reset, Suspend and Resume Status Flags from being cleared.

The USB Reset, Suspend and Resume Status Flags are contained in USBIS2 along with the USB Endpoint 4 In/Out Interrupt Status Flags and the USB Overrun/Underrun Interrupts Status Flag. Because the flags are not all sources for the same interrupt, use caution when clearing one or more of the flags to avoid inadvertently clearing other flags. The Reset, Suspend and Resume Status Flags should be cleared individually by writing a byte value with at “1” only at the position corresponding to the flag to be cleared. The USB Endpoint 4 In/Out Interrupt status Flags and the USB Overrun/Underrun Interrupt Status Flag should be cleared as described in the preceding paragraph because they are sourced for the USB Function Interrupt.

“Read-modify-write” instructions, such as “BCLR” and “BSET”, should not be used to clear any of the interrupt status bits in USBIS1 or USBIS2. Using these instructions could cause pending interrupts to be cleared without the firmware’s knowledge.

1.2.18.4.6 USB Interrupt Enable Registers 1 and 2

The USB Interrupt Enable Registers 1 and 2, shown in Figure 1.36 and Figure 1.37, are used to enable the corresponding interrupt status conditions that can generate a USB Function Interrupt. When the bit to a corresponding interrupt condition is “0”, that condition does not generate a USB function interrupt. When the bit is a “1”, that condition can generate a USB function interrupt. At reset, all USB function interrupt status conditions are enabled.

USB Interrupt Enable Register 1

b7	b6	b5	b4	b3	b2	b1	b0
						1	

Symbol: USBIE1 — Address: 030416 — When reset: FF16

Bit symbol	Bit name	Function	R	W
INTEN0	USB Endpoint 0 Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
Reserved		Must always be set to "1"	O	O
INTEN2	USB Endpoint 1 IN Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
INTEN3	USB Endpoint 1 OUT Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
INTEN4	USB Endpoint 2 IN Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
INTEN5	USB Endpoint 2 OUT Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
INTEN6	USB Endpoint 3 IN Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
INTEN7	USB Endpoint 3 OUT Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O

Figure 1.36: USB Interrupt Enable Register 1

USB Interrupt Enable Register 2

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

Symbol: USBIE2 — Address: 030516 — When reset: 3316

Bit symbol	Bit name	Function	R	W
INTEN8	USB Endpoint 4 IN Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
INTEN9	USB Endpoint 4 OUT Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
Reserved		Must always be set to "0"	O	O
INTEN12	USB Overrun/Underrun Interrupt Enable Bit	0 : Interrupt disabled 1 : Interrupt enabled	O	O
Reserved		Must always be set to "1"	O	O
Reserved		Must always be set to "0"	O	O

Figure 1.37: USB Interrupt Enable Register 2

1.2.18.4.7 USB Frame Number Registers

The USB Frame Number Low Register, shown in Figure 1.38, contains the lower 8 bits of the 11-bit frame number received from the host. The USB Frame Number High Register, shown in Figure 1.39 contains the upper 3 bits of the 11-bit frame number received from the host.

USB Frame Number Low Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: USBSOFL | Address: 030616 | When reset: 0016

Bit symbol	Bit name	Function	R	W
FN0 to FN7	Lower 8 bits of the 11-bit frame number issued with a SOF token		O	X

Figure 1.38: USB Frame Number Low Register

USB Frame Number High Register

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

Symbol: USBSOFH | Address: 030716 | When reset: 0016

Bit symbol	Bit name	Function	R	W
FN8	Upper 3 bits of the 11-bit frame number issued with a SOF token		O	X
FN9				
FN10				
Reserved		Must always be set to "0"	O	X

Figure 1.39: USB Frame Number High Register

1.2.18.4.8 USB ISO Control Register

The USB ISO Control Register, shown in Figure 1.40, contains two global bits, ISO_UPD and AUTO_FL for controlling endpoints 1-4 isochronous data transfer.

When 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 and endpoint number) if the endpoint’s IN_PKT_RDY is set.

When ISO_UPD = “1” and the ISO/TOGGLE_INIT bit of the corresponding endpoint’s IN CSR is set, the internal ‘ready to transmit’ signal to the transmit control logic is not activated when the endpoint’s IN_PKT_RDY is set. Instead, it is activated when the next SOF is received, 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 1-4 and works with isochronous pipes only.

When AUTO_FL = “1”, ISO_UPD = “1”, a particular IN endpoint’s ISO/TOGGLE_INIT bit is set, and the IN endpoint’s IN_PKT_RDY = “1”, the USB FCU detects a SOF packet and 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. Because double buffering is a requirement for ISO transfer, MAXP must be set to less than or equal to 1/2 of the FIFO size.

USB ISO Control Register

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

Symbol: USBISOC | Address: 030816 | When reset: 0016

Bit symbol	Bit name	Function	R	W
Reserved		Must always be set to "0"	O	O
AUTO_FL	AUTO_FLUSH Bit	0 : Hardware auto FIFO flush diabled 1 : Hardware auto FIFO flush enabled	O	O
ISO_UPD	ISO_UPDATE Bit	0 : ISO_UPDATE disabled 1 : ISO_UPDATE enabled	O	O

Figure 1.40: USB ISO Control Register

1.2.18.4.9 USB DMAx Request Registers

The USB DMAx Request Registers, shown in Figure 1.41 and Figure 1.42, are used to select which USB Endpoint x FIFO read/write requests are selected as the DMAC channel 0 or channel 1 request source. The USB DMA0 (DMA1) Request Register should have only one bit set at any given time. When multiple bits are set, no request is selected.

USB DMA0 Request Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: USBSAR0 | Address: 030916 | When reset: 0016

Bit symbol	Bit name	Function	R	W
DMA0R0	Endpoint 1 IN FIFO write request selection bit	0 : Not selected 1 : Selected	0	0
DMA0R1	Endpoint 2 IN FIFO write request selection bit		0	0
DMA0R2	Endpoint 3 IN FIFO write request selection bit		0	0
DMA0R3	Endpoint 4 IN FIFO write request selection bit		0	0
DMA0R4	Endpoint 1 OUT FIFO read request selection bit		0	0
DMA0R5	Endpoint 2 OUT FIFO read request selection bit		0	0
DMA0R6	Endpoint 3 OUT FIFO read request selection bit		0	0
DMA0R7	Endpoint 4 OUT FIFO read request selection bit		0	0

Figure 1.41: USB DMA0 Request Register

USB DMA1 Request Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: USBSAR1 | Address: 030A16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
DMA1R0	Endpoint 1 IN FIFO write request selection bit	0 : Not selected 1 : Selected	0	0
DMA1R1	Endpoint 2 IN FIFO write request selection bit		0	0
DMA1R2	Endpoint 3 IN FIFO write request selection bit		0	0
DMA1R3	Endpoint 4 IN FIFO write request selection bit		0	0
DMA1R4	Endpoint 1 OUT FIFO read request selection bit		0	0
DMA1R5	Endpoint 2 OUT FIFO read request selection bit		0	0
DMA1R6	Endpoint 3 OUT FIFO read request selection bit		0	0
DMA1R7	Endpoint 4 OUT FIFO read request selection bit		0	0

Figure 1.42: USB DMA1 Request Register

1.2.18.4.10 USB Endpoint Enable Register

The USB Endpoint Enable Register, shown in Figure 1.43, is used to enable/disable an individual endpoint. Endpoint 0 is always enabled and cannot be disabled by firmware. All endpoints are enabled after reset.

USB Endpoint Enable Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: USBEPEN | Address: 030B16 | When reset: FF16

Bit symbol	Bit name	Function	R	W
EP1_OUT	Endpoint 1OUT FIFO Enable bit	0 : Disabled 1 : Enabled	0	0
EP1_IN	Endpoint 1 IN FIFO Enable bit		0	0
EP2_OUT	Endpoint 2OUT FIFO Enable bit		0	0
EP2_IN	Endpoint 2 IN FIFO Enable bit		0	0
EP3_OUT	Endpoint 3 OUT FIFO Enable bit		0	0
EP3_IN	Endpoint 3 IN FIFO Enable bit		0	0
EP4_OUT	Endpoint 4 OUT FIFO Enable bit		0	0
EP4_IN	Endpoint 4 IN FIFO Enable bit		0	0

Figure 1.43: USB Endpoint Enable Register

1.2.18.4.11 USB Endpoint 0 Control and Status Register

The Endpoint 0 CSR (Control and Status Register), shown in Figure 1.44 contains the control and status information of Endpoint 0.

- **EP0CSR0 (OUT_PKT_RDY):**

The USB FCU sets this bit to a “1” after it receives a valid SETUP/OUT token from the host. The CPU clears this bit after unloading the packet from the FIFO by writing a “1” to EP0CSR6. The CPU should not clear the OUT_PKT_RDY bit before it finishes decoding the host request. When EP0CSR2 (SEND_STALL) needs to be set (because the CPU decodes an invalid or unsupported request) a “1” should be written to EP0CSR6 and EP0CSR2 at the same time using the same instruction.

- **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 when it decodes an invalid or unsupported standard device request from the host. When the OUT-PKT_RDY bit is a “1” at the time the CPU wants to set the SEND_STALL bit to a “1”, the CPU must also set SERVICED_OUT_PKT_RDY to a “1” to clear the OUT-PKT_RDY at the same time as setting the SEND_STALL bit. 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 it after it receives a new SETUP packet. It is up to the firmware to decide what SETUP packet should lead the clearing of the SEND_STALL bit.

- **EP0CSR3 (DATA_END):**

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. The CPU sets this bit at the same time as it sets the last IN_PKT_RDY bit or sets the last SERVICED_OUT_PKT_RDY bit.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 requests or sends more data, the USB FCU returns a STALL handshake and terminates the current control transfer.

- **EP0CSR4 (FORCE_STALL):**

The USB FCU sets this bit to a “1” to report an error status when one of the following occur:

- Host sends an IN token in the absence of a SETUP stage
- Host sends a bad data toggle in the STATUS stage, (i.e. DATA0 is used)
- Host sends a bad data toggle in the SETUP stage, (i.e. DATA1 is used)
- Host request more data than specified in the SETUP state,
- (i.e. IN token comes after DATA_END bit is set)
- Host sends more data than specified in the SETUP state,
- (i.e. OUT token comes after DATA_END bit is set)
- Host sends larger data packet than MAXP size

All of the conditions stated (except bad data toggle in the SETUP stage) cause the device to send a STALL handshake for the current IN/OUT transaction. For the bad data toggle in the SETUP state, the device sends ACK for the SETUP stage and then sends STALL for the next IN/OUT transaction. A STALL handshake caused by the above listed conditions lasts for one transaction and terminates the ongoing control transfer. Any packet after the STALL handshake will be seen as the beginning of a new control transfer.

The CPU writes a “0” to clear the FORCE_STALL status 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 (status phase starts before DATA_END bit is set) or a control transfer has ended before a new SETUP has arrived and before successfully completing the status phase. The CPU clears this bit by writing a “1” to IN0CSR7. Once the CPU detects the SETUP_END flag as set, it should stop accessing the FIFO to service the previous setup transaction. If the SETUP_END is caused by the reception of the SETUP packet prior to the end of the current control transfer, the OUT_PKT_RDY bit is set once the reception of the SETUP packet has completed (without errors). After the OUT_PKT_RDY bit is set, the new SETUP packet

data will be in the FIFO. For this case, because the SETUP_END bit is set near the beginning of the packet when the SETUP PID is encountered and the OUT_PKT_RDY bit is set at the end of the packet, the value read from EP0IN_CSR in the USB functional interrupt routine may only show that the SETUP_END flag as “1” instead of both the SETUP_END and OUT_PKT_RDY bits.

- **EP0CSR6 and EP0CSR7:**

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

USB Endpoint 0 Control and Status Register (Note 5)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: EP0CS | Address: 031116 | When reset: 0016

Bit symbol	Bit name	Function	R	W
EP0CSR0	OUT_PKT_RDY Flag	0 : Not ready 1 : Ready	0	0 Note 1
EP0CSR1	IN_PKT_RDY Bit	0 : Not ready 1 : Ready	0	0 Note 2
EP0CSR2	SEND_STALL Bit	0 : No action 1 : Stall Endpoint 0 by CPU	0	0
EP0CSR3	DATA_END Bit	0 : No action 1 : Last packet transferred from/to FIFO	0	0 Note 2
EP0CSR4	FORCE_STALL Flag	0 : No action 1 : Stall Endpoint 0 by USB FCU	0	0 Note 3
EP0CSR5	SETUP_END Flag	0 : No action 1 : Control transfer ended before specific length of data transferred during data phase	0	0 Note 1
EP0CSR6	SERVICED_OUT_PKY_RDY Bit	0 : No change 1 : Clear the OUT_PKT_RDY bit (EP0CSR0)	0	0 Note 4
EP0CSR7	SERVICED_SETUP_END Bit	0 : No change 1 : Clear the STUP-END bit (EP0CSR5)	0	0 Note 4

Note 1: Read only
Note 2: Write "1" only or Read
Note 3: Write "0" only or Read
Note 4: Write only - Read "0"
Note 5: Refer to Section 1.5.5 "Programming Notes" for this register

Figure 1.44: USB Endpoint 0 CSR

1.2.18.4.12 USB Endpoint 0 MAXP Register

The USB Endpoint 0 MAXP Register, shown in Figure 1.45, indicates the maximum packet size (MAXP) of Endpoint 0 IN/OUT packet. The default value for Endpoint 0 MAXP is 8 bytes.

USB Endpoint 0 MAXP Register

b7 b6 b5 b4 b3 b2 b1 b0
0 0

Symbol: EP0MP | Address: 031316 | When reset: 0816

Bit symbol	Bit name	Function	R	W
EP0MXP0 to EP0MXP5	Maximum packet size (MAXP) of Endpoint 0 IN/OUT packet		O	O
Reserved		Must always be set to "0"	O	O

Figure 1.45: USB Endpoint 0 MAXP

1.2.18.4.13 USB Endpoint 0 OUT Write Count Register

The USB Endpoint 0 OUT Write Count (WRT CNT) Register, shown in Figure 1.46, 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.

USB Endpoint 0 OUT Write Count Register

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0					

Symbol: EP0WC
Address: 031516
When reset: 0016

Bit symbol	Bit name	Function	R	W
W_CNT0 to W_CNT4	Receive byte count		○	X
Reserved		Must always be set to "0"	○	X

Figure 1.46: USB Endpoint 0 OUT WRT CNT

1.2.18.4.14 USB Endpoint x IN Control and Status Register

The USB Endpoint x IN CSR (Control and Status Register), shown in Figure 1.47, 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 for IN FIFO status when in read operation (see “IN (Transmit) FIFO” operation for details).

The CPU writes a “1” to the INxCSR0 bit to inform the USB FCU that a packet of data is written to the FIFO. The USB FCU updates the pointers up on this bit set. The USB FCU also updates the pointers upon a packet of data successfully sent to the host. When the pointer updates are completed, the IN FIFO status is shown on INxCSR0 and INxCSR5 bits for the CPU to read. The CPU must allow at least one wait state between writing and reading these bits for proper FIFO status.

- **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/TOGGLE_INIT):**

When the endpoint is used for isochronous data transfer, the CPU sets this bit to a “1” for the entire duration of the isochronous transfer. With the ISO bit set to a “1”, the device uses DATA0 as the pid for all packets sent back to the host.

When the endpoint is required to initialize the data toggle, this set/reset of the TOGGLE_INIT bit method assumes that there is no activity IN transaction to the respective endpoint on the bus at the time the initialization process is ongoing. Set/reset of the TOGGLE_INIT bit is performed only when an endpoint experiences a configuration event.

- **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 when the corresponding endpoint is to be used to communicate rate feedback information (see Chapter. IN (Transmit) FIFOs for details).

- **INxCSR5 (TX_FIFO_NOT_EPT):**

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

- INxCSR6 (FLUSH):

The CPU writes a “1” to this bit to flush the IN FIFO. When there is one packet in the IN FIFO, a flush causes the IN FIFO to be empty. When 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):**

When 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).

USB Endpoint x IN Control and Status Register (Note 5)

Symbol: EPiICS (i= 1-4) | Address: 031916, 032116, 032916, 033116 | When reset: 0016

Bit symbol	Bit name	Function	R	W
INxCSR0	IN_PKT_RDY Bit	0 : Not ready 1 : Ready	0	0 Note 1
INxCSR1	UNDER_RUN Flag	0 : No FIFO underrun 1 : FIFO underrun has occured	0	0 Note 2
INxCSR2	SEND_STALL Bit	0 : No action 1 : Stall IN Endpoint x by CPU	0	0
INxCSR3	ISO Bit	0 : Select non-isochronous transfer 1 : Select isochronous transfer	0	0
INxCSR4	INTPT	0 : Select non-rate feedback interrupt transfer 1 : Select rate feedback interrupt transfer	0	0
INxCSR5	TX_NOT_EPT Flag	0 : Transmit FIFO is empty 1 : Transmit FIFO is not empty	0	0 Note 3
INxCSR6	FLUSH Bit	0 : No action 1 : Flush the FIFO	0	0 Note 4
INxCSR7	AUTO_SET Bit	0 : AUTO-SET disabled 1 : AUTO-SET enabled	0	0

Note 1: Write "1" only or read
Note 2: Write "0" only or read
Note 3: Read only
Note 4: Write only - Read "0"
Note 5: Refer to section 1.5.5 "Programming Notes" for this register

Figure 1.47: USB Endpoint x IN CSR

1.2.18.4.15 USB Endpoint x OUT Control and Status Register

The USB Endpoint x OUT CSR (Control and Status Register), shown in Figure 1.48 contains control and status information of the respective OUT Endpoint 1-4.

- **OUTxCSR0 (OUT_PKT_RDY):**

The OUTxCSR0 bit for the OUT FIFO status (see “OUT (Receive) FIFOs” for details).

The USB FCU sets this bit to a “1” and updates the FIFO pointers after a data packet has been successfully received from the host. The CPU writes a “0” to this bit to inform the USB FCU that a data packet has been unloaded. The USB FCU updates the FIFO pointers when this occurs. The CPU must allow at least one clock cycle between writing and reading bit OUTxCSR0.

- **OUTXxCSR1 (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 when two data packets are already present 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 OUTXCSR1.

- **OUTxCSR2 (SEND_STALL):**

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

- **OUTxCSR3 (ISO/TOGGLE_INIT):**

When the endpoint is used for isochronous data transfer, the CPU sets this bit to a “1” for the entire duration of the isochronous transfer. With the ISO/TOGGLE_INIT bit set to a “1”, the device accepts either DATA0 or DATA1 for the PID sent by the host.

When endpoint is required to initialize the data toggle sequence bit (i.e. reset to DATA0 for the next data packet), the CPU sets this bit to a “1” and then resets it to a “0” to initialize the respective endpoint’s data toggle.

Successful initialization of the data toggle sequence bit can only be guaranteed if no active OUT transaction to the respective endpoint is ongoing when the initialization process is taking place. Set/reset of the ISO/TOGGLE_INIT bit should only be performed when an endpoint experiences a configuration event.

- **OUTxCSR4 (FORCE_STALL):**

The USB FCU sets this bit to a “1” when 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 that a CRC error or a bit stuffing error was 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. When there is one packet in the OUT FIFO, a flush causes the OUT FIFO to be empty. When 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):**

When 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” for details).

USB Endpoint x OUT Control and Status Register (Note 3)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: EPiOCS (i = 1-4) — Address: 031A16, 032216, 032A16, 033216 — When reset: 0016

Bit symbol	Bit name	Function	R	W
OUTxCSR0	OUT_PKT_RDY Flag	0 : Not ready 1 : Ready	0	0 Note 1
OUTxCSR1	OVER_RUN Flag	0 : No FIFO overrun 1 : FIFO overrun occured	0	0 Note 1
OUTxCSR2	SEND_STALL Bit	0 : No action 1 : Stall OUT Endpoint x by CPU	0	0
OUTxCSR3	ISO Bit	0 : Select non-isochronous transfer 1 : Select isochronous transfer	0	0
OUTxCSR4	FORCE-STALL Flag	0 : No action 1 : Stall Endpoint X by the USB FCU	0	0 Note 1
OUTxCSR5	DATA-ERR Flag	0 : No error 1 : CRC or bit stuffing error received in ISO packet	0	0 Note 1
OUTxCSR6	FLUSH Bit	0 : No action 1 : Flush the FIFO	0	0 Note 2
OUTxCSR7	AUTO_CLR Bit	0 : AUTO-CLR disabled 1 : AUTO-CLR enabled	0	0

Note 1: Write "0" only or read
Note 2: Write only - Read "0"
Note 3: Refer to section 1.5.5 "Programming Notes" for this register

Figure 1.48: USB Endpoint x OUT CSR

1.2.18.4.16 USB Endpoint x IN MAXP Register

The USB Endpoint x IN MAXP Register, shown in Figure 1.49, indicates the maximum packet size (MAXP) of an Endpoint x IN packet. The default values for Endpoints 1-4 are 0 bytes. The setting of this register also affects the configuration of single/dual packet operation. When MAXP > 1/2 of the FIFO size, single packet mode is set. When MAXP <= 1/2 of the FIFO size, dual packet mode is set.

USB Endpoint x IN MAXP Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: EPiIMP (i = 1-4) — Address: 031B16, 032316, 032B16, 033316 — When reset: 0016

Bit symbol	Bit name	Function	R	W
IMAXP0 to IMAXP7	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 these bits).	O	O

Figure 1.49: USB Endpoint x IN MAXP

1.2.18.4.17 USB Endpoint x OUT MAXP Register

The USB Endpoint x OUT MAXP Register, shown in Figure 1.50, indicates the maximum packet size (MAXP) of an Endpoint x OUT packet. The default values for endpoints 1-4 are 0 bytes. The setting of this register also affects the configuration of single/dual packet operation. When MAXP > 1/2 of the FIFO size, single packet is set. When MAXP <= 1/2 of the FIFO size, dual packet mode is set.

USB Endpoint x OUT MAXP Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: EPiOMP (i = 1-4)
Address: $031C_{16}$, 0324_{16}, $032C_{16}$, 0334_{16}
When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
OMAXP0 to OMAXP7	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 these bits).	O	O

Figure 1.50: USB Endpoint x OUT MAXP

1.2.18.4.18 USB Endpoint x OUT Write Count Register

The USB Endpoint x OUT Write Count (WRT CNT) Register, shown in Figure 1.51, 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.

USB Endpoint x OUT Write Count Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: EPiWC (i = 1-4)
Address: $031D_{16}$, 0325_{16}, $032D_{16}$, 0335_{16}
When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
W_CNT0 to W_CNT7	Receive Byte Count		O	X

Figure 1.51: USB Endpoint x OUT WRT CNT

1.2.18.4.19 USB Endpoint x FIFO Register

The USB Endpoint x FIFO Register, shown in Figure 1.52 is the USB IN (transmit) and OUT (receive) FIFO data register. The CPU writes data to this register for the corresponding Endpoint IN FIFO and reads data from this register for the corresponding Endpoint OUT FIFO.

USB Endpoint x FIFO Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: EPi (i = 0-4)
Address: 0338_{16}, 0339_{16}, $033A_{16}$, $033B_{16}$, $033C_{16}$
When reset: Indeterminate

Bit symbol	Bit name	Function	R	W
DATA_0 to DATA_7	Endpoint x IN/OUT FIFO		O	O

Figure 1.52: USB Endpoint x FIFO Register

1.2.19 DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU.Table 1.15 shows the DMAC specifications. Figure 1.53 shows the block diagram of the DMAC. Figure 1.54, Figure 1.55 and Figure 1.56 show the registers used by the DMAC.

Table 1.15: 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 sources	Falling edge of INT0 or INT1 (INT0 can be selected by DMA0, INT1 by DMA1) Timer A0 to timer A4 Timer B0 to timer B1 UART0 transmission and reception UART1 transmission and reception UART2 transmission and reception A-D conversion complete USB function 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 modes	•Single transfer mode The DMA enable bit is cleared and transfer ends when an underflow occurs in the transfer counter. •Repeat transfer mode When an underflow occurs in the transfer counter, the value in the transfer counter reload register is loaded 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 mode Transfer starts when the DMA is requested after “1” is written to the DMA enable bit •Repeat transfer mode 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 loaded into the forward address pointer. The value in the transfer counter reload register is loaded 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 transfers are not affected by the interrupt enable flag (I-FLAG) of any interrupt or by the interrupt priority level.

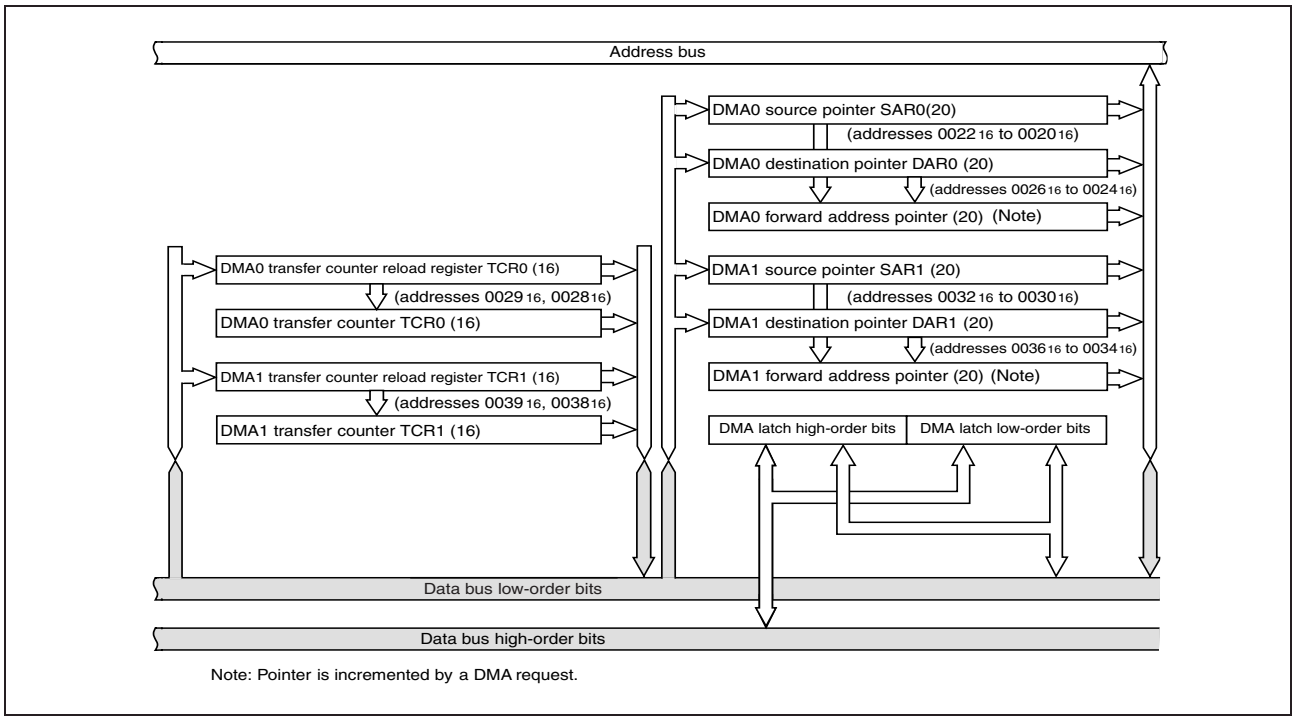


Figure 1.53: Block diagram of DMAC

DMAi request cause select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: DMiSL (i=0,1)

Address: 03B816, 03BA16

When reset: 0016

Bit symbol	Bit name	Function (Note)	R	W
DSEL0	DMA request cause select bits	b3 b2 b1 b0 0 0 0 0 : Falling edge of INT0/INT1pin (Note 1) 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/UART receive (Note 2)	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. Write "0" when writing to these bits. If read, the value 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 03B816 is for INT0, 3BA16 is for INT1.
Note 2: Address 03B816 is for UART1 transmit, 03BA16 is for UART1 receive.
Note 3: Address 03B816 is for USB0, 03BA16 is for USB1.

Figure 1.54: DMAC register (1)

DMAi control register

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

Symbol: DMiCON(i=0,1)

Address: 002C16, 003C16

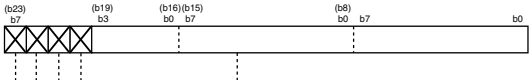
When reset: 00000X002

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. Write "0" when writing to these bits. If read, the value 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 1.55: DMAC register (2)

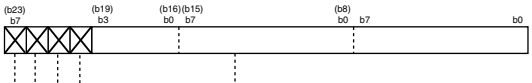
DMAi source pointer (i = 0, 1)



Symbol	Address	When reset
SAR0	0022₁₆ to 0020₁₆	Indeterminate
SAR1	0032₁₆ to 0030₁₆	Indeterminate

Function	Transfer count specification	R	W
• Source pointer Stores the source address	00000₁₆ to FFFFF₁₆	O	O
Nothing is assigned. Write "0" when writing to these bits. If read, the value is "0".		–	–

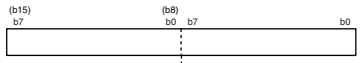
DMAi destination pointer (i = 0, 1)



Symbol	Address	When reset
DAR0	0026₁₆ to 0024₁₆	Indeterminate
DAR1	0036₁₆ to 0034₁₆	Indeterminate

Function	Transfer count specification	R	W
• Destination pointer Stores the destination address	00000₁₆ to FFFFF₁₆	O	O
Nothing is assigned. Write "0" when writing to these bits. If read, the value is "0".		–	–

DMAi transfer counter (i = 0, 1)



Symbol	Address	When reset
TCR0	0029₁₆, 0028₁₆	Indeterminate
TCR1	0039₁₆, 0038₁₆	Indeterminate

Function	Transfer count specification	R	W
• Transfer counter Set a value one less than the transfer count	0000₁₆ to FFFF₁₆	O	O

Figure 1.56: DMAC register (3)

• 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.

• 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.

• Calculations

Any combination of even or odd transfer read and write addresses is possible. Table 1.16 show the number of DMAC transfer cycles. Table 1.17 shows the corresponding coefficient values. Figure 1.57 shows an example of the transfer cycle for a source read.

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

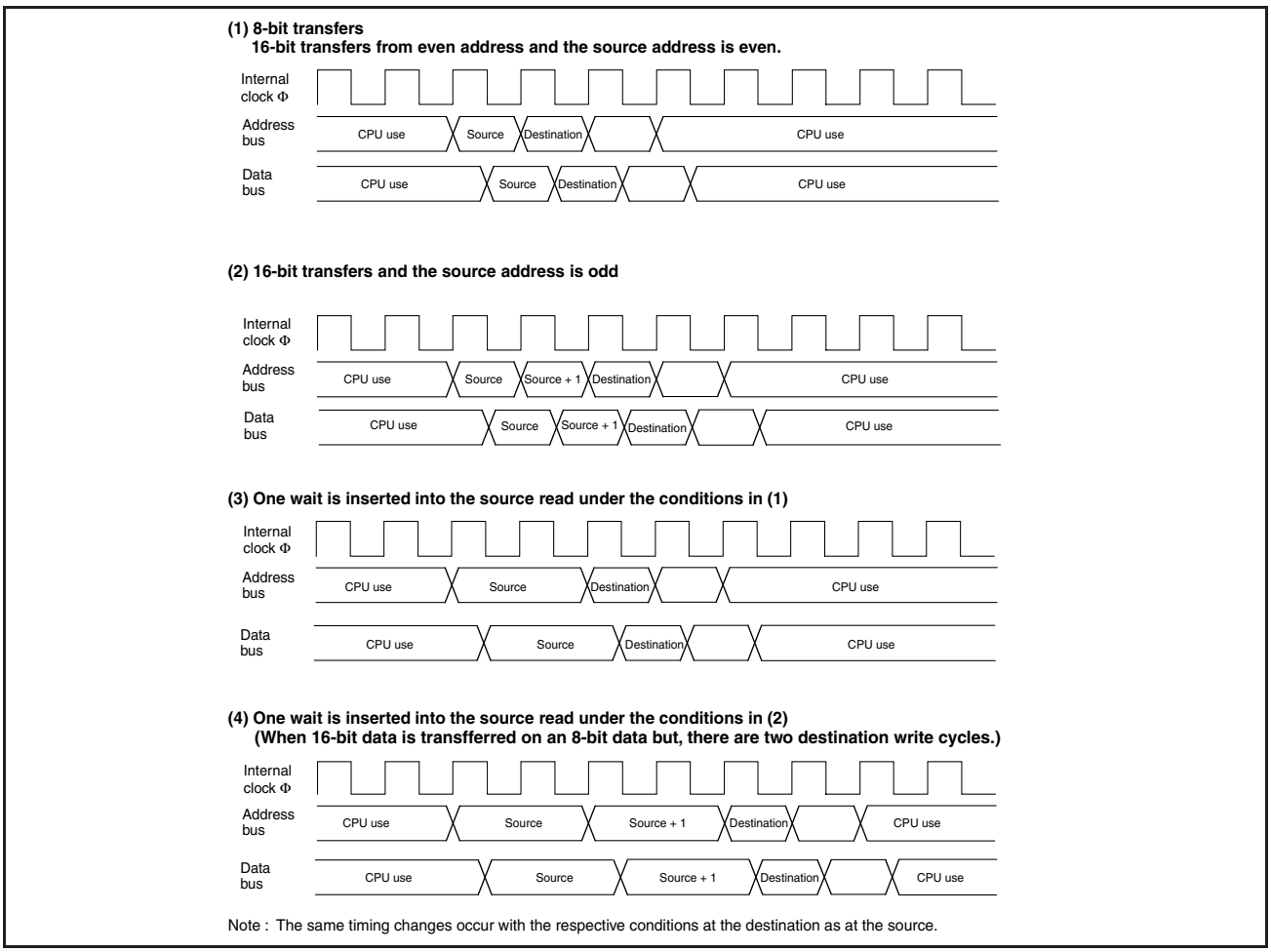
Number of transfer cycles per transfer unit = Number of read cycles x j + Number of write cycles x k

Table 1.16: 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 1.17: 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 1.57: Example of the transfer cycle for a source read

1.2.20 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 1.58 shows the block diagram of Timers A and B.

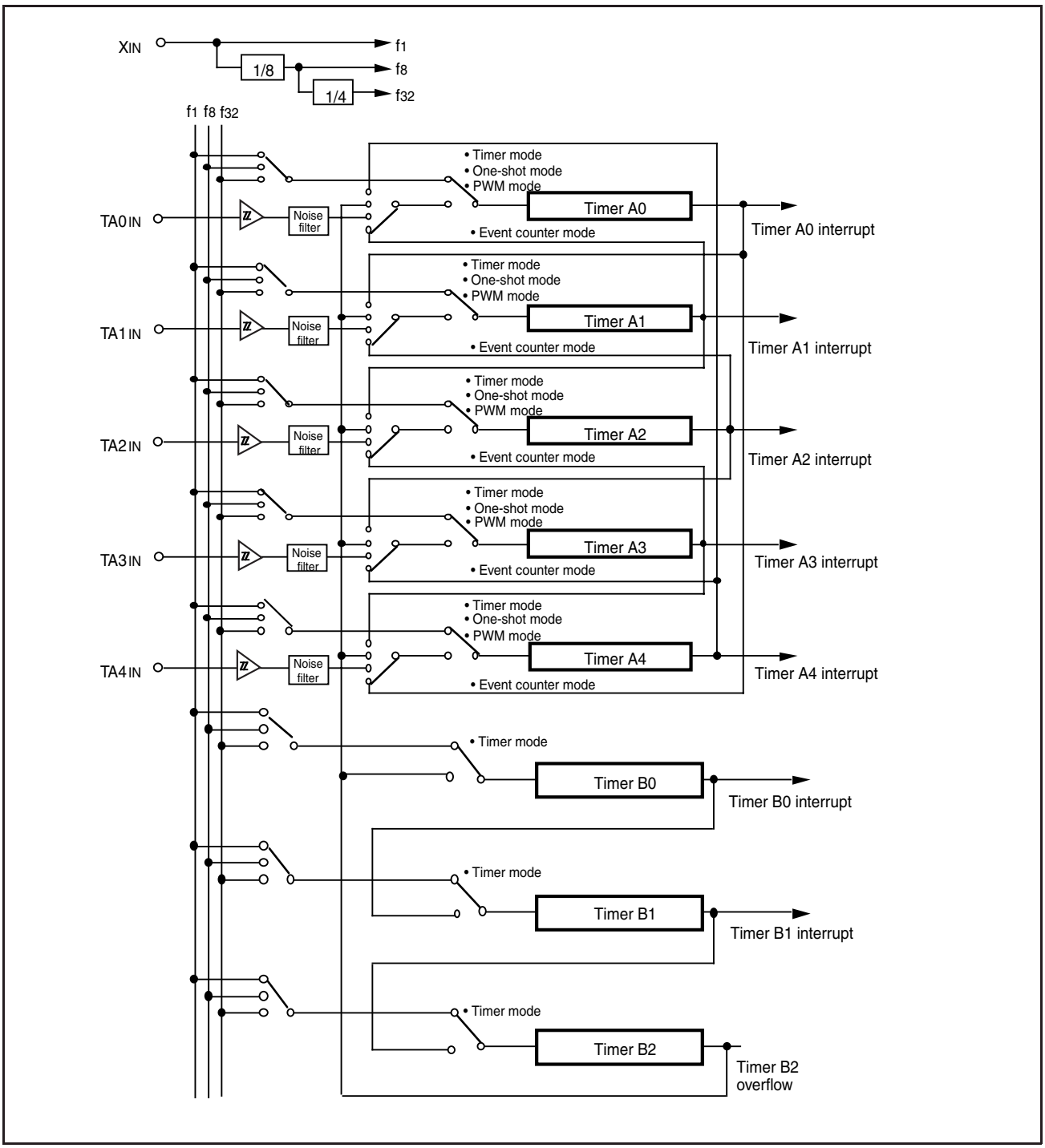


Figure 1.58: Timer A and Timer B block diagram

1.2.21 Timer A

Figure 1.59, Figure 1.60,Figure 1.61, and Figure 1.62 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.

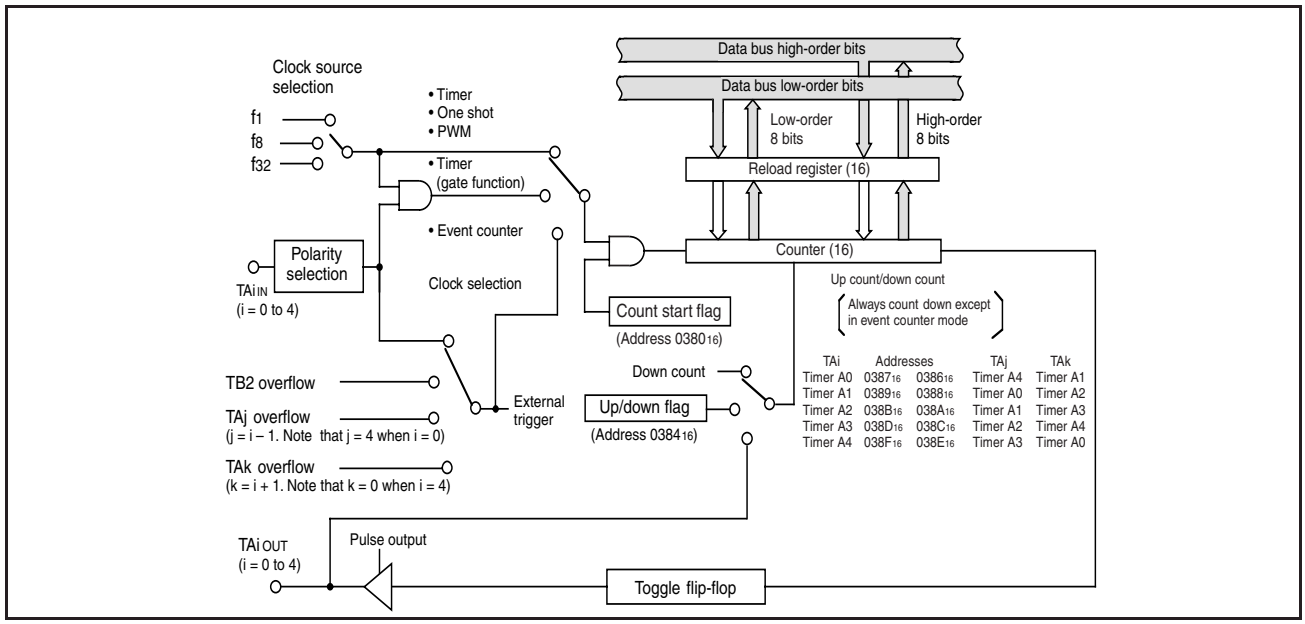


Figure 1.59: Block diagram of Timer A

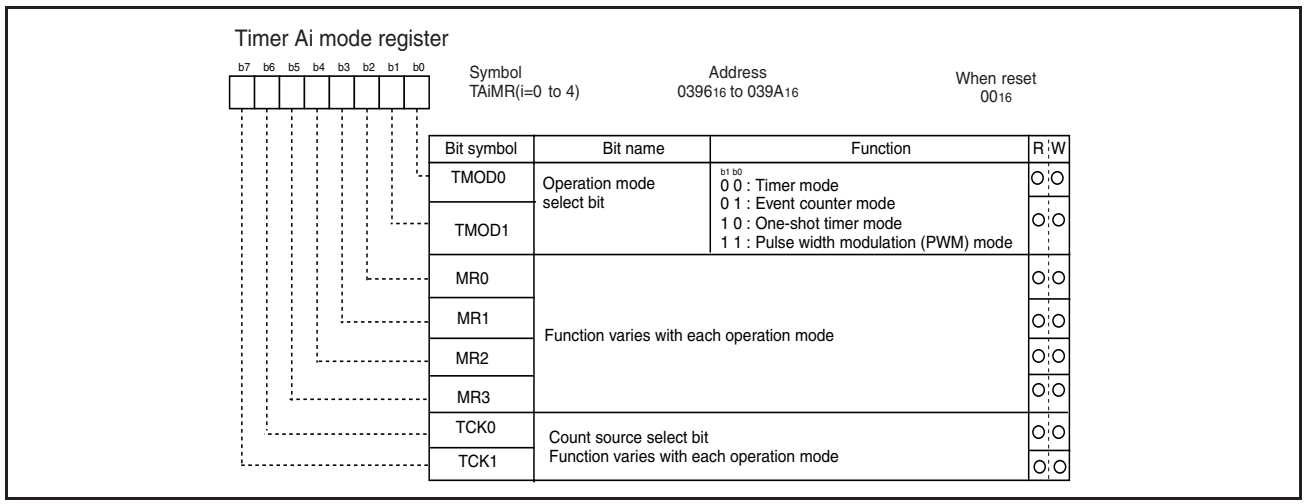
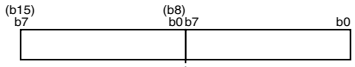


Figure 1.60: Timer A related Registers (1)

Timer Ai register (Note)

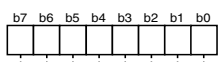


Symbol	Address	When reset
TA0	038716,038616	Indeterminate
TA1	038916,038816	Indeterminate
TA2	038B16,038A16	Indeterminate
TA3	038D16,038C16	Indeterminate
TA4	038F16,038E16	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	000016 to FFFF16	○	○
• Event counter mode Counts pulses from an external source or timer overflow	000016 to FFFF16	○	○
• One-shot timer mode Counts a one shot width	000016 to FFFF16	×	○
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	000016 to FFFE16	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	0016 to FE16 (Both high-order and low-order addresses)	×	○

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

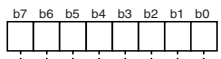
Count start flag



Symbol	Address	When reset
TABSR	038016	0016

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

Up/down flag



Symbol	Address	When reset
UDF	038416	0016

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 1.61: Timer A-related registers (2)

One-shot start flag

Symbol	Address	When reset
ONSF	038216	00X000002

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	O	O
TA1OS	Timer A1 one-shot start flag		O	O
TA2OS	Timer A2 one-shot start flag		O	O
TA3OS	Timer A3 one-shot start flag		O	O
TA4OS	Timer A4 one-shot start flag		O	O
Nothing is assigned. Write "0" when writing to this bit. When read, the value 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	O	O
TA0TGH			O	O

Note: Set the corresponding port direction register to “0”.

Trigger select register

Symbol	Address	When reset
TRGSR	038316	0016

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

Note: Set the corresponding port direction register to “0”.

Figure 1.62: Timer A-related registers (3)

1.2.21.1 Timer mode

In this mode, the timer counts an internally generated count source. See Table 1.18 below. Figure 1.63 shows the timer Ai mode register in timer mode.

Table 1.18: Specifications of timer mode

Item	Specification
Count source	f1, f8, f32
Count operation	• Down count • When the timer underflows, it loads 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
TAi_{IN} pin function	Programmable I/O port or gate input
TAi_{OUT} 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 is stopped and a value is written to timer Ai register, it is written to both reload register and counter • When counting is in progress and a value is written to timer Ai register, it is written only to reload register (to be transferred to counter at the next reload time)
Select function	• Gate function Counting can be started and stopped by TAi_{IN} pin's input signal • Pulse output function Each time the timer underflows, the TAi_{OUT} pin's polarity is reversed

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TAiMR(i=0 to 4) Address: 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 (TA_{iOUT} pin is a normal port pin) 1 : Pulse is output (Note 1) (TA_{iOUT} pin is a pulse output pin)	○	○
MR1	Gate function select bit	b4 b3 0 X : Gate function not available (Note 2) (TA_{iIN} pin is a normal port pin) 1 0 : Timer counts only when TA_{iIN} pin is held "L" (Note 3) 1 1 : Timer counts only when TA_{iIN} 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 : Reserved	○	○
TCK1			○	○

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

Figure 1.63: Timer Ai mode register in timer mode

1.2.21.2 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 1.19 lists the timer specifications when counting a single-phase external signal. Figure 1.64 shows Timer Ai mode register in event counter mode, Note: Timer Ai register's value can be indeterminate when the count starts.

Table 1.19: Timer specification in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAi_{IN} 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 of software • When the timer overflows or underflows, it loads the reload register contents before continuing counting (However, this does not apply when the free-run function is selected)
Divide ratio	$1/(FFFF_{16}-n+1)$ for up count 1/(n+1) for down count n: Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows or underflows
TAi_{IN} pin function	Programmable I/O port or count source input
TAi_{OUT} pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading Timer Ai register
Write to timer	• When counting is stopped and a value is written to Timer Ai register, it is written to both reload register and counter • When counting is in progress and a value is written to Timer Ai register, it is written only to reload register (to be transferred to counter at the next reload time)
Select function	• Free-run count function (Note) When the timer overflows or underflows, the reload register's content is not reloaded. • Pulse output function Each time the timer underflows, the TAi_{OUT} pin's polarity is reversed

Note: Timer Ai register's value can be indeterminate when the count starts.

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol: TAiMR(i = 0, 1) Address: 039616, 039716 When reset: 0016

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (Note 1)	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TA iOUT pin is a normal port pin) 1 : Pulse is output (Note 2) (TA iOUT 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 : TA iOUT 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 (Note 5)	○	○
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 038216 and 038316).
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Valid only when counting an external signal.
Note 4: When an "L" signal is input to the TAiOUT pin, the downcount is activated. When "H", the upcount is activated. Set the corresponding port direction register to "0".
Note 5: This value can be indeterminate when the count starts.

Figure 1.64: Timer Ai mode register in event counter mode, single signal

Table 1.20: Timer specification in event counter mode (when processing two-phase pulse signal with Timers A2, A3, A4.)

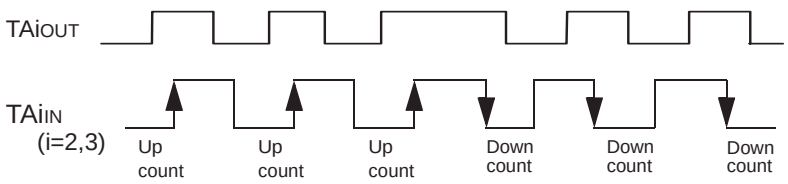
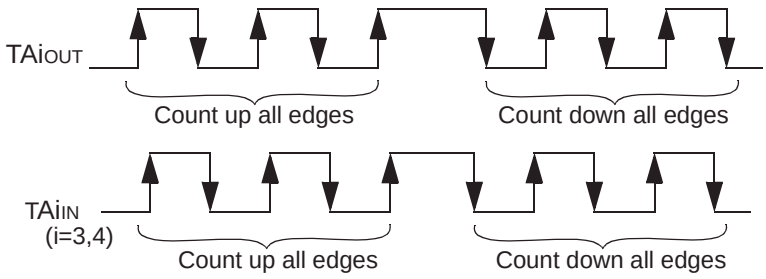
Item	Specification
Count Source	•Two-phase pulse signals input to TAi_{IN} or TAi_{OUT} pin
Count operation	•Up count or down count can be selected by two-phase pulse signal •When the timer overflows or underflows, the reload register content is loaded and the timer starts over again (Note 1)
Divide ratio	1/ ($FFFF_{16}$ - n + 1) for up count 1/ (n+1) for down count n: Set value
Count start condition	Count start flag is set (=1)
Count stop condition	Count start flag is reset (=0)
Interrupt request generation timing	Timer 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 is stopped and a value is written to timer A2, A3, or A4 register, it is written to both the reload register and counter •When counting is in progress and a value is written to timer A2, A3, or A4 register, it is written to only reload register to be transferred to counter at the 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"  •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 down rising and falling edges on the TAiOUT and TAiIN pins. 

Figure 1.65 shows Timer Ai mode register in event counter mode when processing two-phase signal.

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

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol: TAiMR(i = 2 to 4)
Address: 039816 to 039A16
When reset: 0016

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 (TAi OUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAi OUT 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 : TA iOUT 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 (Note 6)	○	○
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 038416) is set to "1". Also, always be sure to set the event/trigger select bit (addresses 038216 and 038316) to "00".
Note 6: This value can be indeterminate when the count starts.

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

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

Symbol: TAiMR(i = 2 to 4)
Address: 039816 to 039A16
When reset: 0016

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 (Note 3)	○	○
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 038416) is set to "1". Also, always be sure to set the event/trigger select bit (addresses 038216 and 038316) to "00".
Note 3: This value can be indeterminate when the count starts.

Figure 1.65: Timer Ai mode register in event counter mode, two-phase signal

1.2.21.3 One-shot timer mode

In this mode, the timer operates only once (See Table 1.21). When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.66 shows the Timer Ai mode register in one-shot mode.

Table 1.21: Timer specifications in one-shot timer mode

Item	Specification
Count source	f1, f8, f32
Count operation	•The timer counts down •When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n: Set value
Count start condition	• An external trigger is input • The selected timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TAi_{IN} pin function	Programmable I/O port or trigger input
TAi_{OUT} pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	•When counting is stopped and a value is written to timer Ai register, it is written to both reload register and counter •When counting is in progress and a value is written to timer Ai register, it is written to the reload register to be transferred to counter at next load time

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

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

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	○	○
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	External trigger select bit (Note 2)	0 : Falling edge of TAi IN pin's input signal (Note3) 1 : Rising edge of TAi IN pin's input signal (Note3)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select register	○	○
MR3	0 (Must always be fixed to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : Reserved	○	○
TCK1			○	○

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

Figure 1.66: Timer Ai mode register in one-shot mode

1.2.21.4 Pulse-width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession (See Table 1.22). In this mode, the counter functions as either a 16-bit pulse-width modulator or an 8-bit pulse-width modulator. Figure 1.67 shows an example of how a 16-bit pulse-width modulator operates. Figure 1.68 shows the Timer Ai mode register in pulse-width modulation mode. Figure 1.69 shows the example of how an 8-bit pulse width modulator operates.

Table 1.22: 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 / f_i n: Set value •Cycle time (2^{16}-1) / f_i fixed
8-bit PWM	•High level width n (m+1) /f_i n: values set to timer Ai register's high-order address •Cycle time (2^8-1) (m+1) /f_i 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"
TAi_{IN} pin function	Programmable I/O port or trigger input
TAi_{OUT} pin function	Pulse output
Read from timer	When Timer Ai register is read, it indicates an indeterminate value
Write to timer	•When counting is stopped and a value is written to Timer Ai register, it is written to both reload register and the counter •When counting in progress and a value is written to Timer A register, it is written to only reload register to be transferred to the counter at next reload timer.

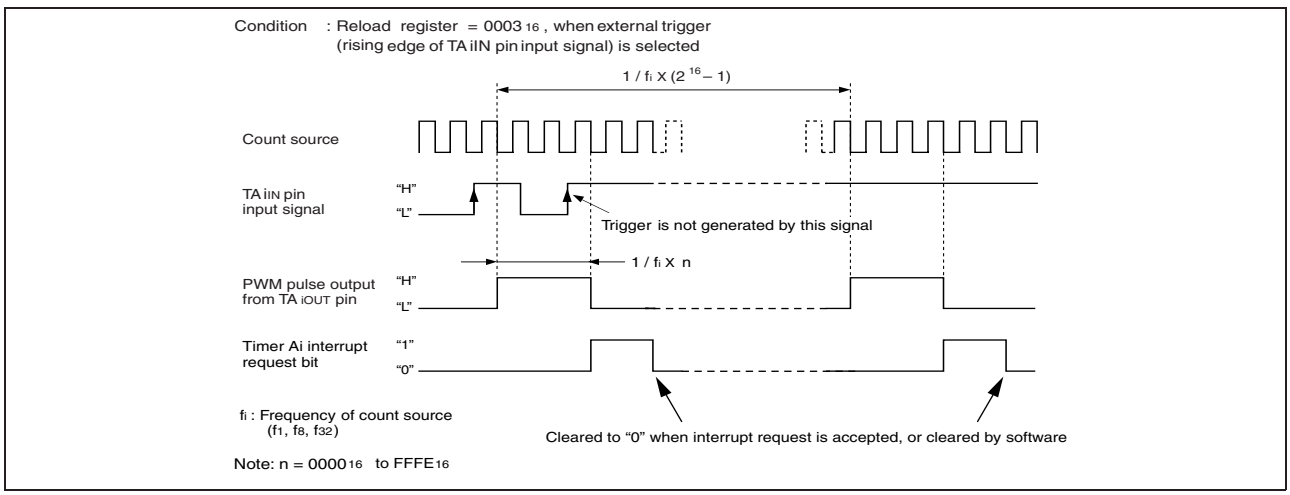


Figure 1.67: Example of how a 16-bit pulse-width modulator operates

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Symbol: TAiMR(i=0 to 4)

Address: 039616 to 039A16

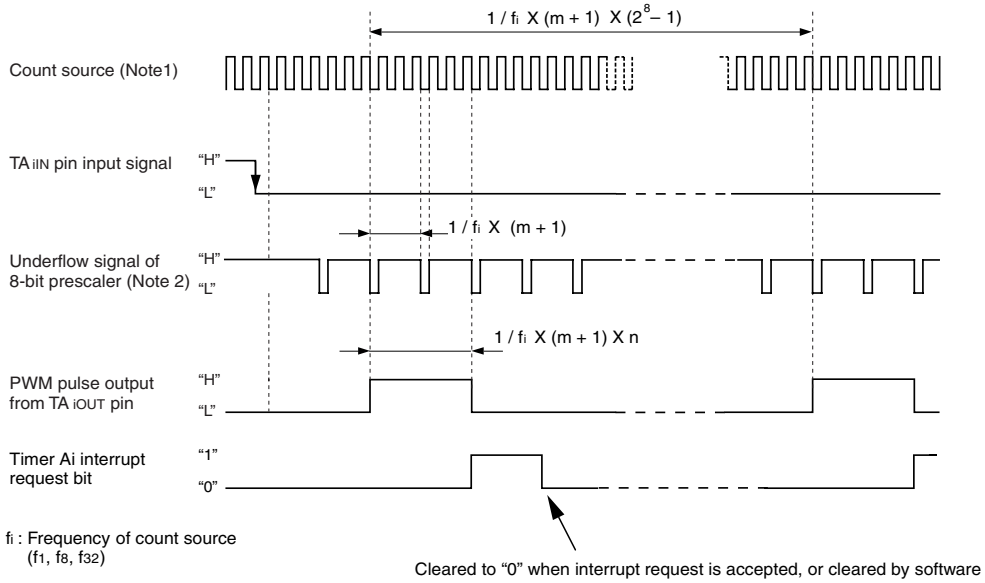
When reset: 0016

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	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 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 : f1 0 1 : f8 1 0 : f32 1 1 : Reserved	○	○
TCK1			○	○

Note 1: Valid only when the TAi IN pin is selected by the event/trigger select bit. (addresses 038216 and 038316). If timer overflow is selected, this bit can be "1", or "0".
Note 2: Set the corresponding port direction register to "0".

Figure 1.68: Timer Ai mode register in pulse-width modulation mode

Condition : Reload register high-order 8 bits = 0216
Reload register low-order 8 bits = 0216
External trigger (falling edge of TAiIN pin input signal) is selected



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

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 = 0016 to FE16; n = 0016 to FE16

Figure 1.69: Example of how an 8-bit pulse-width modulator operates

1.2.22 Timer B

Figure 1.70 shows the block diagram of timer B. Figure 1.71 and Figure 1.72 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).

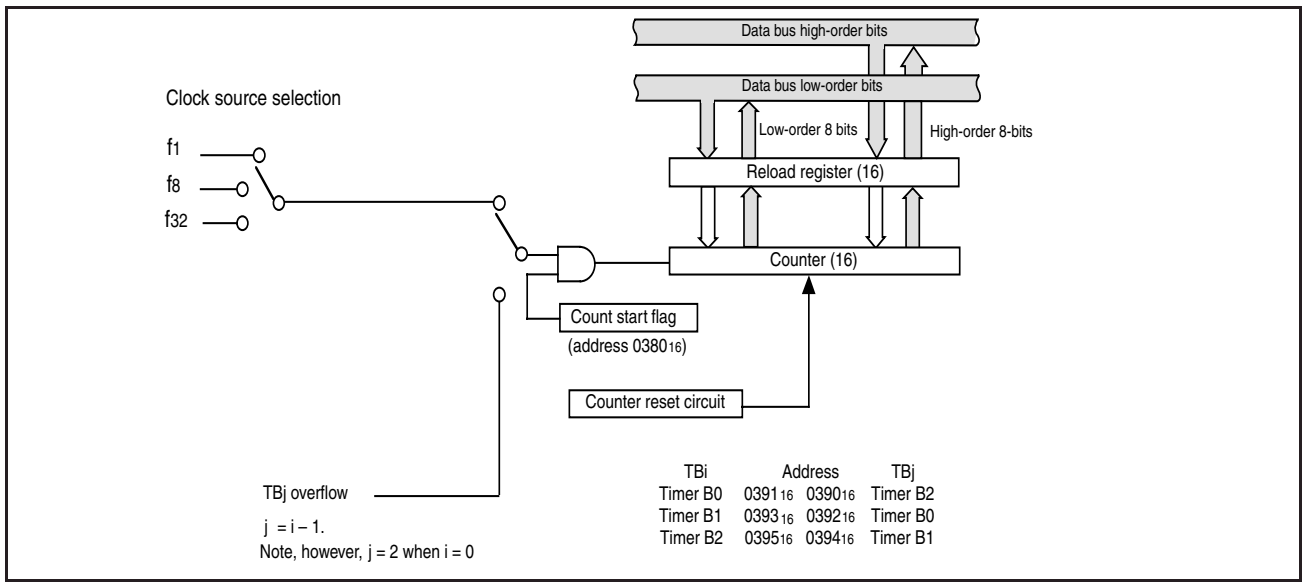


Figure 1.70: Block diagram of Timer B

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol: TBiMR(i=0 to 2)

Address: 039B16 to 039D16

When reset: 00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 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 assigned (i=1,2). In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : Reserved	○	○
TCK1			○	○

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

Figure 1.71: Timer B-related registers (1)

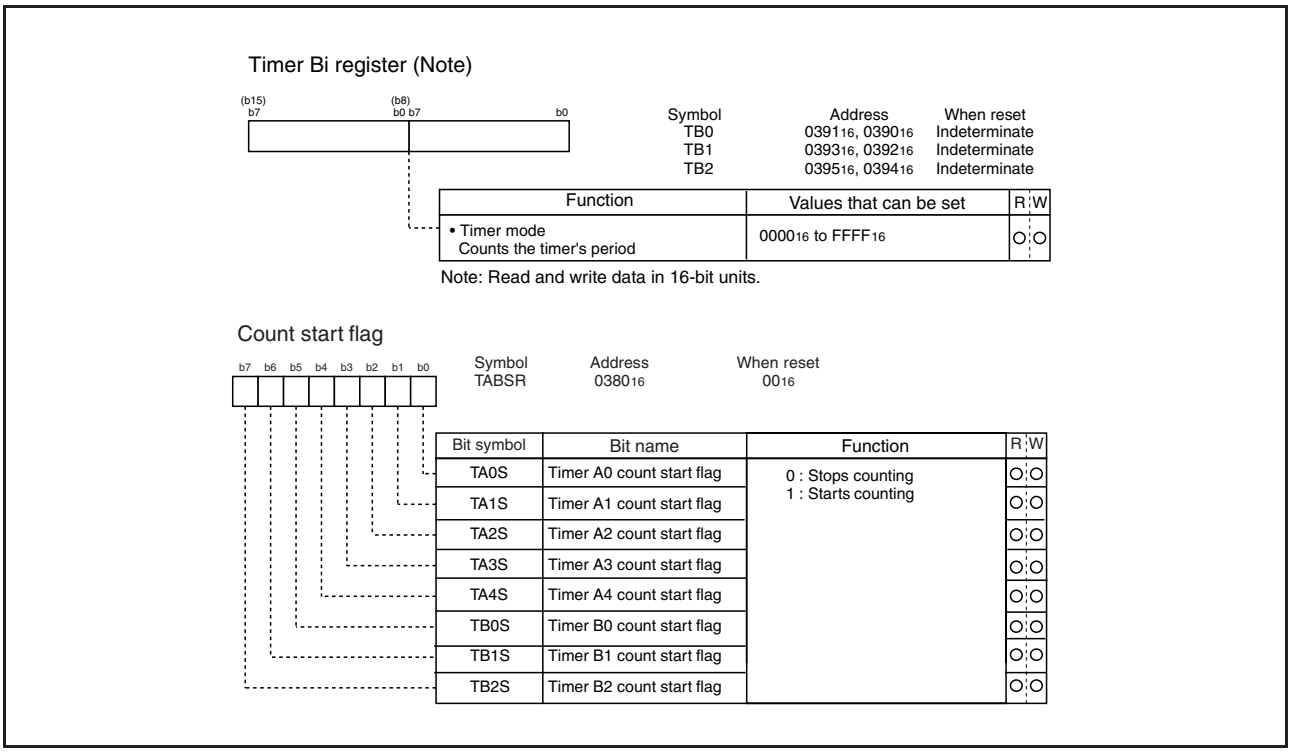


Figure 1.72: Timer B-related registers (2)

1.2.22.1 Timer mode

In this mode, the timer counts an internally generated count source as shown in Table 1.23 .

Table 1.23: Timer B 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)
Read from timer	Count value is read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to Timer Bi register, it is written to both reload register and counter • When counting is in progress When a value is written to Timer Bi register, it is written to only reload register (Transferred to counter at the next reload time)

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

1.2.23 UART0 to 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 1.73 shows the block diagram of UART0, UART1, and UART2. Figure 1.74 and Figure 1.75 show the block diagram of the transmit/receive unit.

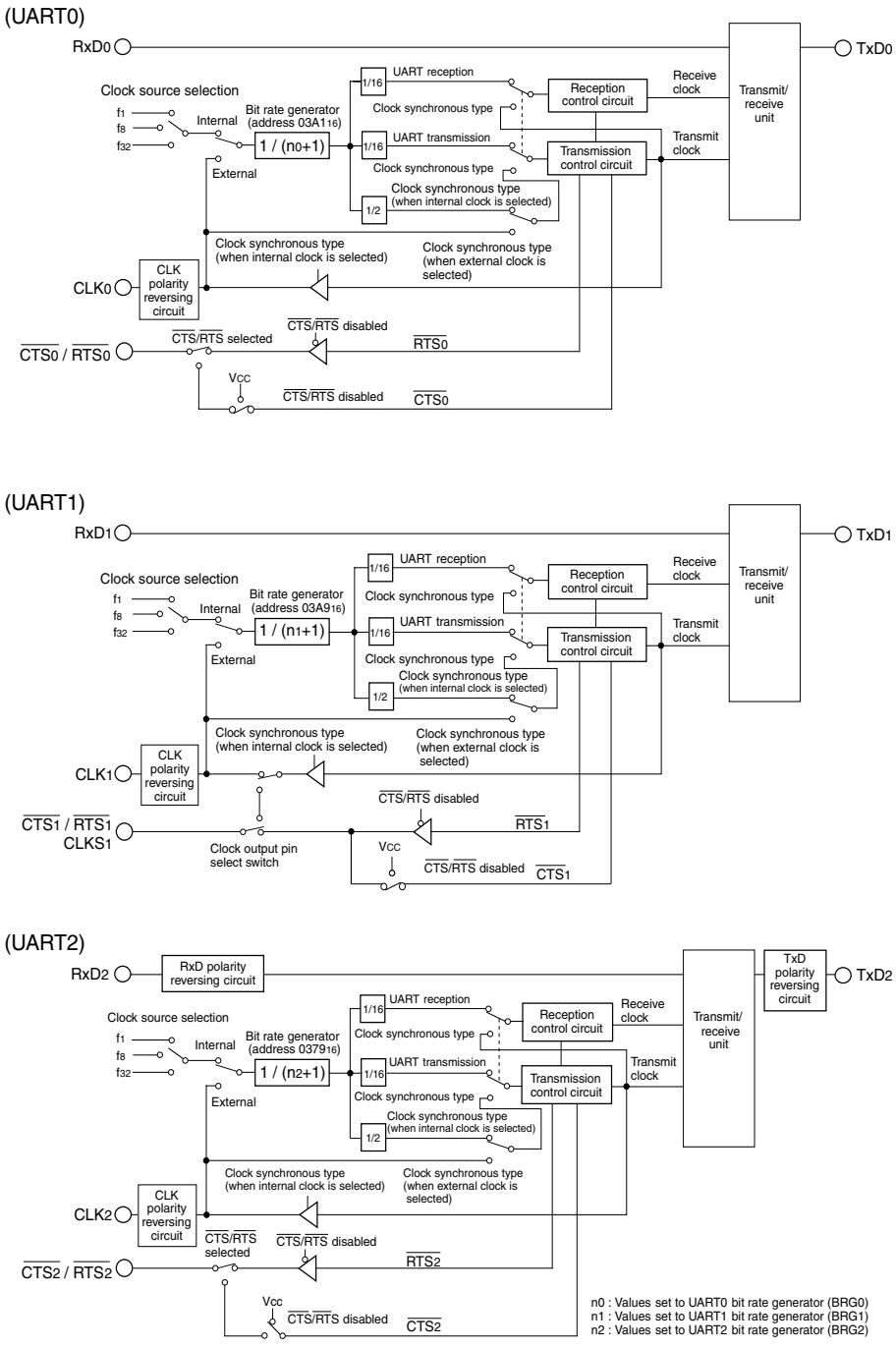


Figure 1.73: Block diagram of UARTi (i=0 to 2)

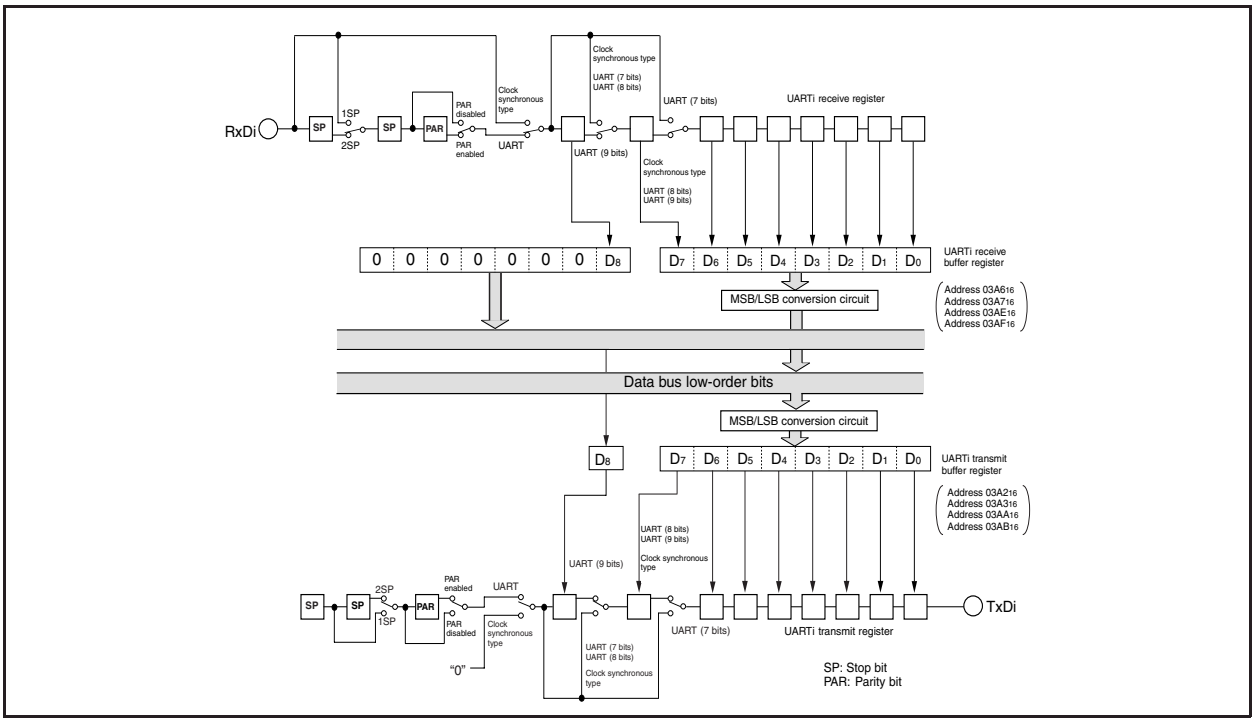


Figure 1.74: Block diagram of UARTi (i=0,1) transmit/receive circuit

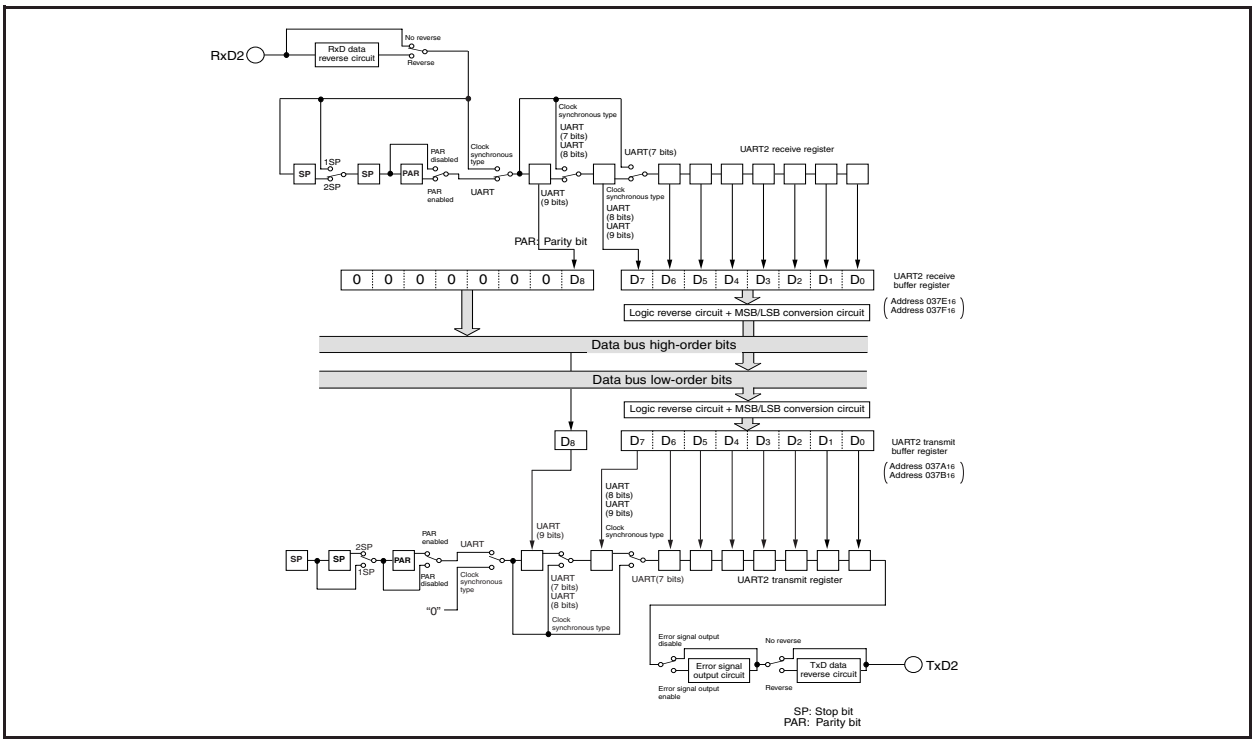


Figure 1.75: Block diagram of UART2 transmit/receive circuit

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.

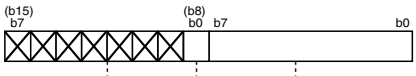
UART0 through UART2 are almost equal in their functions with minor exceptions.Table 1.24 shows the comparison of functions of UART0 through UART2, and Figure 1.76, Figure 1.77, Figure 1.78, Figure 1.79, and Figure 1.80 show the registers related to UARTi.

Table 1.24: Comparison of functions of UART0 through 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
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.

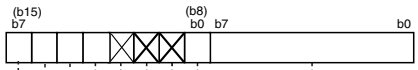
UARTi transmit buffer register



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

UARTi receive buffer register

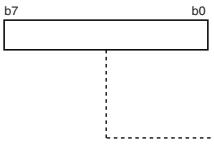


Symbol	Address	When reset
U0RB	03A7$_{16}$, 03A6$_{16}$	Indeterminate
U1RB	03AF$_{16}$, 03AE$_{16}$	Indeterminate
U2RB	037F$_{16}$, 037E$_{16}$	Indeterminate

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)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note)	Invalid	0 : No error 1 : Error found	○	×

Note : 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 1.76: Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

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

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select 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: U2MR Address: 037816 When reset: 00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 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 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	Must always be "0"	○	○
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"	○	○

Figure 1.77: Serial I/O-related registers (2)

UARTi transmit/receive control register 0

Symbol: UiC0(i=0,1) | Address: 03A4₁₆, 03AC₁₆ | When reset: 08₁₆

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f1 is selected 0 1 : f8 is selected 1 0 : f32 is selected 1 1 : Reserved	b1 b0 0 0 : f1 is selected 0 1 : f8 is selected 1 0 : f32 is selected 1 1 : Reserved	○	○
CLK1				○	○
CRS	CTS/RTS function select bit	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 2)	Valid when bit 4 = "0" 0 : 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	CTS/RTS disable bit	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (P6₀ and P6₄ function as programmable I/O port)	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (P6₀ and P6₄ function as 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	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

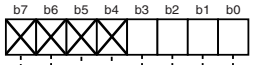
Symbol: U2C0 | Address: 037C₁₆ | When reset: 08₁₆

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f1 is selected 0 1 : f8 is selected 1 0 : f32 is selected 1 1 : Reserved	b1 b0 0 0 : f1 is selected 0 1 : f8 is selected 1 0 : f32 is selected 1 1 : Reserved	○	○
CLK1				○	○
CRS	CTS/RTS function select bit	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 2)	Valid when bit 4 = "0" 0 : 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	CTS/RTS disable bit	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (P7₃ functions programmable I/O port)	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (P7₃ 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 1.78: Serial I/O-related registers (3)

UARTi transmit/receive control register 1



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

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
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. Write "0" when writing to these bits. If read, the value is "0".				–	–

UART2 transmit/receive control register 1

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

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
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 1.79: Serial I/O-related registers (4)

UART transmit/receive control register 2

b7	b6	b5	b4	b3	b2	b1	b0
⊠	0						

Symbol: UCON | Address: 03B0₁₆ | When reset: X0000000₂

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"	○	○
Reserved		Must always be "0"	Must always be "0"	○	○
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 1.80: Serial I/O-related registers (5)

1.2.23.1 Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Figure 1.81 shows the UARTi transmit/receive mode register. Table 1.25 lists the specifications of the clock synchronous serial I/O mode.

UARTi transmit/receive mode registers

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

Symbol: UiMR(i=0,1) | Address: 03A0₁₆, 03A8₁₆ | When reset: 00₁₆

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

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

Symbol: U2MR | Address: 037816 | When reset: 00₁₆

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 1: Usually set to "0".

Figure 1.81: UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 1.25: Specifications of Clock synchronous serial I/O mode

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 (Maximum 5 Mbps.)
Transmission/reception control	$\overline{CTS}$ function/$\overline{RTS}$ function/$\overline{CTS}$, $\overline{RTS}$ 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"
Interrupt request generation timing	•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 •When receiving Interrupt requested when the data transfer from the UARTi receive register to the UARTi receive buffer register is complete.
Error detection	•Overrun error (Note 2) This error occurs when the next data is ready before contents of UARTi receive buffer is read.
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) UART1 transfer clock can be chosen by software to be output from one of the two pins set •Switching serial data logic (UART2) Whether to reverse data in writing to the transmission buffer register or reading the reception buffer register can be selected. •Switching serial data logic (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

Note 1: "n" denotes the value 00_{16} to FF_{16} that is set to the UART bit rate generator.

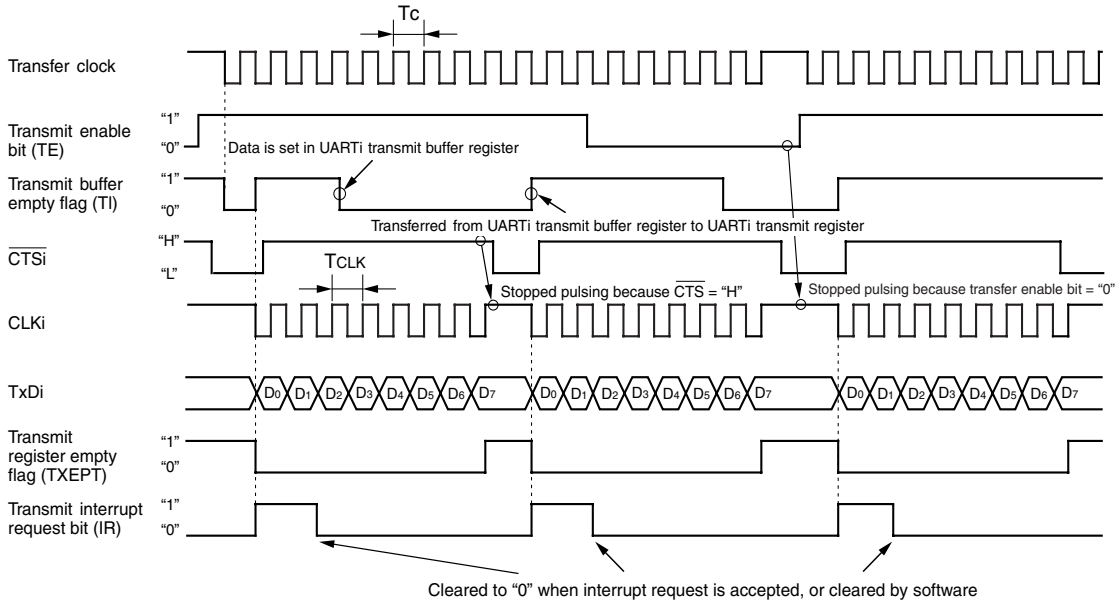
Note 2: The UARTi receive buffer has the next data written when an overrun error occurs. Note: the UARTi receive interrupt request bit is set to "0".

Table 1.26 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions when the transfer clock output from multiple pins function is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxD pin outputs a “H”. The typical clock synchronous timing diagrams are shown in Figure 1.82.

Table 1.26: IART mode input/output pin functions

Pin name	Function	Method of selection
TxDi ($P6_3$, $P6_7$, $P7_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi ($P6_2$, $P6_6$, $P7_1$)	Serial data input	Port $P6_2$, $P6_6$, and $P7_1$ direction register (bits 2 and 6 at address $03EE_{16}$ bit 1 at address $03EF_{16}$)= “0” (Can be used as an input port when performing transmission only.)
CLKi ($P6_1$, $P6_5$, $P7_2$)	Transfer clock output	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, 0378_{16}) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, 0378_{16}) = “1” Port $P6_1$, $P6_5$, and $P7_2$ direction register (bits 1 and 5 at address $03EE_{16}$, bit 2 at address $03EF_{16}$) = “0”
$\overline{CTS}i/\overline{RTS}i$ ($P6_0$,$P6_4$,$P7_3$)	$\overline{CTS}$ input	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” $\overline{CTS}/\overline{RTS}$ function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” Port $P6_0$, $P6_4$ and $P7_3$ direction register (bits 0 and 4 at address $03EE_{16}$, bit 3 at address $03EF_{16}$) = “0”
	$\overline{RTS}$ output	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “0” $\overline{CTS}/\overline{RTS}$ function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “1”
	Programmable I/O port	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = “1”

Example of transit timing (when internal clock is selected)

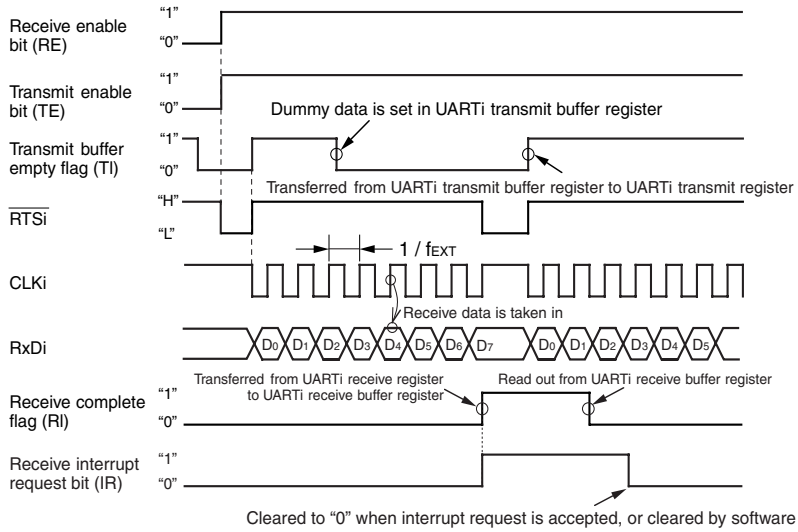


Shown in ( ) are bit symbols.

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

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

Example of receive timing (when external clock is selected)



Shown in ( ) are bit symbols.

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

fEXT: frequency of external clock

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

Figure 1.82: Typical transmit/receive timings in clock synchronous serial I/O mode Polarity select function

1.2.23.1.1 Polarity select function

As shown in Figure 1.83, the CLK polarity select bit (bit 6 at addresses $03A4_{16}$, $03AC_{16}$, $037C_{16}$) allows selection of the polarity of the transfer clock.

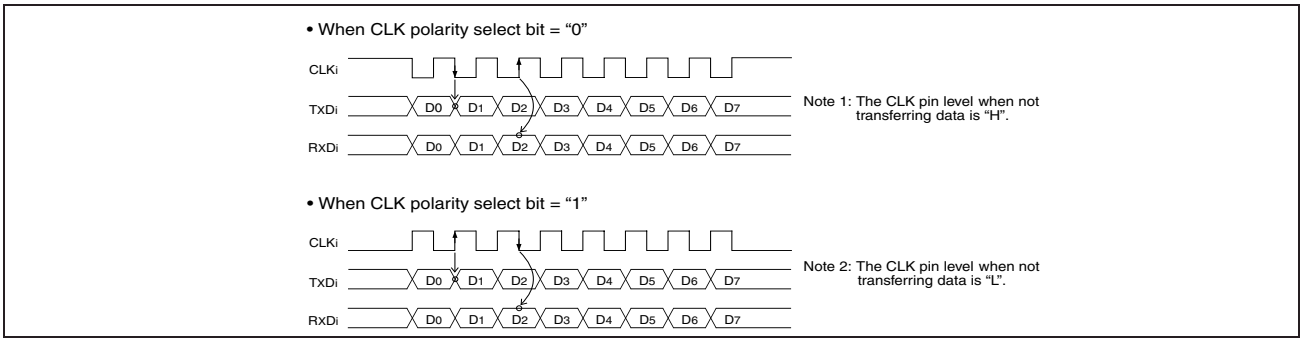


Figure 1.83: Polarity of transfer clock

1.2.23.1.2 LSB first/MSB first select function

As shown in Figure 1.84, 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".

Note: This applies CLK polarity select bit = "0".

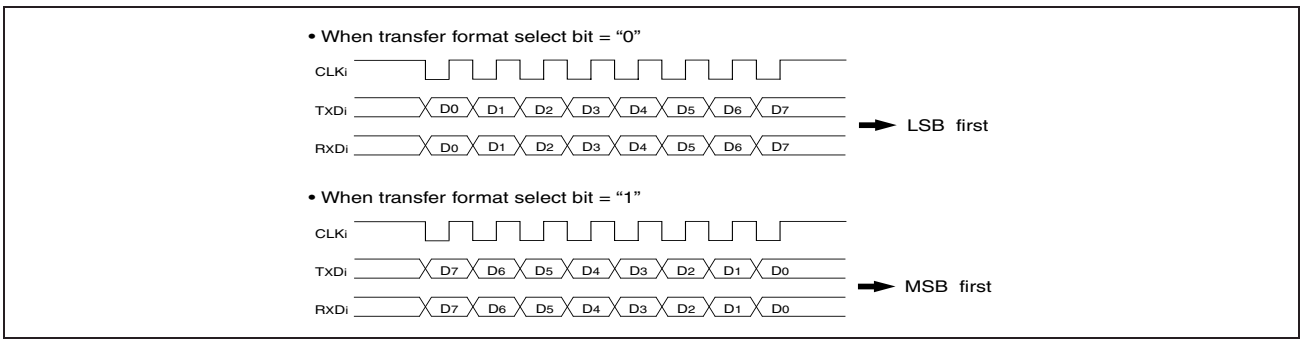


Figure 1.84: Transfer format

1.2.23.1.3 Transfer clock output from multiple pins function (UART1)

This function allows using two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address $03B0_{16}$). See Figure 1.85. The multiple pins function is valid only when the internal clock is selected for UART1. Note that when this function is selected, the UART1 CTS/RTS function cannot be used.

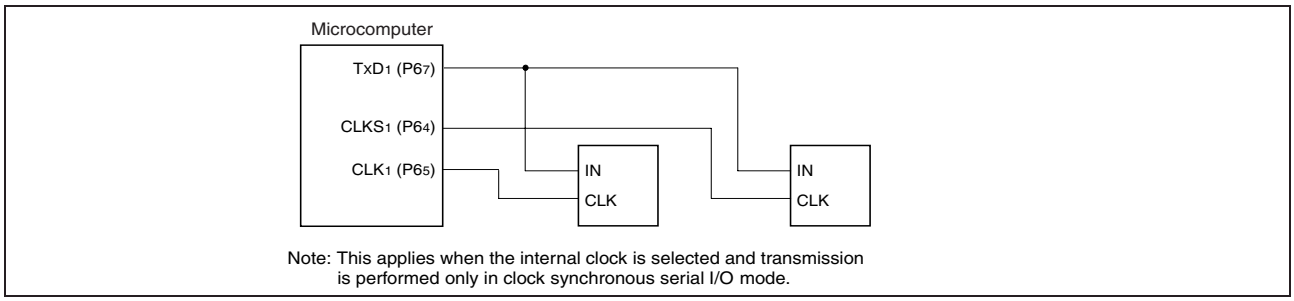


Figure 1.85: The transfer clock output from the multiple pins function usage

1.2.23.1.4 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.

1.2.23.1.5 Serial data logic switch function (UART2)

When the data logic select bit (bit6 at address $037D_{16}$) = "1", and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 1.86 shows the example of serial data logic switch timing.

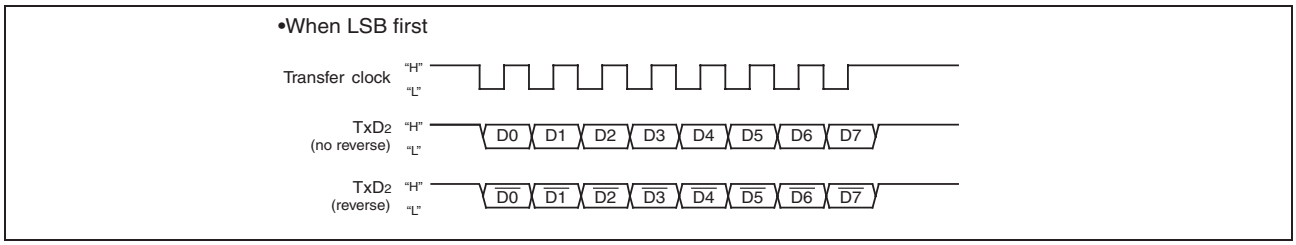


Figure 1.86: Serial data logic switch timing

1.2.23.2 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 1.27 lists the specifications of the UART mode.

Table 1.27: Specifications of UART mode

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 $03A0_{16}$, $03A8_{16}$=“1”): fEXT/16(n+1)(Note 1) (Note 2)
Transmission/reception control	• $\overline{CTS}$ function/$\overline{RTS}$ function/ $\overline{CTS}$, $\overline{RTS}$ function disabled
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 $\overline{CTS}$ function selected, $\overline{CTS}$ input level = “L”
Reception start condition	• To start reception, the following requirements must be met: Receive enable bit (bit 2 at addresses $03A5_{16}$, $03AD_{16}$, $037D_{16}$) = “1” Start bit detection
Interrupt request generation timing	• When transmitting Transmit interrupt cause select bits (bits 0,1 at address $03B0_{16}$, bit4 at address $037D_{16}$) = “0”: Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed Transmit interrupt cause select bits (bits 0, 1 at address $03B0_{16}$, bit4 at address $037D_{16}$) = “1”: Interrupts requested when data transmission from UARTi transfer register is completed • When receiving Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 3) 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
Select function	• 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 (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

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. External clock cannot be selected in UART2.

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”

UARTi transmit / receive mode registers

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: UiMR(i=0,1) | Address: 03A0₁₆, 03A8₁₆ | When reset: 00₁₆

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

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

UART2 transmit / receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: U2MR | Address: 037816 | When reset: 00₁₆

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

Note: Usually set to "0".

Figure 1.87: UARTi transmit/receive mode register.

Table 1.28 lists the functions of the input/output pins during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H".

Table 1.28: Input/output pin functions in UART mode

Pin name	Function	Method of selection
TxDi ($P6_3$, $P6_7$, $P7_0$)	Serial data output	Outputs dummy data when performing reception.
RxDi ($P6_2$, $P6_6$, $P7_1$)	Serial data input	Port $P6_2$, $P6_6$, and $P7_1$ direction register (bits 2 and 6 at address $03EE_{16}$ bit 1 at address $03EF_{16}$)= "0" (Can be used as an input port when performing transmission only.)
CLKi ($P6_1$, $P6_5$, $P7_2$)	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 $03A0_{16}$, $03A8_{16}$) = "1" Port $P6_1$, $P6_5$ direction register (bits 1 and 5 at address $03EE_{16}$) = "0"
$\overline{CTS}i/\overline{RTS}i$ ($P6_0$,$P6_4$,$P7_3$)	$\overline{CTS}$ input	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "0" $\overline{CTS}/\overline{RTS}$ 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"
	$\overline{RTS}$ output	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "0" $\overline{CTS}/\overline{RTS}$ function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "1"
	Programmable I/O port	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "1"

Figure 1.88 and Figure 1.89 show the typical UART mode transmit and receive timing diagrams.

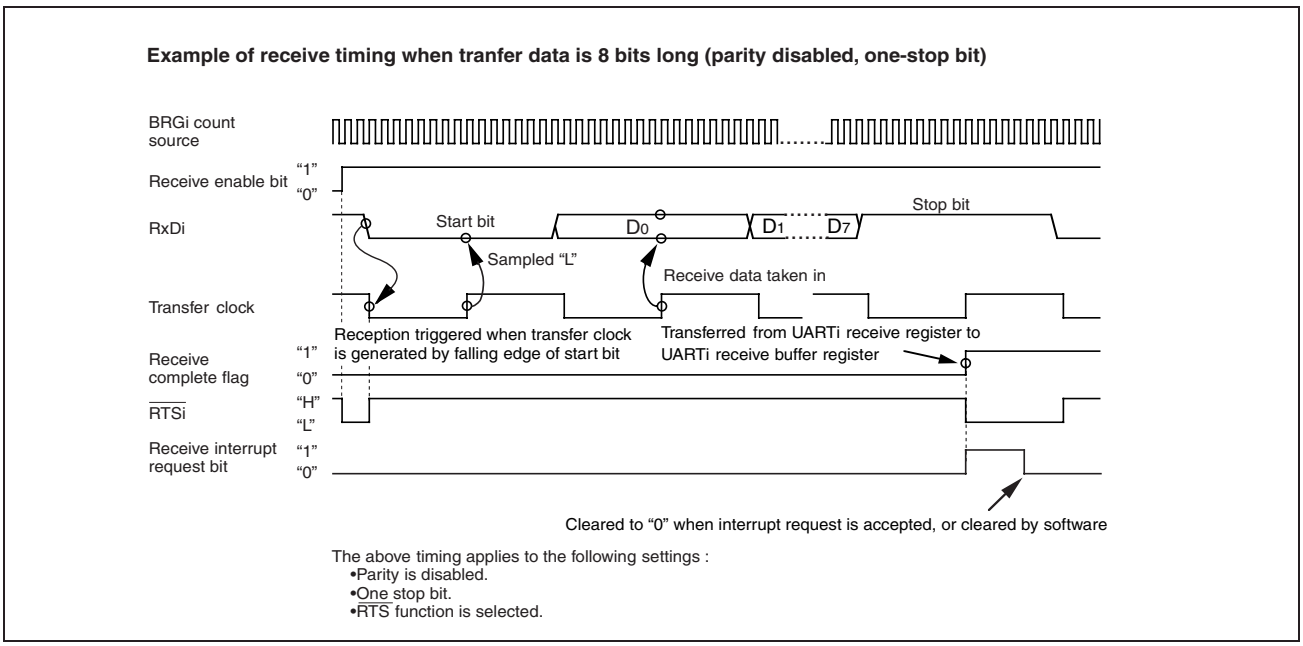
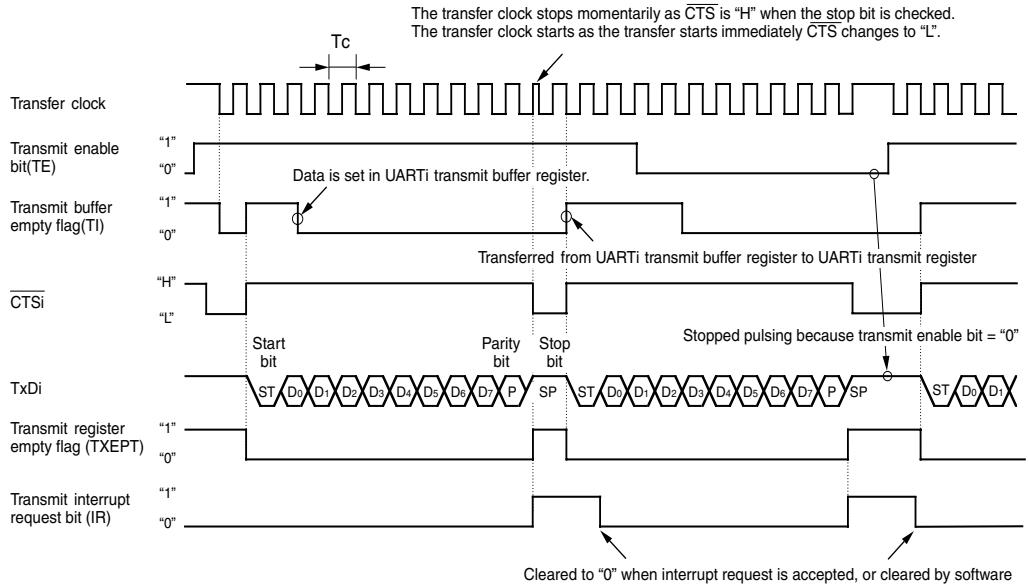


Figure 1.88: Typical receive timing in UART mode

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

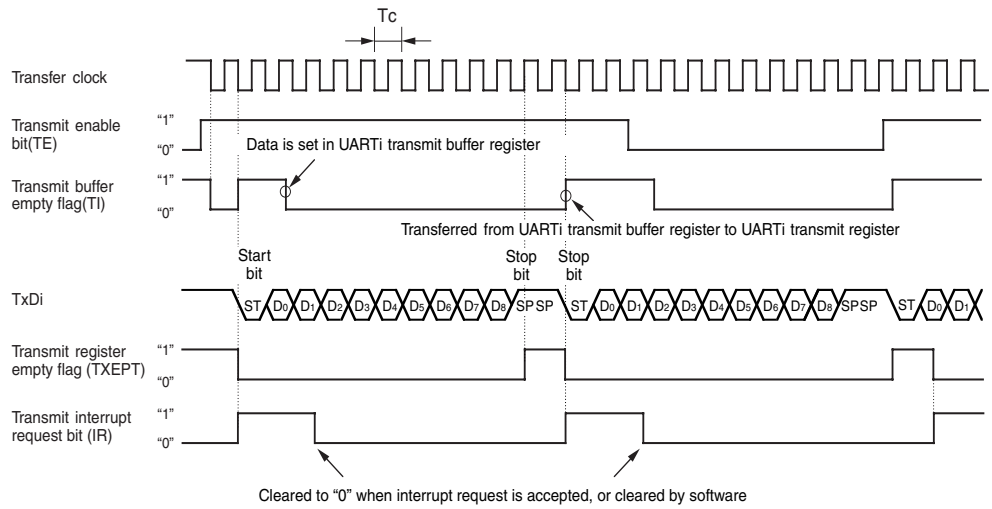


Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Example of transmit timing when transfer data are 9 bits long (parity enabled, two stop bits)



Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 1.89: Typical transmit timings in UART mode

1.2.23.2.1 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”.

1.2.23.2.2 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 1.90 shows the example of timing for switching serial data logic.

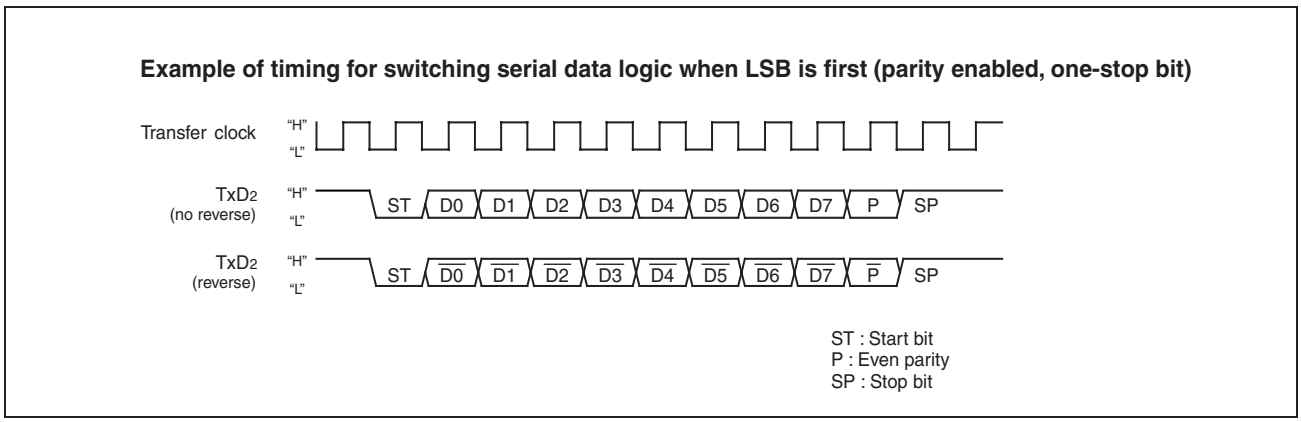


Figure 1.90: Timing for switching serial data logic

1.2.23.2.3 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.

1.2.23.2.4 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 1.91 shows the example of detection timing of a buss collision (in UART mode).

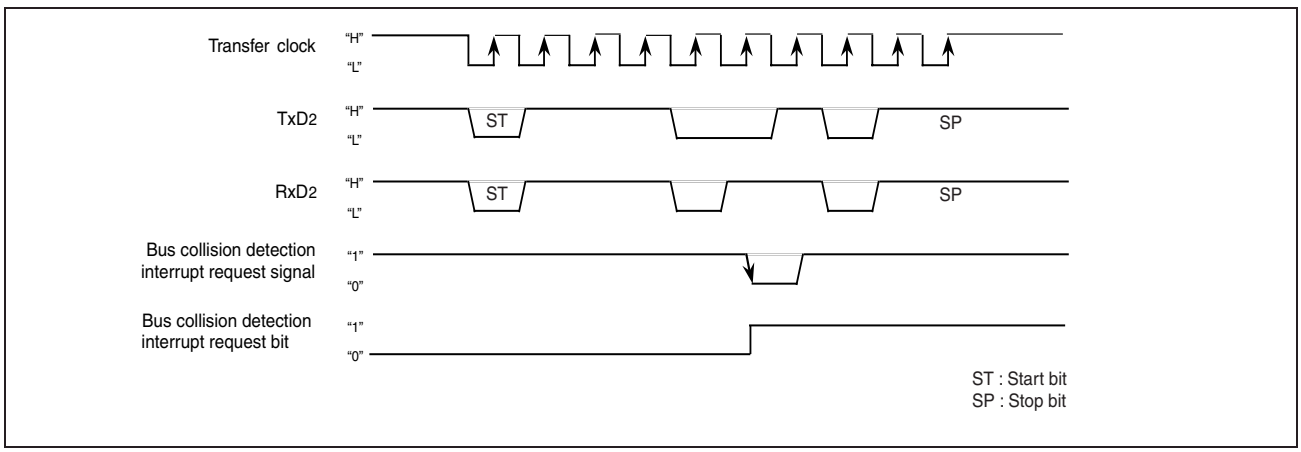


Figure 1.91: Detection timing of a bus collision (in UART mode)

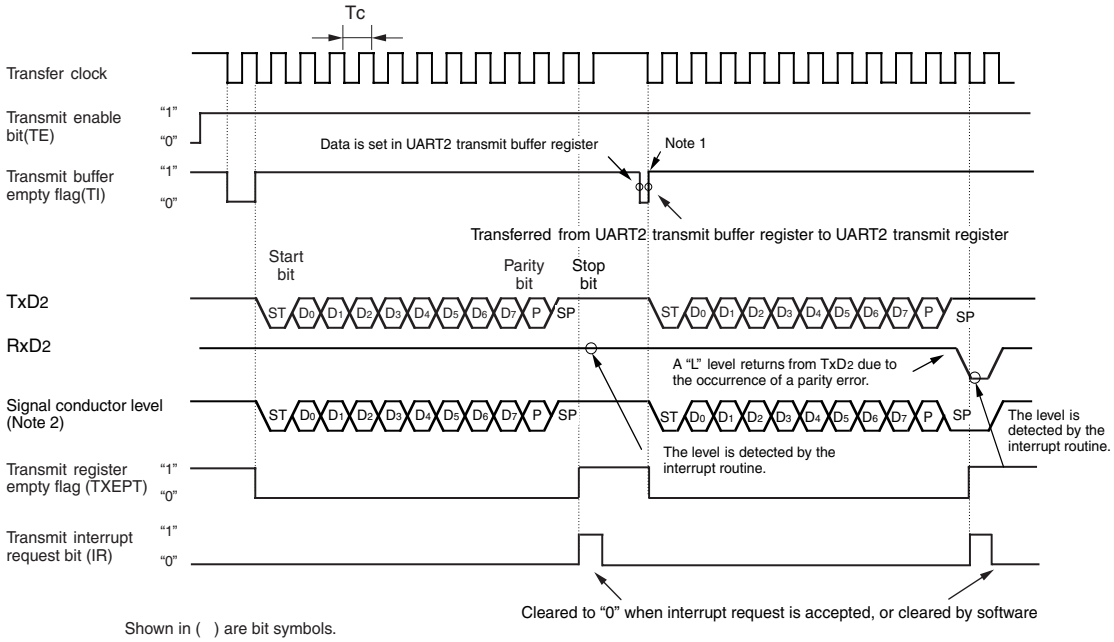
1.2.23.3 Clock-asynchronous serial I/O mode (compliant with the SIM interface)

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 1.29 shows the specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface). Figure 1.92 shows the typical transmit/receive timing in UART mode.

Table 1.29: 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} = “101_2”) • 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
Transmission / reception control	• Disable the $\overline{CTS}$ and $\overline{RTS}$ 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
Interrupt request generation timing	• 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: If an overrun error occurs, the UART2 receive buffer will have the next data written in. Also, the UARTi receive interrupt request bit is not set to “1”.

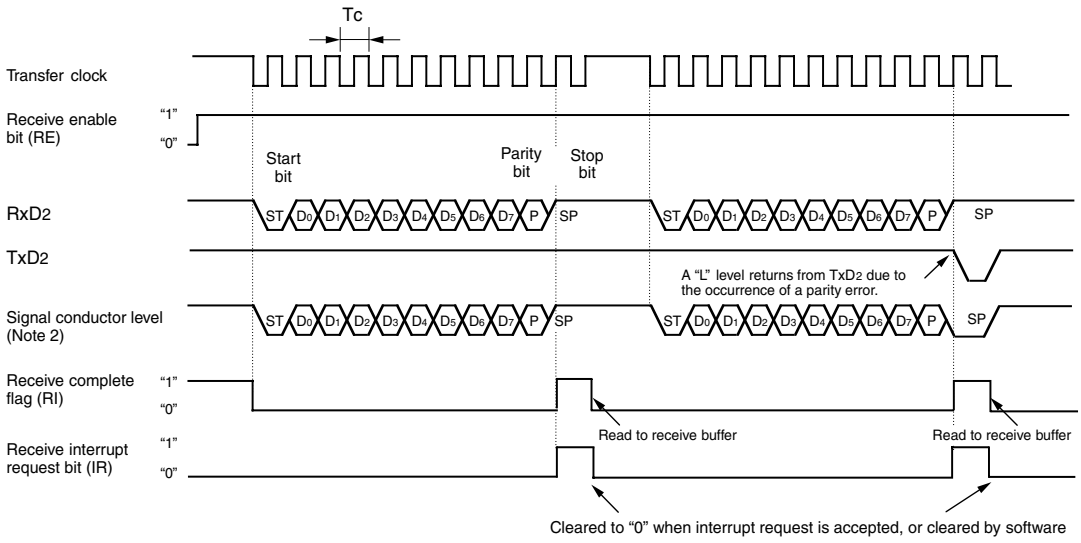


Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fi
fi : frequency of BRG2 count source (f1, f8, f32)
n : value set to BRG2

Note 1: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.
Note 2: Equal in waveform because TxD2 and RxD2 are connected.



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
fi : frequency of BRG2 count source (f1, f8, f32)
n : value set to BRG2

Note 2: Equal in waveform because TxD2 and RxD2 are connected.

Figure 1.92: Typical transmit/receive timing in UART mode (compliant with the SIM interface)

1.2.23.3.1 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 1.93 shows the output timing of the parity error signal.

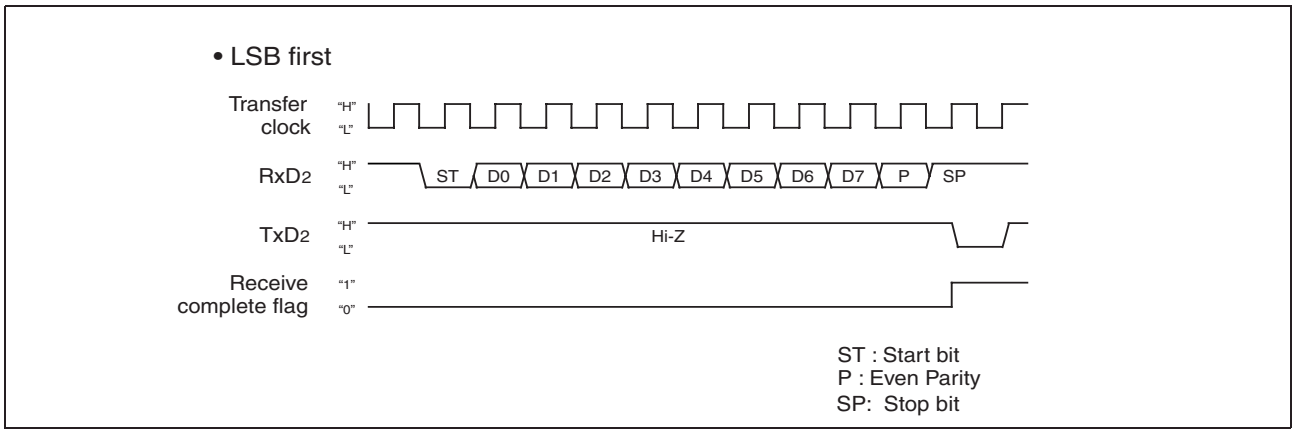


Figure 1.93: Output timing of the parity error signal

1.2.23.3.2 Direct format/inverse format

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

Figure 1.94 shows the SIM interface format.

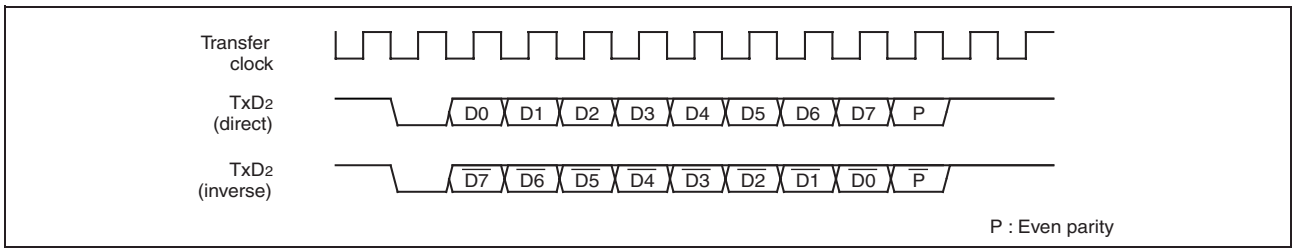


Figure 1.94: SIM interface format

Figure 1.95 shows the example of connecting the SIM interface with TxD2 and RxD2.

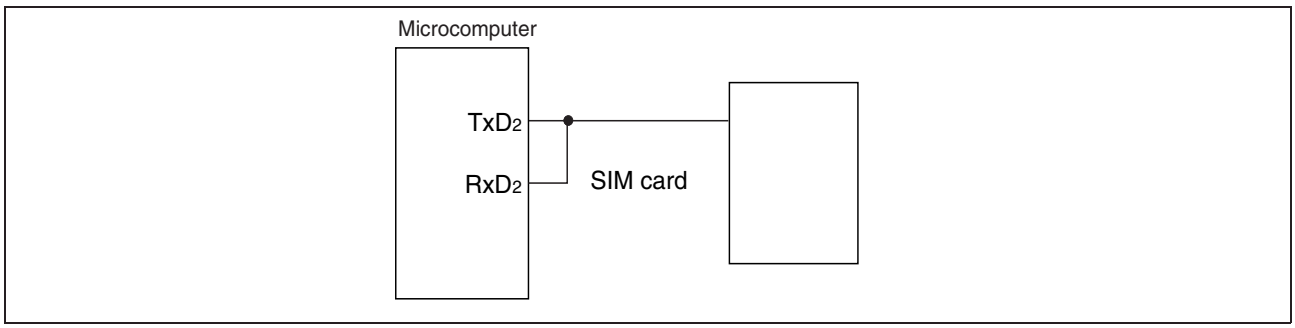


Figure 1.95: Connecting the SIM interface

1.2.24 A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P10_0$ to $P10_7$ 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 (V_{REF}) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of $03D7_{16}$ to connect V_{REF}.

The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table 1.30 shows the performance of the A-D converter. Figure 1.96 shows the block diagram of the A-D converter, and Figure 1.97 and Figure 1.98 show the A-D converter-related registers.

Table 1.30: Performance of A-D Converter[a]

Item	Performance	
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)	
Analog input voltage (Note)	0V to AV_{CC} (V_{CC})	
Operating clock fAD	V_{CC} = 5V	f_{AD}/divide-by-2 or f_{AD}/divide-by-4 or f_{AD}, f_{AD},$f(X_{IN})$
Resolution	8-bit or 10-bit (selectable)	
Absolute precision	V_{CC} = 5V	• Without sample and hold function ±3LSB • With sample and hold function (8-bit resolution) ±2LSB • With sample and hold function (10-bit resolution) 3LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1	
Analog input pins	8pins (AN_0 to AN_7)	
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 re-triggered) A-D conversion starts when the A-D conversion start flag is “1” and the $\overline{AD}_{\overline{TRG}}$/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: Does not depend on use of sample and hold function

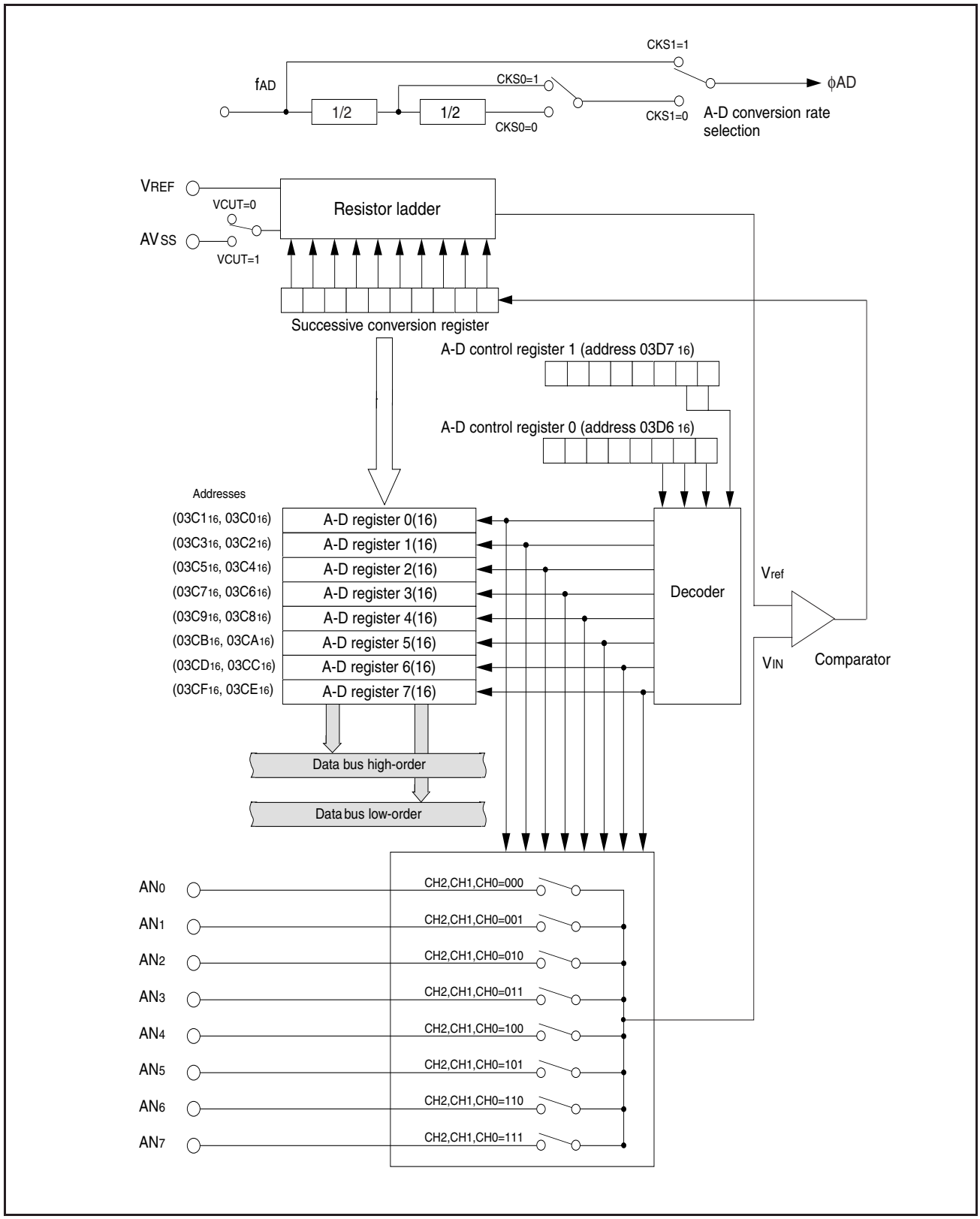


Figure 1.96: Block diagram of A-D converter

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: ADCON0 Address: 03D6₁₆ When reset: 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1 (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 : fAD/4 is selected 1 : fAD/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
0	0						

Symbol: ADCON1 Address: 03D7₁₆ When reset: 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins)	○	○
SCAN1		When repeat sweep mode 1 is selected b1 b0 0 0 : AN0 (1 pin) 0 1 : AN0, AN1 (2 pins) 1 0 : AN0 to AN2 (3 pins) 1 1 : AN0 to AN3 (4 pins)	○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : fAD/2 or fAD/4 is selected 1 : fAD is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
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 1.97: A-D converter-related registers (1)

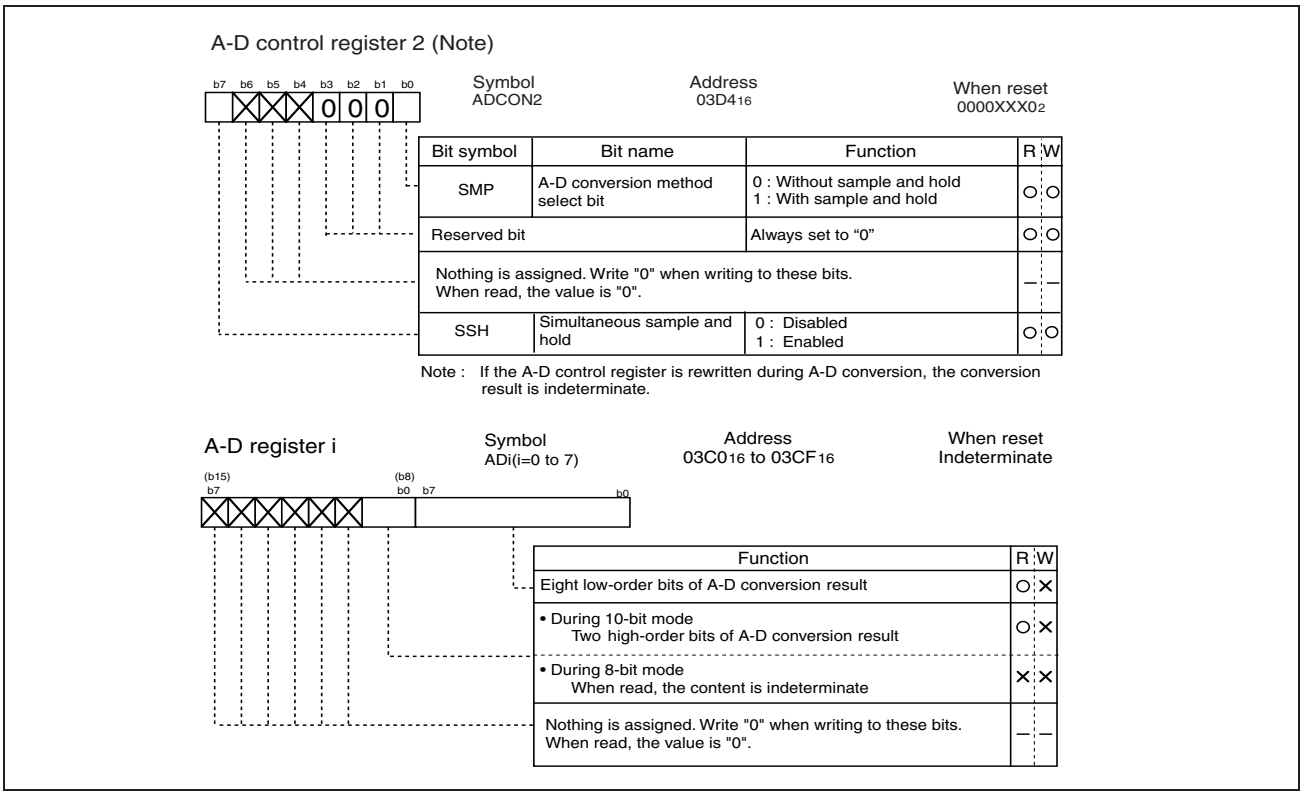


Figure 1.98: A-D converter-related registers (2)

1.2.24.1 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 1.31 shows the specifications of one-shot mode. Figure 1.99 shows the A-D control register in one-shot mode.

Table 1.31: 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 AN_0 to AN_7, 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 03D6₁₆ — When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN0 is selected 0 0 1 : AN1 is selected 0 1 0 : AN2 is selected 0 1 1 : AN3 is selected 1 0 0 : AN4 is selected 1 0 1 : AN5 is selected 1 1 0 : AN6 is selected 1 1 1 : AN7 is selected (Note 2)	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode (Note 2)	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0: fAD/4 is selected 1: fAD/2 is selected	O	O

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
0	0	1			0		

Symbol ADCON1 — Address 03D7₁₆ — When reset 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in one-shot mode	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit1	0 : fAD/2 or fAD/4 is selected 1 : fAD is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
Reserved bit		Always set to "0"	O	O

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

Figure 1.99: A-D conversion register in one-shot mode

1.2.24.2 Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion.Table 1.32 shows the specifications of repeat mode. Figure 1.100 shows the A-D control register in repeat mode.

Table 1.32: Repeat mode specification

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D
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)

b7	b6	b5	b4	b3	b2	b1	b0
			0	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)	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode (Note 2)	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{ADTRG}$ trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	O	O

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

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

Figure 1.100: A-D conversion register in repeat mode

1.2.24.3 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 1.33 shows the specifications of single-sweep mode. Figure 1.101 shows the A-D control register in single-sweep mode.

Table 1.33: 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	AN_0 and AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	0			

Symbol ADCON0 — Address 03D6₁₆ — When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in single sweep mode	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : fAD/4 is selected 1 : fAD/2 is selected	O	O

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

A-D control register 1 (Note)

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

Symbol ADCON1 — Address 03D7₁₆ — When reset 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit 1	0 : fAD/2 or fAD/4 is selected 1 : fAD is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
Reserved bit		Always set to "0"	O	O

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

Figure 1.101: A-D conversion register in single-sweep mode

1.2.24.4 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 1.34 shows the specifications of repeat-sweep mode 0. Figure 1.102 shows the A-D control register in repeat-sweep mode 0.

Table 1.34: 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	AN_0 and AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

Symbol ADCON0 — Address 03D6₁₆ — When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	(b4 b3) 1 1 : Repeat sweep mode 0	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : fAD/4 is selected 1 : fAD/2 is selected	O	O

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

A-D control register 1 (Note)

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

Symbol ADCON1 — Address 03D7₁₆ — When reset 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected (b1 b0) 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit 1	0 : fAD/2 or fAD/4 is selected 1 : fAD is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
Reserved bit		Always set to "0"	O	O

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

Figure 1.102: A-D conversion register in repeat-sweep mode 0

1.2.24.5 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 1.35 shows the specifications of repeat-sweep mode 1. Figure 1.103 show the A-D control in repeat-sweep mode 1.

Table 1.35: 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: AN_0 selected AN_0 -> AN_1 -> AN_0 -> AN_2 -> AN_0 -> AN_3, 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	AN_0 (1 pin), AN_0 and AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (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)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

Symbol: ADCON0 Address: 03D6₁₆ When reset: 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 1	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	O	O

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

A-D control register 1 (Note)

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

Symbol: ADCON1 Address: 03D7₁₆ When reset: 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)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	1 : Repeat sweep mode 1	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
Reserved bit		Always set to "0"	O	O

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

Figure 1.103: A-D conversion register in repeat-sweep mode 1

1.2.24.6 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.

1.2.25 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 ($X^{16} + X^{12} + X^5 + 1$) 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 1.104 shows the block diagram of the CRC circuit. Figure 1.105 shows the CRC-related registers.

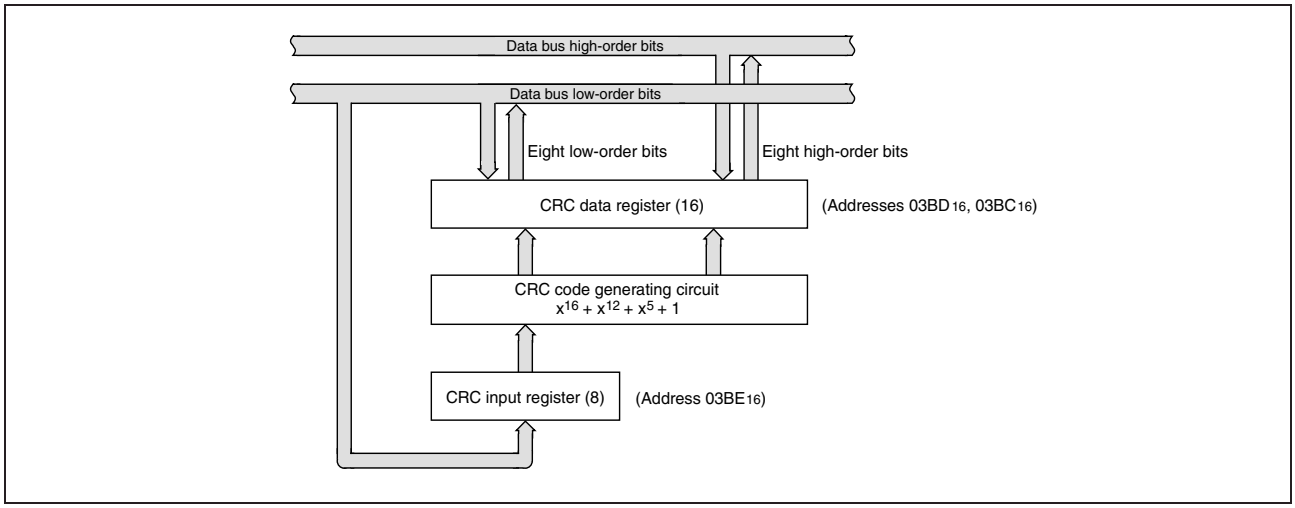


Figure 1.104: Block diagram of CRC circuit

CRC data register

(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
CRCD	03BD16, 03BC16	Indeterminate

Function	Values that can be set	R	W
CRC calculation result output register	000016 to FFFF16	O	O

CRC input register

b7 b0

Symbo	Address	When reset
CRCIN	03BE16	Indeterminate

Function	Values that can be set	R	W
Data input register	0016 to FF16	O	O

Figure 1.105: CRC-related registers

1.2.26 Programmable I/O Ports

There are 63 programmable I/O ports: P0 to P3, P6 to P8 (excluding $P8_5$), 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. $P8_5$ is an input-only port and has no built-in pull-up resistance.

Figure 1.106, Figure 1.107 and Figure 1.108 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 1.113 and Table 1.36 .

1.2.26.1 Direction registers

Figure 1.109 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 $P8_5$.

1.2.26.2 Port registers

Figure 1.110 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.

1.2.26.3 Pull-up control registers

Figure 1.111 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.

1.2.26.4 High drive capacity registers

Figure 1.112 shows the Port 2 and PWM drive capacity register. Port 2 can be configured to drive an LED by increasing the drive strength of the corresponding bit's N-channel transistor. Each Timer output ($TA0_{OUT}$ to$TA4_{OUT}$) can be configured for high-drive capability by increasing the drive strength of the corresponding bits.

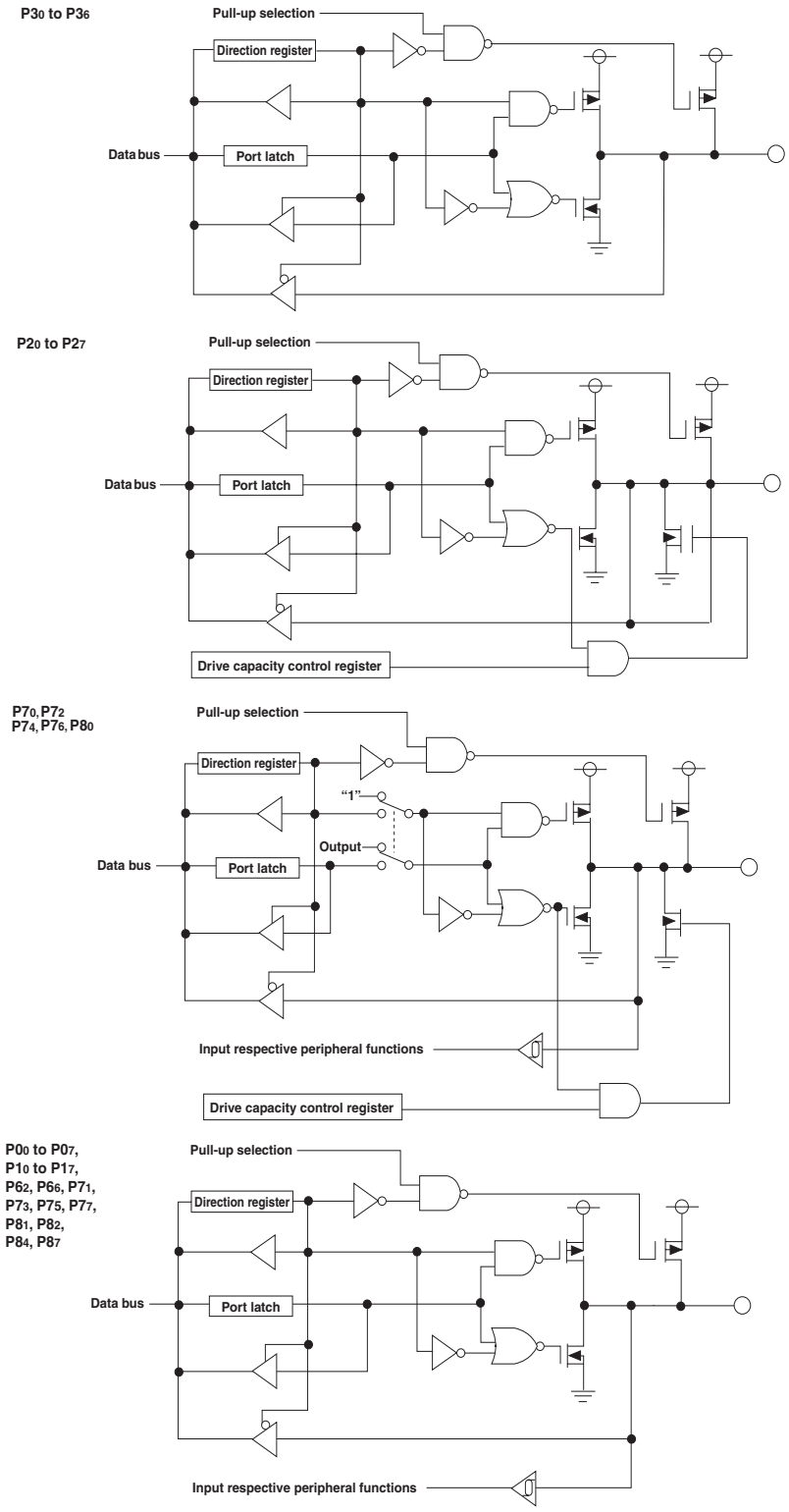


Figure 1.106: Programmable I/O ports (1)

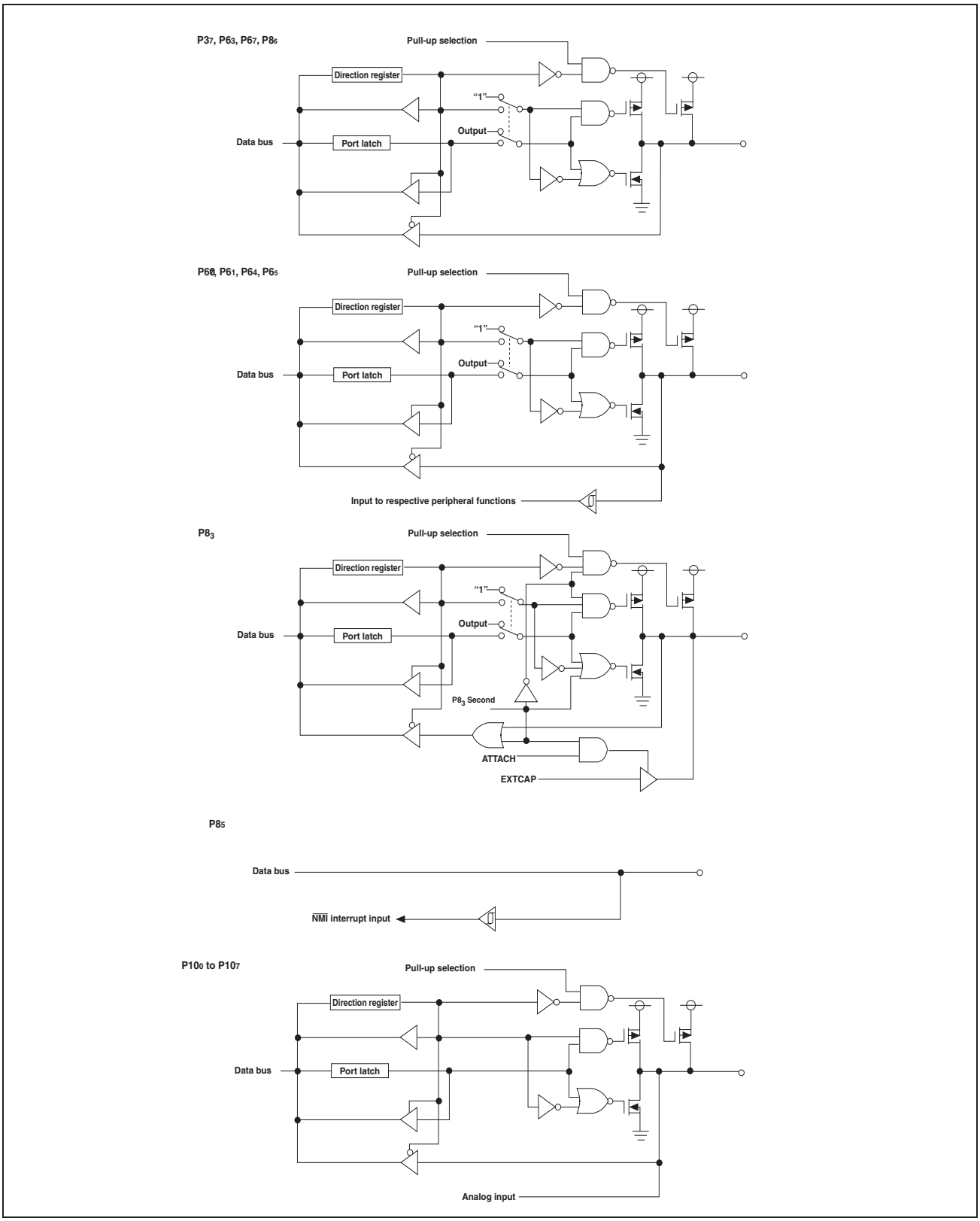
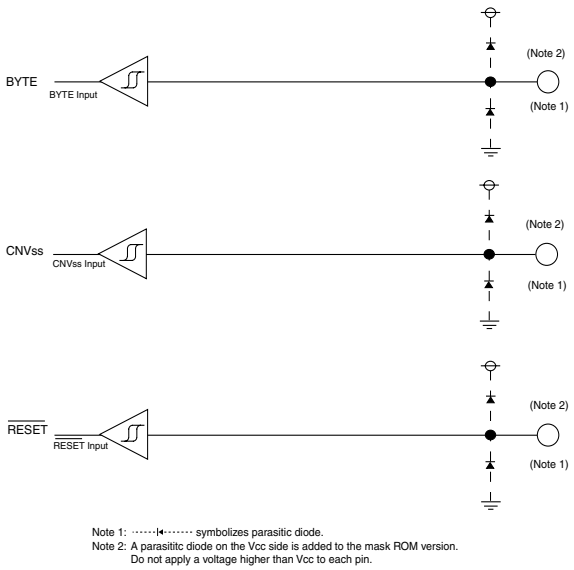


Figure 1.107: Programmable I/O ports (2)



Note 1: ·····|<····· symbolizes parasitic diode.
Note 2: A parasititc diode on the Vcc side is added to the mask ROM version.
Do not apply a voltage higher than Vcc to each pin.

Figure 1.108: Programmable I/O Ports (3)

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
PDi (i = 0 to 3,6,7,10)	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$, 03EE$_{16}$, 03EF$_{16}$, 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 3,6,7,10)	○	○
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		○	○

Port P8 direction register

b7 b6 b5 b4 b3 b2 b1 b0

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 1.109: Direction register

Port Pi register

Symbol	Address	When reset
Pi (i = 0 to 3,6,7,10)	03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$ 03EC$_{16}$, 03ED$_{16}$, 03F4$_{16}$	Indeterminate Indeterminate

Bit symbol	Bit name	Function	R	W
Pi_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 3,6,7,10)	○	○
Pi_1	Port Pi$_1$ register		○	○
Pi_2	Port Pi$_2$ register		○	○
Pi_3	Port Pi$_3$ register		○	○
Pi_4	Port Pi$_4$ register		○	○
Pi_5	Port Pi$_5$ register		○	○
Pi_6	Port Pi$_6$ register		○	○
Pi_7	Port Pi$_7$ register		○	○

Port P8 register

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

Bit symbol	Bit name	Function	R	W
P8_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	○	○
P8_1	Port P8$_1$ register		○	○
P8_2	Port P8$_2$ register		○	○
P8_3	Port P8$_3$ register		○	○
P8_4	Port P8$_4$ register		○	○
P8_5	Port P8$_5$ register		○	×
P8_6	Port P8$_6$ register		○	○
P8_7	Port P8$_7$ register		○	○

Figure 1.110: 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		○	○

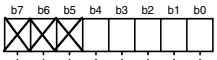
Figure 1.111: Pull-up control register

Port 2 Drive Capacity register

Symbol: P2DR | Address: 03FA16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
P2DR0	P20 LED drive capacity	The N-channel high-drive capacity is activated for the corresponding bit. 0 : Normal drive 1 : N-channel high drive	O	O
P2DR1	P21 LED drive capacity		O	O
P2DR2	P22 LED drive capacity		O	O
P2DR3	P23 LED drive capacity		O	O
P2DR4	P24 LED drive capacity		O	O
P2DR5	P25 LED drive capacity		O	O
P2DR6	P26 LED drive capacity		O	O
P2DR7	P27 LED drive capacity		O	O

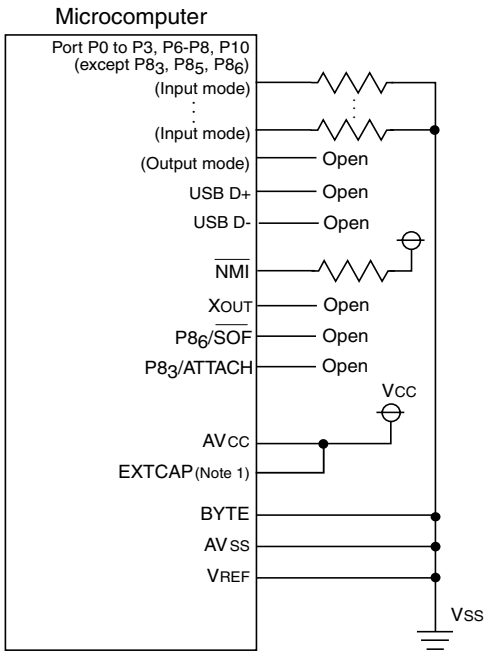
Timer A Output Drive Capacity register



Symbol: TADR | Address: 03FB16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
TADR0	TA0OUT drive capacity	High-drive capacity is activated for the corresponding TAiOUT pin. 0 : Normal drive 1 : High drive	O	O
TADR1	TA1OUT drive capacity		O	O
TADR2	TA2OUT drive capacity		O	O
TADR3	TA3OUT drive capacity		O	O
TADR4	TA4OUT drive capacity		O	O
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are "0".			—	—

Figure 1.112: Port 2 and Timer A Output drive capacity registers



Note: This is an example when the DC-DC converter is disabled

Figure 1.113: Example connection unused pins

Table 1.36: Example connection of unused pins in single-chip mode

Pin name	Connection
Ports P0 to P3, P6 to P8, P10 (excluding $P8_3$, $P8_5$, $P8_6$)	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
X_{OUT}	Open (When using external clock)
$\overline{NMI}$	Connect via resistor to Vcc (pull-up)
AVcc	Connect to Vcc
Avss, Vref, BYTE	Connect to Vss
USB D+, USB D-	Open
EXTCAP	Connect to Vcc (when DC-DC converter is disabled)
$P8_6/\overline{SOF}$	After setting for output mode in normal operation, leave this pin open
$P8_3$/ATTACH	After setting for output mode in normal operation, leave this pin open

1.3 Usage Precautions

1.3.1 Precautions

1.3.1.1 A-D Converter

Connect a capacitor between: the V_{REF} pin and the AVss pin; AVcc pin and AVss pin; and each analog input pin and AVss pin.

- 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 V_{REF} connection bit is changed from “0” to “1”, start A-D conversion after a lapse of 1µs or longer.
- When changing A-D operation mode, select analog input pin again.
- Using one-shot mode or single sweep mode

  Read the corresponding A-D register after confirming the 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.

1.3.1.2 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 1.114 before use.

  Wiring for the Vpp pin of the One-Time PROM version should be as follows (Vpp pin is also used as the CNVss pin):
  - Make the length of wiring between the Vpp pin and Vss pin or Vcc pin the shortest possible.
  - When the wiring length has to be longer, connect an approximately 5K ohm resistor in series from the Vpp pin to the Vss pin or Vcc pin with the shortest possible wiring. This is because the Vpp pin is the power source input pin for the built-in PROM. When programming in the built-in PROM, the impedance of the Vpp pin is low to allow the electric current for wiring flow into the PROM. Because of this, noise can enter easily. If noise enters the Vpp pin, abnormal instruction codes or data are read from the built-in PROM which may cause a program runaway.

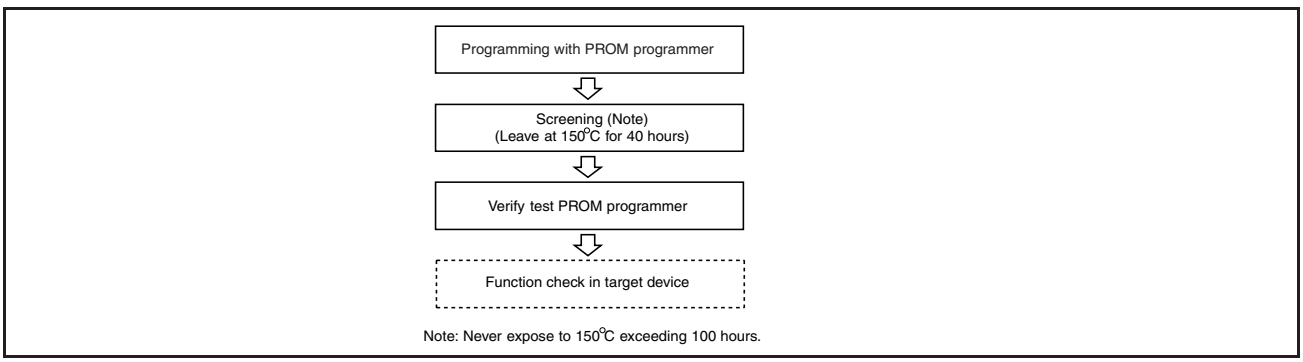


Figure 1.114: Programming and test flow for One-time PROM (OTP) version

1.3.1.3 Dedicated Input Pins

If a dedicated input pin is connected to a power supply different from the supply that Vcc is connected to, a resistor (approximately 1k ohm) should be added between the input pin and the connected power supply. However, if the dedicated input pin voltage is higher than Vcc, latch up could occur.

A resistor is not required when using a Vcc voltage equal or greater than the voltage of the dedicated input pin.

1.3.1.4 DMAC

DMA enable bit

The DMA enable bit is assigned to bit 3 of the DMA0 and DMA1 control registers. DMA becomes active when the DMA enable bit is set to “1”. Immediately after the DMA becomes active, the data transfers start and perform the following operations:

- Reloads the value of the source pointer or destination pointer depending on which is specified for the forward direction address pointer
- Reloads the value of the transfer counter reload register to the transfer counter.

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

DMA request bit

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

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

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

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

To best judge the state of the DMAC, the DMA enable bit should be read instead of the DMA request bit.

1.3.1.5 Frequency Synthesizer

- Refer to Section 1.5.1.2 for setup procedures required after a hardware reset.
- Set the value of the Frequency Synthesizer Prescaler register (FSP) so that f_{PIN} is 1 MHZ or higher.

1.3.1.6 Interrupt

Reading address 00000_{16}

- When a maskable interrupts occurs, the CPU reads the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.
- The interrupt request bit of the corresponding interrupt written in address 00000_{16} is then set to “0”.
- Do not read address 00000_{16} by software.
  - Reading address 00000_{16} by software sets the highest priority enabled interrupt source request bit to “0” Therefore the interrupt routine may not be executed even though the interrupt is generated.

Setting the stack pointer

- The value of the stack pointer is initialized to 00000_{16} immediately after reset. Accepting an interrupt before setting a value in the stack pointer may cause program runaway. Be sure to set a value in the stack pointer before accepting an interrupt.
- When using the $\overline{\text{NMI}}$ interrupt, initialize the stack pointer at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupts including the $\overline{\text{NMI}}$ interrupt is prohibited.

Setting interrupts

- Changing the Interrupt Priority Level select bit (ILVL) and clearing the Interrupt Request bit (IR) in the Interrupt Control Registers (ICR) while the Interrupt enable flag (I-FLAG) is "1", may result in unintended operations, such as BRK and other interrupts being generated. Disable the interrupts by clearing the I-FLAG before setting ILVL or clearing the IR bit.
- To prevent the I-FLAG from being set before the ICR is rewritten due to the effects of the instruction queue, instructions that equal a minimum of 2 cycles should be inserted between writing to the ICR and setting the I-FLAG (2-NOPs, I MOV, I POP, etc.)

Modifying interrupt control registers

- Do not modify any interrupt control register when an interrupt request can be generated.
- If an interrupt request occurs, modify the interrupt control register after the interrupt is disabled.

External interrupts

- When the polarity of the INT0 and INT1 pins is changed, the interrupt request bit can be set to "1". After changing the polarity, the interrupt request bit should be cleared to "0."

The $\overline{\text{NMI}}$ interrupt

- As for the $\overline{\text{NMI}}$ interrupt pin, an interrupt cannot be prohibited. Connect it to the Vcc pin if unused.
- Do not get into stop mode or wait mode with the $\overline{\text{NMI}}$ pin set to "L".

1.3.1.7 Noise

To reduce the possibility of noise problems:

- Connect a bypass capacitor (approximately 0.1 μF) across the Vss pin and the Vcc pin with the shortest possible wiring
- Use circuit traces with a larger diameter than other signal traces for Vss and Vcc.

Vpp connection of One-time PROM version

- The Vpp (power input for PROM programming) connection for the internal PROM is connected to the CNVss pin on the One-time PROM version. Therefore, CNVss should be a short circuit trace to improve noise resistance. If the CNVss trace is long, insert a 5kΩ resistor close to the CNVss pin and connect it to Vss.
- Note: Inserting a 5kΩ resistor will not cause any problem when switching to a mask ROM version.

1.3.1.8 Software reset

Software reset with f_{SYN} is selected as a clock source. If the Frequency Synthesizer related registers are written to while f_{SYN} is selected as the clock source, the MCU will runaway. To avoid this:

1. Select f(XIN) as the clock source before the software reset
2. Perform software reset
3. Confirm software reset using Frequency Synthesizer Enable bit
4. Reselect f_{SYN} as clock source.

1.3.1.9 Stop Mode and Wait Mode

When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to “L” level until main clock oscillation is stabilized.

When entering either wait or stop mode, you must first enable any interrupts you want to cancel the wait or stop. Also, make sure to disable any interrupts that you don’t want to cancel the wait or stop. If only hardware reset or NMI interrupts are desired to cancel wait or stop, all other interrupt priority levels should be set to “0”

If using f_{SYN} as the internal clock, switch it to $f(X_{IN})$ before entering stop mode.

When switching to either wait or stop mode, the WAIT instruction or the instruction that sets the all clock stop bit to “1” are prefetched within the instruction queue before the program stops. Put at least four NOPs in succession after the WAIT instruction or after the instruction that sets the all clock stop control bit to “1.”

1.3.1.10 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 “$FFFF_{16}$”. 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.

1.3.1.11 Timer A (Event counter mode)

1. 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 “$FFFF_{16}$” by underflow or “0000_{16}” 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.

2. When counting has stopped in free-run type, set the timer again.

3. When using Free-run type, the timer’s register contents may be undefined when counting starts. Set the timer value immediately after counting has started.

- When the up/down count is not switched:
  - •Enable the reload function and set a value to the timer register before counting starts.
  - •Rewrite the value to the timer register immediately after counting has started. (This is the same operation as free-run type.)
    - •If counting up, rewrite $0000_{16.}$
    - •If counting down, rewrite $FFFF_{16.}$
- When the up/down count is switched:
  - •Use the reload type until the first count pulse is input.
  - •Switch to free-run type afterwards.

1.3.1.12 Timer A (One-shot Timer mode)

1. Setting the count start flag to “0” while the count is in progress causes:
   - The counter to stop counting
   - The contents of the reload register are reloaded
   - The TAiOUT pin outputs “L” level
   - The interrupt request is generated
2. Timer Ai interrupt request bit goes to “1” if the operation mode is set by:
   - Selecting one-shot timer mode after reset
   - Changing operation mode from timer mode to one-shot timer mode
   - Changing operation mode from event counter mode to one-shot timer mode

Note: To use Timer Ai interrupt request bit, clear the Timer Ai interrupt request bit to “0” after these changes have been made.

1.3.1.13 Timer A (Pulse-width Modulation mode)

1. 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.

2. Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the TAi_{OUT} 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 TAi_{OUT} pin is outputting an “L” level in this instance, the level does not change, and the Timer Ai interrupt request bit does not become “1”.

1.3.1.14 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 “$FFFF_{16}$”. 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.

1.3.1.15 UART2

When using UART2 in clock asynchronous serial I/O mode (UART), use the internal clock only, otherwise, one of the following may occur:

- The interrupt may not be issued at the end of the data transmission when the hardware transfers the data from the transmit buffer to the transmit register.
- Data may be corrupted when the hardware transfers data fro the transmit buffer register to the transmit register. This only applies to UART2 asynchronous serial I/O mode and does not apply to UART0 or UART1.

1.3.1.16 USB

- When the USB Reset Interrupt Status flag is set to “1”, the contents in the USB internal register (addresses 0300_{16}-0335_{16}) will return to their reset values. However, the following registers are not affected by a USB Reset:
  - USB Control register (USBC)
  - Frequency Synthesizer Control register (FSC)
  - USB Endpoint x FIFO (addresses 0338_{16} -$033C_{16}$)
- All LPF pin passive components must be located as close as possible to the LPF pin.
- An insulation connector (Ferrite beads) must be connected between the AVss and digital Vss pins, and between AVcc and digital Vcc pins.
- When using DC-DC converter to supply 3.3V to the drive, connect a capacitor between the EXTCAP pin and the Vss pin. The connection should consist of a 2.2µF capacitor (tantalum capacitor) and a 0.1µF capacitor (ceramic capacitor) connected in parallel. Use a ceramic capacitor equivalent to the X7R type as a 0.1 µF capacitor.
- Connect a 27-33Ω resistor between the USB D+ and USB D- pins to meet USB specification impedance requirements.
- Connect a ceramic capacitor (33pF recommended) after the resistor between USB D+ and USB D- or between USB D+/D- pins and the Vss pin to control the slew rate and rise/fall timing. This cap should be placed after the 27-33Ω resistor. See section 1.5 for more details.
- Connect a 1.5kΩ resistor between the EXTCAP pin and the USB D+ pin during normal operation.
- Connect a 1.5kΩ resistor between the $P8_3$/ATTACH pin and the USB D+ pin and leave the EXTCAP open when using Attach/Detach function.
- Read or write to the USB internal registers (address 0300_{16}-$033C_{16}$) by 8-bit mode only. Accessing by 16-bit mode will cause incorrect read/write values.

- When using an isochronous transfer, set the FLUSH bit by:
  - OUT FIFO data flush: When OUT_PKT_RDY flag is “1”, set the FLUSH bit to “1”
  - IN FIFO data flush: Use AUTO_FLUSH bit.
- Do not write to the USB internal registers (address 0300_{16}-$033C_{16}$) when the USB clock is disabled in Suspend mode.
- Precautions for accessing the USB Interrupt Status registers 1 & 2 (USBIS1, USBIS2)
  - When reading from/writing to both registers, access USBIS1 first and then USBIS2.
  - When writing to these registers, use transfer instructions such as the MOV instruction. Do not use Read Modify Write instructions such as OR or BSET because this could cause improper values to be written back.
  - Each status flag can be cleared to”0” by writing back the same value “1” which was read from the USB Interrupt Status register. Make sure to clear only the corresponding status flag in each USB interrupt routine. Mask the other status flags so that they are not accidentally cleared.
  - Example of USB Function Interrupt

```
C Language
ram1=USBIS1;        /*Read from USB Interrupt Status register 1*/
ram2=USBIS2;        /*Read from USB Interrupt Status register 2*/
ram2 &=0x1F;        /*Mask flags except USB Function Interrupt status flags*/
USBIS1=ram1;        /*Write to USB Interrupt Status register 1*/
USBIS2=ram2;        /*Write to USB Interrupt Status register 2*/

Assembly Language:
mov.b USBIS1, ram1  ;Read from USB Interrupt Status register 1
mov.b USBIS2, ram2  ;Read from USB Interrupt Status register 2
and.b #1Fh, ram2    ;Mask flags except USB Function Interrupt status flags
mov.b ram1, USBIS1  ;Write to USB Interrupt Status register 1
mov.b ram2, USBIS2  ;Write to USB Interrupt Status register 2
```

- Do not use Read, Modify, Write instructions on the following USB Control and Status registers:
  - USB Endpoint 0 Control and Status (EP0CS)
  - USB Endpoint x IN Control and Status (EPxICS)
  - USB Endpoint x OUT Control and Status (EPxOCS)
  - Refer to Section 1.5.5.1 for more information on these registers and how to modify their contents.

Items to be submitted when ordering masked ROM version

Please submit the following when ordering masked ROM products:

(1) Mask ROM confirmation form

(2) Mark specification sheet

(3) ROM data: EPROMs or floppy disks

*: In the case of EPROMs, there are sets of EPROMs are required per pattern.

*: In the case if floppy disks, 3.5-inch double-sided high-density disk (IBM format) is required per pattern.

1.4 Specifications

1.4.1 Electrical characteristics

Table 1.37: Absolute maximum ratings only, not operating conditions

Symbol	Parameter		Condition	Rated Value	Unit
V_{cc}	Supply voltage		Vcc=AVcc	-0.3 to 6.5	V
AV_{cc}	Analog supply voltage		VCC=AVCC	-0.3 to 6.5	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 6.5 (Note 1)	V
V_O	Output voltage	Port0, Port1, Port2, Port3, Port6, Port7, Port8 (except P85), Port10, RESET, VREF, XOUT		-0.3 to Vcc+0.3	V
P_d	Power dissipation		Ta=25°C	760	mW
T_{opr}	Operating ambient temperature			0 to 70	°C
T_{stg}	Storage temperature			-65 to 150	°C

Note 1: When writing to EPROM, CNVss rated value is -0.3 to 13 volts

Table 1.38: Recommended operating conditions (Vcc=4.1~5.25V, Vss=0V, Ta= 0°C~ 70°C, f(Xin) = 12MHz)

Symbol	Parameter		Standard			Unit
			Min	Typ	Max	
V_{cc}	Supply voltage		4.1	5.0	5.25	V
AV_{cc}	Analog supply voltage			Vcc		V
Vss	Supply voltage			0		V
Avss	Analog supply voltage			0		V
V_{IH}	High input voltage	Port0, 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
$f(X_{IN})$	Main clock input oscillation frequency		1		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 1.39: Electrical Characteristics (Vcc=4.1~5.25V, Vss=0V, Ta= 0°C~ 70°C, f(Xin) = 12MHz)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min	Typ	Max	
VOH	High output voltage	Port0, Port1, Port2, Port3, Port6, Port71, P73,P75,P77,Port8 (except P85), Port10	IOH = -5mA, Vcc=5V	3.0			V
VOH	High output voltage	Port 70,P72,P74,P76,P80	IOH = -10mA, Vcc=5V	3.0			V
VOH	High output voltage	Port0, Port1, Port2, Port3, Port6, Port71, P73,P75,P77,Port8 (except P85), Port10	IOH = -200µA, Vcc=5V	4.7			V
VOH	High output voltage	Xout high power	IOH = -1mA, Vcc=5V	3.0			V
		Xout low power	IOH = -0.5mA, Vcc=5V	3.0			V
VOL	Low output voltage	Port0, Port1, Port2, Port3, Port6, Port71, P73,P75,P77,Port8 (except P85), Port10	IOL = 5mA, Vcc=5V			2.0	V
VOL	Low output voltage	High-drive mode Port 2	IOL = 10mA, Vcc=5V			2.0	V
VOL	Low output voltage	Port 70,P72,P74,P76,P80 NOTE 1	IOL= 10mA, Vcc=5V			2.0	V
VOL	Low output voltage	Port0, Port1, Port2, Port3, Port6, Port71, P73,P75,P77,Port8 (except P85), Port10	IOL = 200µA, Vcc=5V			0.45	V
VOL	Low output voltage	Xout high power	IOH = 1mA, Vcc=5V			2.0	V
		Xout low power	IOH = 0.5mA, Vcc=5V			2.0	V
VT+-VT-	Hysteresis	TA0in to TA4in, INT0 to INT1, ADTRG, CTS0, CTS1, CLK0, CLK1, TA2out to TA4out, NMI, KI0 to KI15	Vcc=5V	0.2		0.8	V
VT+-VT-	Hysteresis	RESET	Vcc=5V	0.2		1.8	V
Iih	High input current	Port0, Port1, Port2, Port3, Port6, Port7,Port8, Port10, RESET, CNVss	VI = 5V, Vcc=5V			5.0	µA
Iil	Low input current	Port0, Port1, Port2, Port3, Port6, Port7, Port8, Port10, RESET, CNVss	VI = 0V			-5.0	µA
RPULLUP	Pull-up resistance	Port0, Port1, Port2, Port3, Port6, Port7, Port8, Port10	VI = 0V, Vcc=5V	30	50	167	kΩ
RXIN	Feedback resistance, Xin				1.0		MΩ
VRAM	RAM retention voltage		When clock is stopped	2.0			V
Icc	Power supply current		Output pins open, other pins tied to Vss; Icc run with USB ON (Mask)			80	mA
			Icc run with USB ON (OTP)			95	mA
			Icc run with USB OFF			50	mA
			Ta=25°C clock stopped			1	µA
			Ta=70°C clock stopped			20	µA
			Ta=25°C wait mode with internal clocks ON			8	mA
			Ta=25°C wait mode with internal clocks OFF			4	mA

Note 1: Only high drive when Timer A is enabled and drive registers set for high drive mode.

Table 1.40: USB Electrical Characteristics (Vcc=4.1~5.25V, Vss=0V, Ta= 0°C~ 70°C, f(Xin) = 12MHz)

Symbol	Parameter	Measuring Condition	Standard			Unit
			Min	Typ	Max	
VOH	D+, D-	I=18.3 mA, RX=33 Ω, VXcap =3.0 V	2.2			V
VOL	D+, D-	I=18.3 mA, RX=33 Ω, VXcap =3.0 V			0.8	V
Isusp	Suspend current	USB suspend mode, internal clock stopped			175	µA
Xcap	DC-DC converter voltage	DC-DC converter output voltage on Xcap pin	3.0	3.3	3.6	V

Note: See Fig. 1.120 for recommended configuration.

Table 1.41: A-D conversion characteristics (Vcc, Avcc = 4.1~5.25V, Vss=0V, Ta= 0°C~ 70°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
R_{LADDER}	Ladder resistance		VREF = Vcc = 5V	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 (min. operating frequency =x)			0		VREF	V
ϕAD	A-D clock frequency			1		12	MHz

1.4.2 Timing

Timing requirements referenced to Vcc = 4.1~5.25V, Vss=0V, Ta= 0°C~70°C unless otherwise specified.

Table 1.42: 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 1.43: Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(TA)	TAiIN input cycle time	100		ns
tw(TAH)	TAiIN input HIGH pulse width	40		ns
tw(TAL)	TAiIN input LOW pulse width	40		ns

Table 1.44: Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(TA)	TAiIN input cycle time	400		ns
tw(TAH)	TAiIN input HIGH pulse width	200		ns
tw(TAL)	TAiIN input LOW pulse width	200		ns

Table 1.45: Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(TA)	TAiIN input cycle time	200		ns
tw(TAH)	TAiIN input HIGH pulse width	100		ns
tw(TAL)	TAiIN input LOW pulse width	100		ns

Table 1.46: Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tw(TAH)	TAiIN input HIGH pulse width	100		ns
tw(TAL)	TAiIN input LOW pulse width	100		ns

Table 1.47: Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(UP)	TAiOUT input cycle time	2000		ns
tw(UPH)	TAiOUT input HIGH pulse width	1000		ns
tw(UPL)	TAiOUT input LOW pulse width	1000		ns
tsu(UP-TIN)	TAiOUT input setup time	400		ns
th(TIN-UP)	TAiOUT input hold time	400		ns

Table 1.48: A-D trigger input

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(AD)	$\overline{AD}_{\overline{TRG}}$ input cycle time (triggerable minimum)	1000		ns
tw(ADL)	$\overline{AD}_{\overline{TRG}}$ input LOW pulse width	125		ns

Table 1.49: Serial I/O

Symbol	Parameter	Standard		Unit
		Min	Max	
tc(CK)	CLKi input cycle time	200		ns
tw(CKH)	CLKi input HIGH pulse width	100		ns
tw(CKL)	CLKi input LOW pulse width	100		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 1.50: External interrupt INTi inputs

Symbol	Parameter	Standard		Unit
		Min	Max	
tw(INH)	INTi input HIGH pulse width	250		ns
tw(INL)	INTi input LOW pulse width	250		ns

1.4.3 Timing Diagram

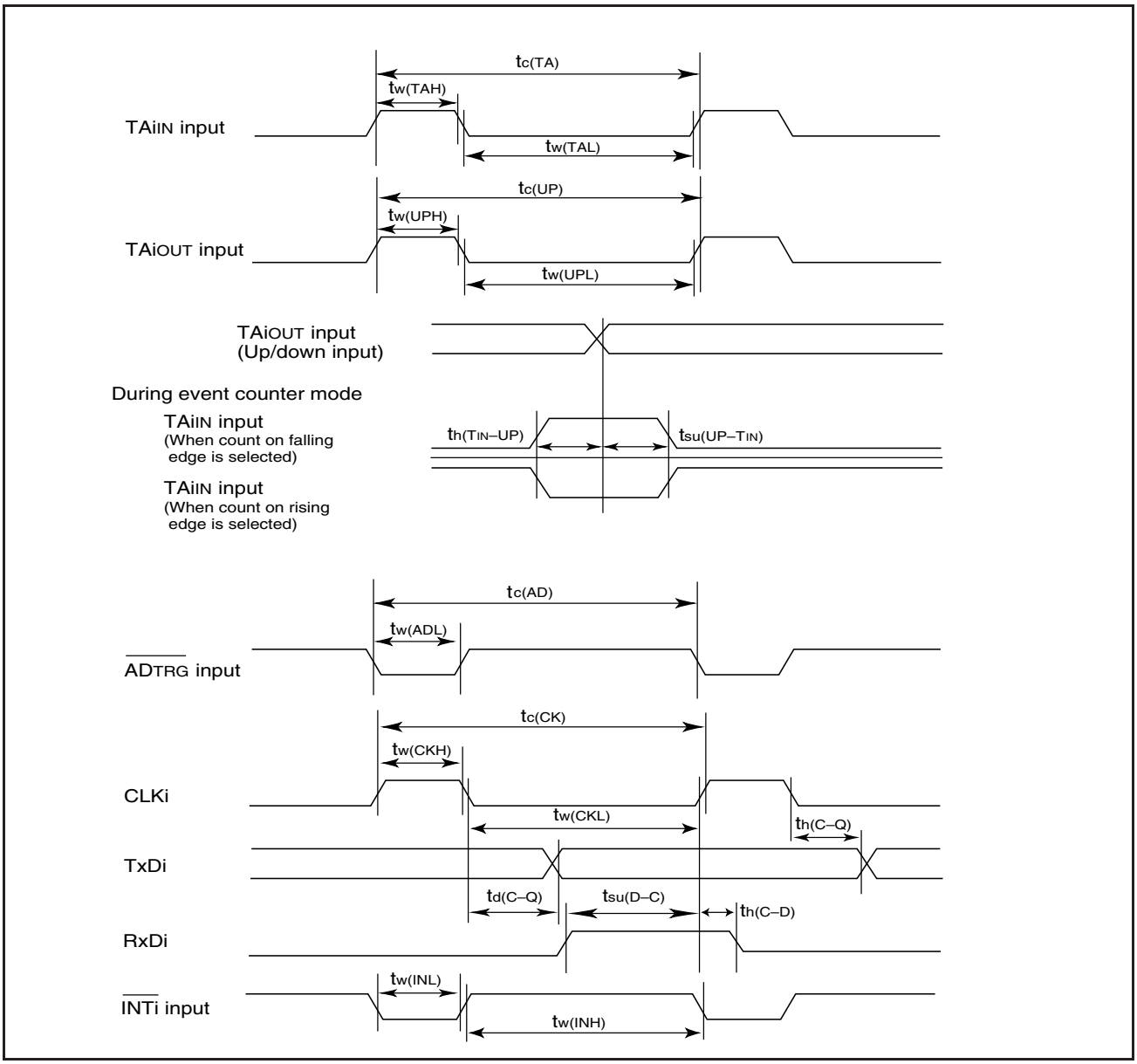


Figure 1.115: Peripheral/interrupt timing diagram

1.5 Applications

1.5.1 Frequency Synthesizer

This section presents the recommended method of setting up and using the frequency synthesizer that generates the 48MHz clock needed by the USB FCU and the DC-DC converter that provides power to the D+/D- drivers.

1.5.1.1 Reset of USB-related registers

The special function registers (SFRs) that govern the operation of the frequency synthesizer, DC-DC converter and USB FCU are affected by one or more reset events. The addresses of the special function registers (SFRs) that are affected by Hardware Reset, USB Reset, or both are shown in Figure 1.116.

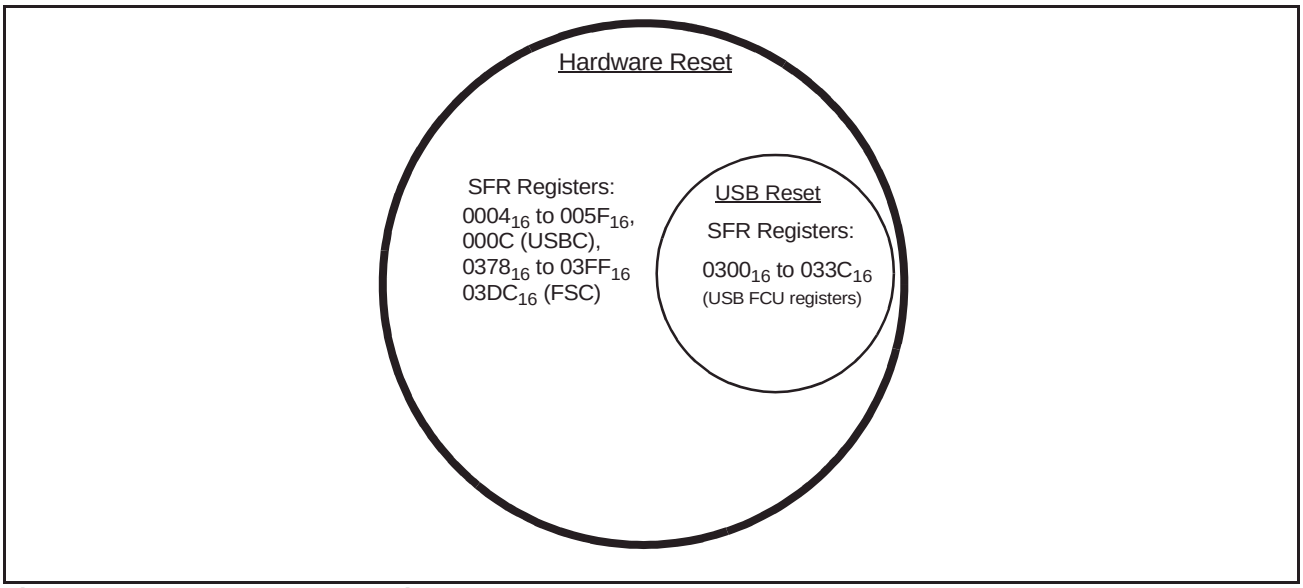


Figure 1.116: SFR Reset Venn Diagram

All resetable SFRs, including SFRs and other registers internal to the USB FCU, are affected by a Hardware Reset, which occurs when the RESET pin is brought low or an undefined opcode is fetched. See Section 2.4 for a complete listing of SFRs and their reset values.

Only registers internal to the USB FCU are reset when a USB Reset sent by the Host/Hub is detected. These USB registers are reset to their default values except for bit 5 of USBIS2 (USB Reset Interrupt Status Flag), which is set to a “1”. USB FCU registers are registers from address $0300_{16}$to $033C_{16}$ and all other registers within the USB FCU, many of which the MCU does not have direct access to (e.g. FIFO address pointers). The USB FIFO registers are empty after USB reset because the FIFO address pointers are reset. However, the physical contents of the FIFOs are not set to all ‘1‘s or all ‘0‘s. Other SFRs such as USBC, FSC, and CM0, CM1 are not affected by a USB Reset.

1.5.1.2 Set up of Frequency Synthesizer and DC-DC Converter

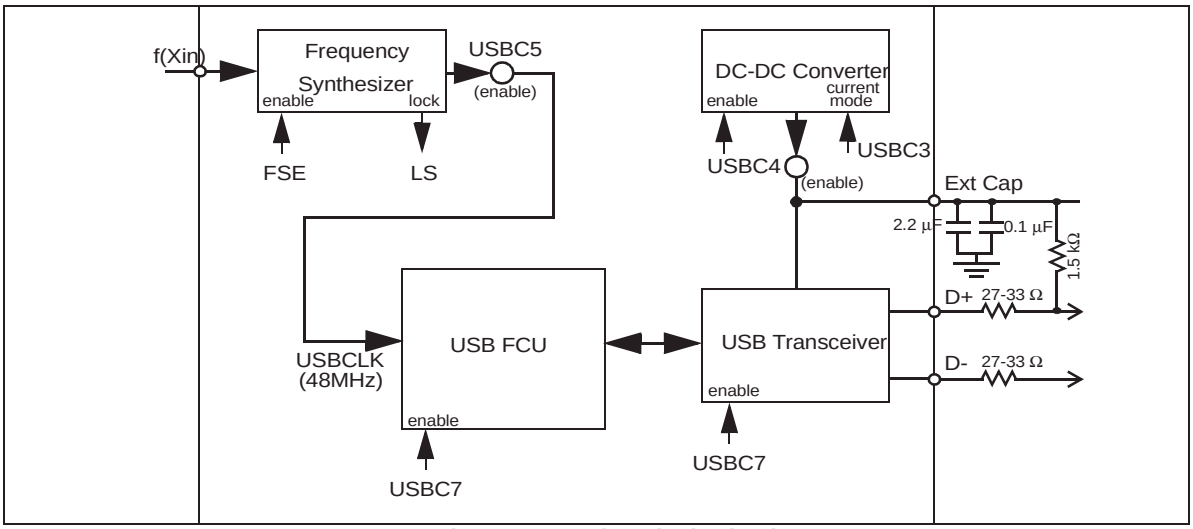


Figure 1.117: PLL, DC-DC Converter and USB Functional Block Diagram

A functional block diagram of the USB system on the M30240 which shows how the control signals affect operation is given in Figure 1.117

1.5.1.3 Set up after Hardware Reset

A Hardware Reset occurs when either the $\overline{\text{RESET}}$ pin is brought low for more than 2 µs or an invalid opcode is fetched by the CPU. The frequency synthesizer (PLL) and DC-DC converter should be set up as follows in the Hardware Reset routine (see Figure 1.118),

- Power up the M30240 and other components on the peripheral device for less than 100 mA operation. The current limit only applies for bus powered devices.
- Configure the PLL for 48MHz f(VCO) operation.
- Enable the PLL by setting FSE (bit 0 of the Frequency Synthesizer Control Register (FSC)) to a “1”, then wait for 2 ms.
- Check the lock status bit (LS, bit 7 of FSC).
  - If the bit is a “1”, go on.
  - If the bit is a “0”, wait 0.1 ms longer and then re-check the bit.
- Enable the DC-DC converter in high current mode by setting USBC4 (bit 4 of the USB Control Register (USBC)) to a “1” and keeping USBC3 (bit 3 of USBC) a “0”. High current mode should always be used during normal USB operation. Low current mode should only be used during a USB suspend.
- Wait (C + 1)ms (where C equals the external capacitance connected to the Ext Cap pin in µF) for the voltage on Ext Cap to reach a steady state voltage of approximately 3.3V. (Since the D+ pullup is connected to the Ext Cap pin, the upstream hub will detect that the peripheral device has been plugged in once the voltage on D+ reaches approximately 2.0 V.)
  - Example: A 2.2 µF capacitor connected to Ext Cap requires 3.2 ms for the voltage on Ext Cap to be stable.
- Enable the USB clock by setting USBC5 (bit 5 of USBC) to a “1”. (If the USB clock and FCU are enabled before the voltage on Ext Cap is stable, a phantom USB Reset may be detected, or the actual USB Reset may not be detected.)
- Wait at least 4 cycles of Φ, then enable the USB FCU by setting USBC7 (bit 7 of USBC) to a “1”.
- Enable other blocks as necessary.

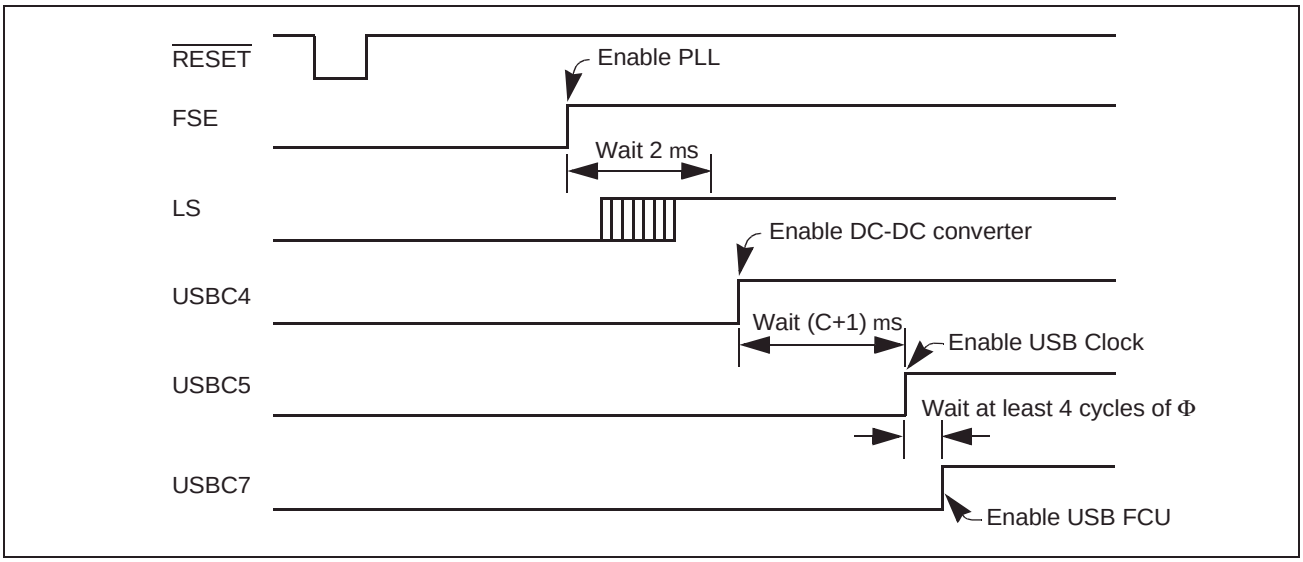


Figure 1.118: PLL and DC-DC Converter Set Up Timing after Hardware Reset

1.5.1.3.1 Precautions after Software Reset

A software reset occurs after writing a '1' to bit '3' of the processor mode register 0 (address 0004_{16}). During software reset, the contents of the internal RAM are preserved as well as all USB, DC-DC converter, and PLL registers. If the PLL is used as the system clock source, it is important to note that after a software reset occurs, any writes to the frequency synthesizer register will cause it to freeze. This can cause erratic device behavior. In order to avoid this, it is recommended that the following procedure be used:

- Prior to software reset, switch device clock source from 'fsyn to f(Xin)'. Please see the Frequency Synthesizer specification for more details.
- After software reset using firmware, evaluate the condition of the synthesizer control register (FSC register, address $03DC_{16}$, bit '0'). This bit is not effected by a software reset and can check to see if the PLL is still enabled. If so, any setup routine that involves writing to the PLL registers should not be called. At this point, the clock source can be changed back to fsyn.

1.5.1.4 Set up after USB Reset Signaling Detected

A USB Reset is detected by the USB FCU when an SE0 is present on D+/D- for at least 2.5 µs. Detection of a USB Reset results in bit 5 of USB Interrupt Status Register 2 (USBIS2) being set to a "1" and the registers within the USB FCU being reset to their default values. Register USBC and the PLL registers are not affected by a USB Reset. A USB Function Interrupt request is also generated when the USB Reset is detected.

No modifications to the frequency synthesizer or DC-DC converter configuration should be made in the USB Function Interrupt routine. However, all USB FCU registers (addresses 300_{16} to $33C_{16}$) must be reconfigured to their pre-enumeration state.

1.5.1.5 Set up after USB Suspend Detected

A USB Suspend occurs if the USB FCU does not detect any bus activity on D+/D- for at least 3 ms. Detection of a suspend results in bit 7 of USBIS2 and bit 0 of USBPM (SUSPEND) being set to a "1". This causes bit 3 of SUSPIC to be set to a "1". Bit 7 of USBIS2 then needs to be cleared by writing a "1" to the bit in order to allow a future suspend event.

The configuration of the frequency synthesizer and DC-DC converter should be changed as follows in the USB Suspend Interrupt routine (if the device is bus powered):

- Change the DC-DC converter from high current mode to low current mode by setting USBC3 (bit 3 of the USBC) to a “1”
- Disable the USB clock by setting USBC5 (bit 5 of USBC) to a “0”. Once the USB clock is disabled, registers internal to the USB FCU should not be written to. This includes all USB SFRs from address 0300_{16} to $033C_{16}$. It does not include USBC or FSC.
- Perform other tasks to reduce total current to below 500μA.
- Disable the PLL by setting FSE (bit 0 of FSC) to a “0”.
- Make sure the I-FLAG is set to “1”.
- Stop the system clock by setting CM10 (bit 0 of CM1) to a “1”. Make sure to first enable writing to the system clock control register by setting PRCO (bit 0 of PRCR) to “1’. Also, make sure to enable the USB Resume Interrupt (RSMIC register) and clear or execute any pending interrupts prior to stopping the clock so the MCU can wake up once resume signaling is detected. If the clock is stopped using an interrupt routine, make sure to set the priority of the Resume Interrupt (RSMIC) higher than the current interrupt.
- Note that no action may be necessary if the device is self powered.

1.5.1.6 Set up after USB Resume Signaling Detected

A resume occurs when the USB FCU is in the suspend state and detects a non-idle signaling on D+/D-. Detection of a resume results in bit 6 of USBIS2 and bit 1 of USBPM (RESUME) being set to a “1”. This causes bit 3 of RSMIC to also be set to “1”. If the MCU was in the stop state prior to the detection of the resume, the USB Resume Interrupt request will cause the MCU to wake up from the stop state. Bit 6 of USBIS2 needs to be cleared (by writing a “1” to the bit) in order to allow a future resume event. See section 2.9 “Stop Mode” for details on waking up from the stop state.

The configuration of the frequency synthesizer and DC-DC converter should be changed as follows in the USB Resume Interrupt routine (if the device is bus powered):

- Re-enable the PLL for 48MHz f(VCO) by setting FSE (bit 0 of the FSC) to a “1”, then wait for 2 ms.
- Wait for 2 ms.
- Check the lock status bit (LS, bit 7 of FSC).
  - If the bit is a “1”, continue.
  - If the bit is a “0”, wait 0.1 ms longer and then re-check the bit.
- Enable the USB clock by setting USBC5 (bit 5 of USBC) to a “1”.
- Wait for a minimum of 4 cycles.
- Change the DC-DC converter from low current mode to high current mode by setting USBC3 (bit 3 of the USBC) to a “0”.
- Enable other blocks as necessary.

Registers internal to the USB FCU should not be written to until the USB clock is re-enabled. This includes all USB SFRs from address 0300_{16} to $033C_{16}$. It does not include USBC or FSC.

Note that the configuration changes described above may not need to be made if the MCU was not placed in a suspend state as described in section 5.1.2.3 Set up after USB Suspend Detected.

1.5.1.7 PLL Lock Bit

The PLL lock bit is used to indicate when the PLL is first locked. Accordingly, after the PLL is enabled and it has been given 2.0 ms to stabilize, the lock bit status should be checked. Once the lock bit is HIGH, the USB check should be enabled. After this stage, the lock bit is no longer valid and should not be monitored, unless the PLL is re-enabled.

1.5.2 Attach/Detach Function

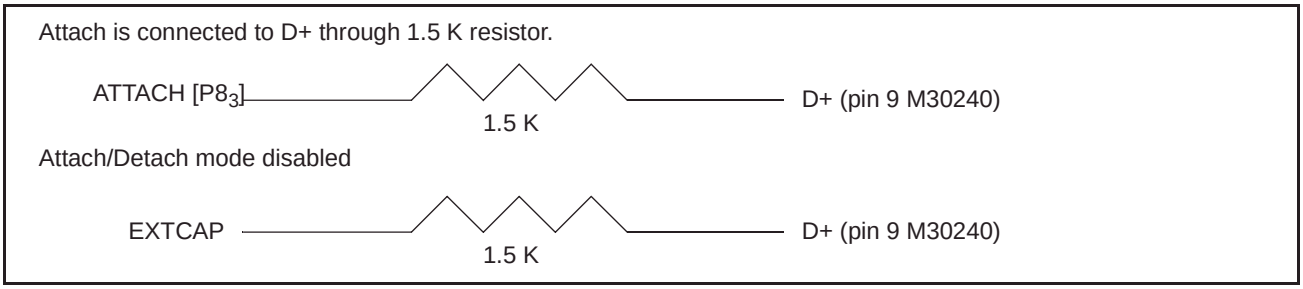
The Attach/Detach Function can be used to attach or detach a USB function from the host without disconnecting the cable. When attaching a USB function, the connect registers should be set to 03_{16} at the same time on or before the DC-DC Converter is enabled. Similarly, when detaching the connect register, it should be set to 01_{16} when powering down the DC-DC Converter.

If you do not set the connect (address $001F_{16}$) to HIGH, the system will default to its normal mode.

Note: If the D+ is connected to EXTCAP, this mode will not work.

D+ is connected to EXTCAP through a 1.5 K resistor in compliance with the USB specification. USB Suspend/Resume Function

Hardware connections are shown below



1.5.3 Low Pass Filter Network

All passive components should be in close proximity to pin 78 (LPF), capacitors should be X7R dielectric or better. The recommended values are listed in Table 1.51. See Figure 1.119 for schematic of the LPF.

Analog V_{ss} and Analog V_{cc}, pins 77 and 80 should have isolated connections to the digital V_{ss} and V_{cc} ground planes. Figure 1.120 shows the power supply isolation.

Table 1.51: Recommended values

R = 1000 Ω	**10%**
C2 = 680 pf	**10%**
C1 = 0.1 μf	**10%**

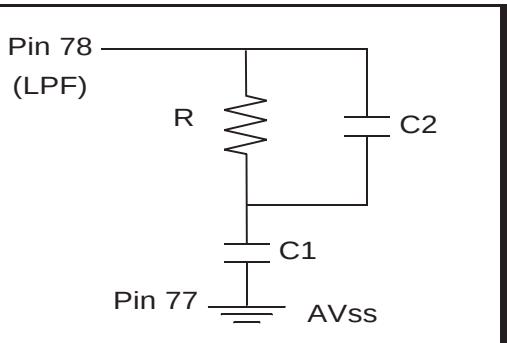


Figure 1.119

Figure 1.119: LPF Filter Schematic

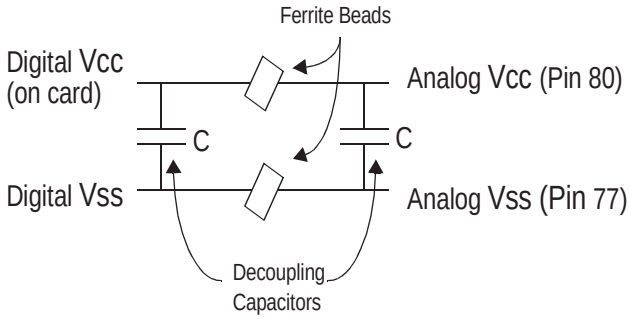


Figure 1.120

Figure 1.120: Power Supply

1.5.4 USB Transceiver

When using the on-chip voltage converter to supply the necessary 3.3V to the driver circuit, a capacitor network must be connected between Ext. Cap (pin 6) and V_{SS} (pin 13). Two capacitors are required as shown in Figure 1.21. The high frequency 0.1 µF capacitor should be an X7R type or better. The low frequency decoupling capacitor of 2.2 µF should be of tantalum di-electric or better. The start-up time for this value of the capacitor is 3.2 ms, approximately (1ms/ µF) + 1 ms.

After enabling the on-chip voltage converter, a certain amount of time must pass before a wait or stop clock instruction is executed. The amount of time is given by (C+1) ms, when C is the value in µF of the external capacitance connected to the Ext. Cap pin. For example, if the external capacitance is 2.2 µF, at least 3.2 ms must elapse from the time that the on-chip voltage converter is enabled until a WAIT instruction or STOP command (CM10 = 1) is executed.

In order to meet the impedance matching requirements of the USB Specification, a 27Ω-33Ω resistor must be added to USB D+ (pin 9) and to USB D- (pin 10). In addition, capacitors connected between USB D+ and USB D- or USB D+/D- and Vss may need to be added for rise/fall time matching and edge control. These capacitors should be placed after the 33 Ω resistors. Their configuration and values will depend on the PCs layout. The placement of external components is illustrated in Figure .

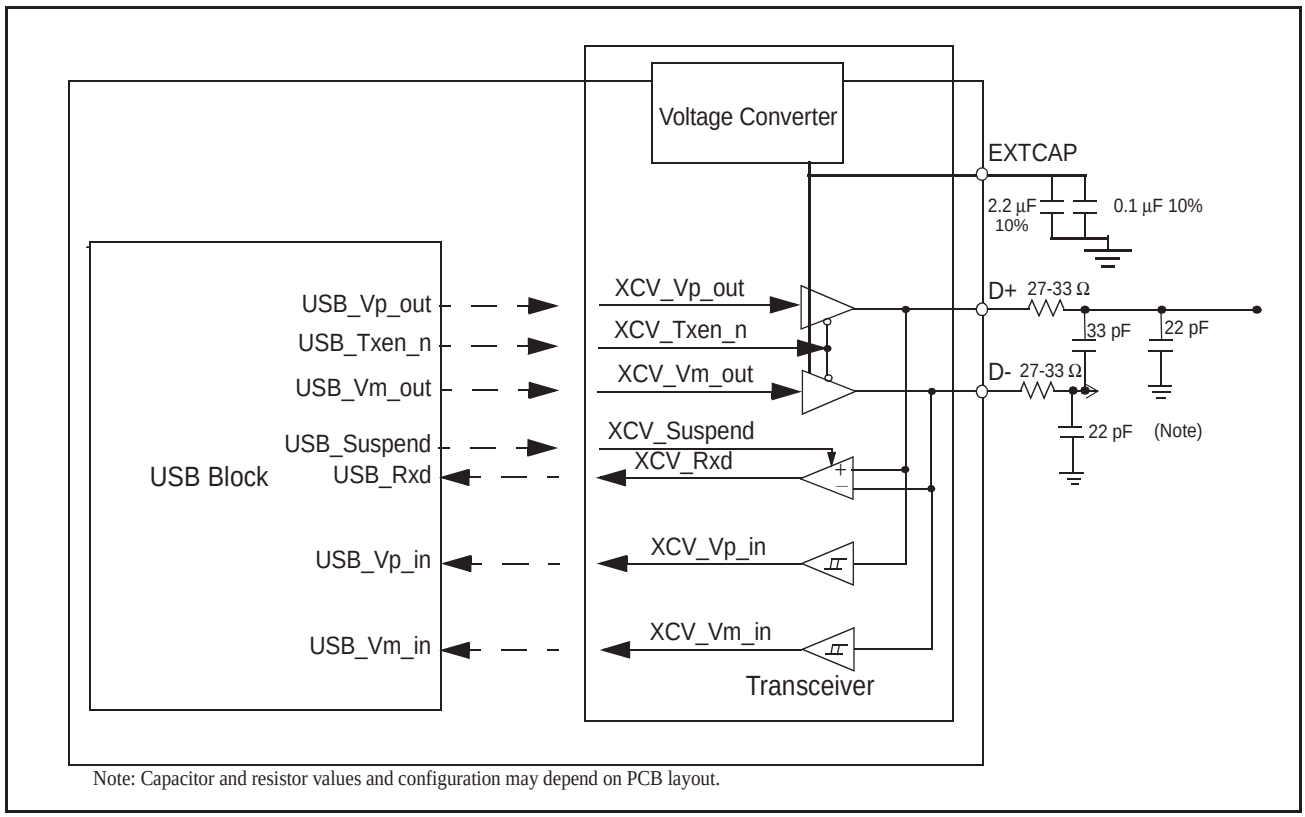


Figure 1.121: Configuration of External USB components

1.5.5 Programming Notes

1.5.5.1 Accessing USB IN/OUT Control and Status Registers

Do not use read-modify-write instruction on these registers because they contain control and status bits that can be changed by both hardware and software. There is a possibility that using a read-modify-write instruction might cause incorrect data to be written back to these registers. See Table 1.52 for a list of bits that may have incorrect data written to them and the value you should write back in order to prevent this from occurring.

Table 1.52: Bits that might have incorrect data

Register name	Bit name	Value to write for “No change”
EP0CS	IN_PKT_RDY (b1)	“0”
	DATA_END (b3)	“0”
	FORCE_STALL (b4)	“1”
EPxICS (x = 1-4)	IN_PKT_RDY (b0)	“0”
	UNDER_RUN (b1)	“1”
EPxOCS (x = 1-4)	OUT_PKT_RDY (b0)	“1”
	OVER_RUN (b1)	“1”
	FORCE_STALL (b4)	“1”
	DATA-ERR (b5)	“1”

The Endpoint 1-4 IN CSR’s (EPiICS, i = 1-4) have a bit IN_PKT_RDY (bit 0) that is set to a “1” by the firmware after a packet of data is loaded to the respective endpoint’s FIFO. This signifies that a packet is ready for transmission. If the firmware wants to send a NULL packet to the host, it can simply write a “1” to the IN_PKT_RDY bit without loading data to the FIFO. This bit is cleared by the hardware. If the firmware manipulates (writes) the IN CSR for a purpose other than to signify to the hardware that a data packet is ready for transmission (for instance, set/reset ISO bit, set/reset SEND_STALL bit), it must make sure that a “0” is written back to the IN_PKT_RDY bit. Failure to do so could cause improper operation of the device. Writing a “0” to the IN_PKT_RDY bit has no effect on its state.

The Endpoint 1-4 OUT CSRs (EPiICS, i = 1-4) have a bit OUT_PKT_RDY (bit 0) that is set to a “1” by the hardware after a packet of data is received from the host to the respective endpoint’s FIFO. This signifies that a packet is ready for download. This bit is cleared by the firmware by writing a “0” to it after the data packet is unloaded from the FIFO. If the firmware manipulates (writes) the OUT CSR for a purpose other than to signify to the hardware that a data packet has been unloaded (for instance, set/reset ISO bit, set/reset SEND_STALL bit), it must make sure that a “1” is written back to the OUT_PKT_RDY bit. Failure to do so could cause improper operation. Writing a “1” to the OUT_PKT_RDY bit has no effect on its state.

Below is an example of how to set/reset the ISO bit of the IN CSR register (for initializing the respective endpoint as an isochronous endpoint):

```
[R1L] = [EPiICS].B
OR.B #0AH, R1L          ;set ISO bit = 1, write “1” back to UNDER_RUN bit
AND.B #0FEH, R1L        ;write “0” back to IN_PKT_RDY bit
[EPiICS].B = [R1L]
[R1L] = [EPiICS].B
OR.B #02, R1L           ;write “1” back to UNDER_RUN bit
AND.B #0F6H, R1L        ;reset ISO bit = 0, write “0” back to IN_PKT_RDY bit
[EPiICS].B= [R1L]
```

1.5.5.2 USB Consecutive Set Address

The USB Specification states that the host can send a SET_ADDRESS request for the following cases:

1. During enumeration when the device is in default state. (The host assigns a non-zero address.)
2. When the device is in the address state. (The host can re-assign a new address.)

The device handles case #1 (when the device is in the default state) and case #2 (when the device is in the address state) differently. The following is a segment of code to illustrate the program flow to properly deal with these cases.

```
DEFAULT_STATE:
If [USBA].B ==0
[USBA.].B = wValue _ lo     ;If the device is in default state, update address before STATUS
                             completion
R1L = [EP0CS].B             ;USB ENDPOINT 0 CSR
OR.B #58H, R1L              ;Set serviced_out_pkt_rdy & data_end, write “1” back to FORCE_STALL bit
[EP0CS].B = R1L
wait for the completion of the status
JMP ADDR_END
else
ADDR_STATE
R1L [EP0CS].B               ;USB ENDPOINT 0 CSR
OR.B #58H, R1L              ;Set serviced_out_pkt_rdy & data_end, write “1” back to FORCE_STALL bit
[EP0CS].B = R1L
wait for the completion of the status
[USBA].B= wValue_lo         ;If the device is in address state, update address before STATUS
                             completion
ADDR_END
endif
end of the set_address routine
```

Note: wValue_lo = assigned address from the host in SET-ADDRESS request.

Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS

http://www.renesas.com

REVISION HISTORY	M30240 Group Data Sheet

Rev.	Date	Description	
		Page	Summary
G	10 / 23 / 2001		First Edition Issued
H	09 / 18 / 2003	--	Changed to new Renesas format
		125	Table 1.38 : Removed reference to Table 2, added X_{IN} and CNV_{SS} to V_{IH}, V_{IL}
		126	Table 1.39: Removed Icc typical measurements
		125-137	Tables 1.1 to 1.14 numbering changed to Tables 1.39 to 1.52